aws

Entwicklerhandbuch

AWS Elastic Beanstalk



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Elastic Beanstalk: Entwicklerhandbuch

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS Elastic Beanstalk? ... 1
Unterstützte Plattformen ... 2
Arbeitsablauf für die Anwendungsbereitstellung ... 3
Preisgestaltung ... 4
Nächste Schritte ... 4
Erste Schritte-Tutorial ... 5
Was werden Sie bauen ... 7
Schritt 1 — Erstellen Sie eine Anwendung ... 8
Schritt 2 — Stellen Sie Ihre Anwendung bereit ... 10
Schritt 3 — Erkunden Sie die Umgebung ... 13
Fehlerbehebung mit Protokollen ... 16
Schritt 4 — Aktualisieren Sie Ihre Anwendung ... 17
Schritt 5 — Skalieren Sie Ihre Anwendung ... 19
Erhöhen Sie die Kapazitätseinstellungen ... 20
Überprüfen Sie die erhöhte Kapazität ... 21
Bereinigen ... 21
Nächste Schritte ... 23
EB CLI einrichten ... 24
Installieren Sie EB CLI ... 25
Manuelle Installation ... 26
In virtualenv installieren ... 28
Installieren der EB-CLI mit Homebrew ... 29
Konfigurieren der EB CLI ... 30
Ignorieren von Dateien, die .ebignore verwenden ... 33
Verwenden von benannten Profilen ... 33
Bereitstellen eines Artefakts anstelle des Projektordners ... 34
Konfigurationseinstellungen und Vorrang ... 34
Instance-Metadaten ... 35
Verwenden der EB CLI mit Git ... 35
Zuordnen von Git-Verzweigungen zu Elastic Beanstalk-Umgebungen ... 36
Bereitstellen von Änderungen ... 36
Verwenden von Git-Submodulen ... 37
Zuweisen von Git-Tags zu Ihrer Anwendungsversion ... 38
EB-CLI-Befehle ... 39

Allgemeine Optionen 40
eb abort 41
Description 41
Syntax 41
Optionen 41
Ausgabe 41
Beispiel 42
eb appversion 42
Description 42
Syntax 42
Optionen 43
Interaktives Verwenden des Befehls 45
Ausgabe 45
Beispiele 45
eb clone 47
Description 47
Syntax 47
Optionen 48
Ausgabe 49
Beispiel 49
eb codesource 50
Description 50
Syntax 50
Optionen 50
Ausgabe 51
Beispiele 51
eb config 52
Description 52
Syntax 52
Optionen 54
Ausgabe 57
Beispiele 58
eb console 62
Description 62
Syntax 62
Optionen 62

eb create 62
Description 62
Syntax 63
Optionen 64
Ausgabe 78
Beispiele 78
eb deploy 81
Description 81
Syntax 81
Optionen 81
Ausgabe 83
Beispiel 83
eb events 83
Description 83
Syntax 83
Optionen 83
Ausgabe 84
Beispiel 84
eb health 85
Description 85
Syntax 85
Optionen 85
Ausgabe 85
Beispiel 85
eb init 87
Description 87
Syntax 87
Optionen 87
CodeBuild Unterstützung 90
Ausgabe 90
Beispiel 90
eb labs 91
Description 91
eb list 92
Description 92
Syntax 92

Optionen 92
Ausgabe 92
Beispiel 1 93
Beispiel 2 93
eb local 93
Description 93
Syntax 94
Optionen 94
Ausgabe 95
Beispiele 95
eb logs 97
Description 97
Syntax 97
Optionen 98
Ausgabe 101
Beispiele 101
eb migrate 102
Description 102
Syntax 103
Unterbefehle 104
Optionen 106
Ausgabe 114
Beispiele 115
eb open 119
Description 119
Syntax 119
Optionen 119
Ausgabe 119
eb platform 119
Description 119
Verwenden der eb-Plattform für benutzerdefinierte Plattformen 120
Verwenden der eb-Plattform für Umgebungen 128
eb printenv 130
Description 130
Syntax 130
Optionen 130

Ausgabe 130
Beispiel 130
eb restore 131
Description 131
Syntax 131
Optionen 131
Ausgabe 131
Beispiel 131
eb scale 132
Description 132
Syntax 132
Optionen 132
Ausgabe 133
Beispiel 133
eb setenv 133
Description 133
Syntax 133
Optionen 134
Ausgabe 134
Beispiel 134
eb ssh 134
Description 134
Syntax 135
Optionen 136
Ausgabe 137
Beispiel 137
eb status 138
Description 138
Syntax 138
Optionen 138
Ausgabe 138
Beispiel 139
eb swap 139
Description 139
Syntax 140
Optionen 140

Ausgabe ... 140
Beispiele ... 140
eb tags ... 141
Description ... 141
Syntax ... 141
Optionen ... 142
Ausgabe ... 144
Beispiele ... 144
eb terminate ... 145
Description ... 145
Syntax ... 146
Optionen ... 146
Ausgabe ... 147
Beispiel ... 147
eb upgrade ... 148
Description ... 148
Syntax ... 148
Optionen ... 148
Ausgabe ... 148
Beispiel ... 149
eb use ... 149
Description ... 149
Syntax ... 149
Optionen ... 149
Konzepte ... 151
Anwendung ... 152
Anwendungsversion ... 152
Umgebung ... 152
Umgebungsebene ... 152
Konfiguration der Umgebung ... 153
Gespeicherte Konfiguration ... 153
Plattform ... 153
Webserver-Umgebungen ... 153
Worker-Umgebungen ... 155
Designüberlegungen ... 156
Skalierbarkeit ... 157

Sicherheit ... 158
Persistenter Speicher ... 158
Fehlertoleranz ... 160
Bereitstellung von Inhalten ... 160
Software-Aktualisierungen und Patches ... 160
Konnektivität ... 161
Verwalten von -Anwendungen ... 162
Konsole zur Anwendungsverwaltung ... 163
Verwalten von Anwendungsversionen ... 164
Anwendungsversionen erstellen ... 165
Anwendungsversionen löschen ... 166
Lebenszyklus der Version ... 167
Markieren von Anwendungsversionen ... 169
Erstellen eines Quell-Bundles ... 172
Erstellen eines Quell-Bundles mit der Befehlszeile ... 173
Erstellen eines Quell-Bundles mit Git ... 173
Komprimieren von Dateien im Mac OS X Finder oder Windows-Explorer ... 173
Erstellen eines Quell-Bundles für eine .NET-Anwendung ... 174
Testen des Quell-Bundles ... 176
Bauen mit CodeBuild ... 176
Erstellen einer Anwendung ... 177
Erstellen und Bereitstellen Ihres Anwendungscodes ... 177
Markieren von Anwendungen ... 179
Hinzufügen von Tags während der Anwendungserstellung ... 179
Verwalten der Tags einer vorhandenen Anwendung ... 180
Taggen von Ressourcen ... 181
Ressourcen, die markiert werden können ... 183
Tag-Propagierung zu Startvorlagen ... 183
Verwalten von Umgebungen ... 185
Environment Management Console ... 187
Zugriff auf die Konsole ... 187
Bereich „Übersicht über die Umgebung“ ... 188
Einzelheiten zur Umgebung ... 190
Umgebungsaktionen ... 193
Erstellen von Umgebungen ... 196
Der Assistent zum Erstellen einer neuen Umgebung ... 201

Klonen einer Umgebung ... 223
Beenden einer Umgebung ... 226
Mit dem AWS CLI ... 228
Mit der API ... 230
Launch-Now-URL ... 234
Compose-Umgebungen ... 238
Zusammenstellen von Umgebungen ... 242
Bereitstellungen ... 244
Auswählen einer Bereitstellungsrichtlinie ... 245
Bereitstellen einer neuen Anwendungsversion ... 248
Erneute Bereitstellung einer früheren Version ... 248
Weitere Möglichkeiten zur Bereitstellung Ihrer Anwendung ... 248
Optionen für die Bereitstellung ... 249
Blau/Grün-Bereitstellungen ... 258
CI/CD-Integration ... 261
Konfigurationsänderungen ... 263
Fortlaufende Updates ... 265
Unveränderliche Updates ... 271
Plattformaktualisierungen ... 275
Methode 1 – Aktualisieren der Plattformversion Ihrer Umgebung ... 279
Methode 2 — Führen Sie eine Bereitstellung durch Blue/Green ... 280
Verwaltete Aktualisierungen ... 282
Aktualisieren einer Legacy-Umgebung ... 289
Migrieren Sie zu AL2 023/ AL2 ... 291
Häufig gestellte Fragen zur Einstellung von Plattformen ... 311
Abbrechen eines Updates ... 316
Neuerstellung einer Umgebung ... 317
Neuerstellung einer laufenden Umgebung ... 317
Neuerstellung einer beendeten Umgebung ... 318
Typen von Umgebungen ... 320
Skalierbare Umgebung mit Lastausgleich ... 321
Single-Instance-Umgebung ... 321
Ändern des Umgebungstyps ... 321
Worker-Umgebungen ... 323
SQS-Daemon in der Worker-Umgebung ... 325
Warteschlangen mit unzustellbaren Briefen ... 326

Regelmäßige Aufgaben ... 327
Verwenden Sie Amazon CloudWatch für die automatische Skalierung in den Stufen der Mitarbeiterumgebung ... 329
Konfigurieren von Worker-Umgebungen ... 330
Umgebungslinks ... 334
Wiederherstellung nach einem ungültigen Stack-Status ... 336
Behebung des Fehlers ... 336
Warum tritt der Fehler auf ... 337
Konfigurieren von Umgebungen ... 339
Bereitgestellte Ressourcen ... 339
Konfiguration mithilfe der Konsole ... 341
Konfigurationsseite ... 341
Seite "Änderungen überprüfen" ... 344
EC2 Amazon-Instanzen ... 345
EC2 Amazon-Instance-Typen ... 346
Konfiguration mit der Konsole ... 347
Verwaltung von EC2-Sicherheitsgruppen ... 356
Konfiguration mit dem AWS CLI ... 358
Konfiguration mit Namespace ... 369
IMDS ... 370
Auto Scaling-Gruppe ... 373
Startvorlagen ... 375
Spot-Instance-Unterstützung ... 378
Auslöser ... 396
Geplante -Aktionen ... 398
Zustandsprüfungseinstellung ... 403
Load Balancer ... 404
Classic Load Balancer ... 406
Application Load Balancer ... 418
Geteile Application Load Balancer ... 440
Network Load Balancer ... 460
Konfiguration von Dual-Stack ... 473
Konfigurieren von Zugriffsprotokollen ... 485
Datenbank ... 485
Datenbank-Lebenszyklus ... 486
Hinzufügen einer Amazon-RDS-DB-Instance zur Umgebung mithilfe der Konsole ... 487

Verbinden mit der Datenbank ... 489
Konfigurieren einer integrierten RDS-DB-Instance mithilfe der Konsole ... 490
Konfigurieren einer integrierten RDS-DB-Instance mithilfe von Konfigurationsdateien ... 491
Entkoppeln einer RDS-DB-Instance über die Konsole ... 491
Entkoppeln einer RDS-DB-Instance mithilfe von Konfigurationsdateien ... 495
Sicherheit ... 497
Konfigurieren der Umgebungssicherheit ... 497
Namespaces für die Konfiguration der Umgebungssicherheit ... 500
Tagging-Umgebungen ... 501
Hinzufügen von Tags beim Erstellen einer Umgebung ... 501
Verwalten von Tags von bestehenden Umgebungen ... 502
Umgebungsvariablen und Softwareeinstellungen ... 505
Plattformspezifische Einstellungen konfigurieren ... 505
Konfiguration von Umgebungseigenschaften (Umgebungsvariablen) ... 506
Softwareeinstellung Namespaces ... 508
Zugreifen auf Umgebungseigenschaften ... 510
Debuggen ... 512
Anzeigen von Protokollen ... 515
Benachrichtigungen ... 518
Benachrichtigungen mit der Elastic Beanstalk-Konsole konfigurieren ... 519
Konfigurieren von Benachrichtigungen mithilfe von Konfigurationsoptionen ... 520
Konfigurieren von Berechtigungen zum Senden von Benachrichtigungen ... 523
Amazon VPC ... 525
VPC-Einstellungen in der Elastic Beanstalk-Konsole konfigurieren ... 525
Der aws:ec2:vpc-Namespace ... 528
Migration von EC2 -Classic zu einer VPC ... 529
Domainname ... 534
Konfigurieren von Umgebungen (Erweitert) ... 537
Konfigurationsoptionen ... 538
Precedence ... 538
Empfohlene Werte ... 539
Vor der Umgebungserstellung ... 542
Während der Erstellung ... 548
Nach der Erstellung ... 555
Allgemeine Optionen ... 566
Plattformspezifische Optionen ... 666

Benutzerdefinierte Optionen ... 679
.Ebextensions ... 680
Optionseinstellungen ... 683
Linux-Server ... 685
Windows-Server ... 703
Benutzerdefinierte Ressourcen ... 713
Gespeicherte Konfigurationen ... 743
Markieren von gespeicherten Konfigurationen ... 749
`env.yaml` ... 752
Benutzerdefiniertes Image ... 755
Erstellen eines benutzerdefinierten AMI ... 755
Verwaltung eines benutzerdefinierten AMI ... 760
Bereinigen eines benutzerdefinierten AMI ... 761
AMI basiert auf einer außer Betrieb genommenen Plattform ... 761
Statische Dateien ... 768
Konfigurieren statischer Dateien über die Konsole ... 769
Konfigurieren statischer Dateien mithilfe von Konfigurationsoptionen ... 770
HTTPS ... 771
Serverzertifikate ... 772
Beenden Sie HTTPS am Load Balancer ... 780
Beenden Sie HTTPS auf der Instanz ... 784
End-to-end Verschlüsselung ... 821
TCP-Pass-Through ... 826
Umleitung von HTTP zu HTTPS ... 827
Plattformen ... 829
Glossar zu Plattformen ... 829
Modell der geteilten Verantwortung ... 833
Plattform-Unterstützungsrichtlinie ... 835
Ausgemusterte Plattformvarianten ... 835
Über die 90-Tage-Nachfrist hinaus ... 836
Zeitplan für die Plattform ... 836
Ressourcen planen ... 837
Kommende Veröffentlichungen in den Plattformbranchen ... 837
Zeitplan für die Ausmusterung von Plattformvarianten ... 838
Bisherige eingestellte Plattformzweige ... 839
Server- und Betriebssystemverlauf ... 844

Unterstützte Plattformen ... 846
Unterstützte Plattformen und Komponentenverlauf ... 846
Linux-Plattformen ... 847
Unterstützte Amazon-Linux-Versionen ... 848
Liste der Elastic Beanstalk-Linux-Plattformen ... 851
Instance-Bereitstellungs-Workflow ... 851
Workflow zur Instanzbereitstellung für ECS auf AL2 und später ... 855
Plattform-Skript-Tools ... 857
Erweitern von Linux-Plattformen ... 867
Buildfile und Procfile ... 867
Plattform-Hooks ... 870
Konfigurationsdateien ... 872
Reverse-Proxy-Konfiguration ... 873
Anwendungsbeispiel mit Erweiterungen ... 877
Bereitstellen von.NET (Windows) ... 879
QuickStart für .NET Core unter Windows ... 880
Ihr AWS Konto ... 881
Voraussetzungen ... 883
Schritt 1: Erstellen Sie eine.NET Core-Anwendung für Windows ... 884
Schritt 2: Führen Sie Ihre Anwendung lokal aus ... 884
Schritt 3: Stellen Sie Ihre.NET Core on Windows-Anwendung mit der EB CLI bereit ... 885
Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus ... 886
Schritt 5: Bereinigen ... 887
AWS Ressourcen für Ihre Anwendung ... 887
Nächste Schritte ... 888
Mit der Konsole bereitstellen ... 888
QuickStart für ASP.NET ... 888
Ihr AWS Konto ... 889
Voraussetzungen ... 891
Schritt 1: Erstellen Sie eine ASP.NET-Anwendung ... 892
Schritt 2: Führen Sie Ihre Anwendung lokal aus ... 893
Schritt 3: Stellen Sie Ihre ASP.NET-Anwendung mit dem bereit AWS Toolkit for Visual Studio ... 893
Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus ... 894
Schritt 5: Bereinigen ... 894
AWS Ressourcen für Ihre Anwendung ... 894

Nächste Schritte ... 895
Mit der Konsole bereitstellen ... 896
Entwicklungsumgebung ... 896
Installieren einer IDE ... 897
Installation des AWS Toolkit for Visual Studio ... 897
Die .NET-Plattform ... 897
Konfigurieren Ihrer .NET-Umgebung in der Elastic Beanstalk-Konsole ... 898
Der Namespace aws:elasticbeanstalk:container:dotnet:apppool ... 899
Migration über Hauptversionen ... 900
Bereitstellungsmanifest ... 903
EC2 Schneller Start ... 929
Hinzufügen einer Datenbank ... 930
Hinzufügen einer DB-Instance zu Ihrer Umgebung ... 931
Herunterladen eines Treibers ... 932
Verbinden mit einer Datenbank ... 932
Die AWS Toolkit for Visual Studio ... 934
Ausführen lokaler Tests ... 935
Erstellen einer Elastic Beanstalk-Umgebung ... 935
Beenden einer Umgebung ... 944
Bereitstellen ... 945
Verwalten von Umgebungen ... 948
Verwalten von Konten ... 962
Debuggen ... 963
Monitor ... 965
Bereitstellungstools ... 967
Migrieren von On-Premise-Anwendungen ... 967
Empfehlungen für ausrangierte Komponenten ... 967
Ausrangierte Windows 2012 R2-Filialen ... 968
TLS 1.2 Kompatibilität ... 968
Bereitstellen von.NET Core (Linux) ... 970
QuickStart für .NET Core unter Linux ... 971
Ihr AWS Konto ... 971
Voraussetzungen ... 973
Schritt 1: Erstellen Sie eine.NET-Core-Anwendung unter Linux ... 974
Schritt 2: Führen Sie Ihre Anwendung lokal aus ... 974
Schritt 3: Stellen Sie Ihre.NET Core on Linux-Anwendung mit der EB CLI bereit ... 975

Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus 977
Schritt 5: Bereinigen 887
AWS Ressourcen für Ihre Anwendung 978
Nächste Schritte 978
Mit der Konsole bereitstellen 979
Entwicklungsumgebung 979
Installation des .NET Core SDK 979
Installieren einer IDE 897
Installation des AWS Toolkit for Visual Studio 897
Der.NET-Kern auf der Linux-Plattform 980
Überlegungen zur Plattform .NET Core auf Linux 980
Konfigurieren Ihrer .NET Core unter Linux-Umgebung 981
.NET Core unter Linux-Konfigurations-Namespace 983
Anwendungsbündelung 983
Procfile 985
Proxy-Server 986
Das AWS Toolkit for Visual Studio 987
Voraussetzungen 987
Erstellen eines neuen Anwendungsprojekts 988
Erstellen einer Elastic Beanstalk-Umgebung und Bereitstellen Ihrer Anwendung 989
Beenden einer Umgebung 991
Verwalten von Umgebungen 992
Monitor 1006
Migration von Windows zu Linux 1008
Aspekte, die bei der Migration zur .NET Core unter Linux-Plattform berücksichtigt werden sollten 1008
Bereitstellung von Go 1009
QuickStart für Go 1009
Ihr AWS Konto 1010
Voraussetzungen 1012
Schritt 1: Erstellen Sie eine Go-Anwendung 1012
Schritt 2: Stellen Sie Ihre Go-Anwendung mit der EB CLI bereit 1013
Schritt 3: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus 1014
Schritt 4: Bereinigen 887
AWS Ressourcen für Ihre Anwendung 1015
Nächste Schritte 1016

Mit der Konsole bereitstellen ... 1016
Entwicklungsumgebung ... 1016
Installieren von Go ... 1016
Das AWS SDK for Go installieren ... 1017
Die Go-Plattform ... 1017
Konfigurieren Ihrer Go-Umgebung ... 1018
Namespace der Go-Konfiguration ... 1019
Procfile ... 1021
Buildfile ... 1023
Proxykonfiguration ... 1024
Bereitstellung von Java ... 1027
QuickStart für Java ... 1028
Ihr AWS Konto ... 1029
Voraussetzungen ... 1031
Schritt 1: Erstellen Sie eine Java-Anwendung ... 1032
Schritt 2: Führen Sie Ihre Anwendung lokal aus ... 1035
Schritt 3: Stellen Sie Ihre Java-Anwendung mit der EB CLI bereit ... 1035
Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus ... 1037
Schritt 5: Bereinigen ... 1037
AWS Ressourcen für Ihre Anwendung ... 1037
Nächste Schritte ... 1038
Mit der Konsole bereitstellen ... 1039
QuickStart für Java auf Tomcat ... 1039
Ihr AWS Konto ... 1039
Voraussetzungen ... 1042
Schritt 1: Erstellen Sie eine Java-JSP-Anwendung ... 1042
Schritt 2: Stellen Sie Ihre Java-JSP-Anwendung mit der EB CLI bereit ... 1043
Schritt 3: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus ... 1044
Schritt 4: Bereinigen ... 887
AWS Ressourcen für Ihre Anwendung ... 1044
Nächste Schritte ... 1045
Mit der Konsole bereitstellen ... 1046
Entwicklungsumgebung ... 1046
Installieren des Java Development Kits ... 1046
Installieren eines Webcontainers ... 1046
Herunterladen von Bibliotheken ... 1047

Das AWS SDK for Java installieren ... 1047
Installieren einer IDE oder eines Texteditors ... 1047
Beispielanwendungen und Tutorials ... 1048
Starten einer Umgebung mit einer Java-Beispielanwendung ... 1048
Nächste Schritte ... 1057
Die Tomcat-Plattform ... 1057
Konfigurieren Ihrer Tomcat-Umgebung ... 1059
Tomcat-Konfigurations-Namespaces ... 1063
Bündeln von WAR-Dateien ... 1065
Strukturieren Ihres Projektordners ... 1067
Proxykonfiguration ... 1070
Die Java SE-Plattform ... 1076
Konfigurieren der Java SE-Umgebung ... 1077
Java SE-Konfigurations-Namespace ... 1079
Buildfile ... 1080
Procfile ... 1081
Proxykonfiguration ... 1083
Hinzufügen einer Datenbank ... 1086
Herunterladen eines JDBC-Treibers ... 1088
Verbinden mit einer Datenbank (Java SE-Plattformen) ... 1089
Verbinden mit einer Datenbank (Tomcat-Plattformen) ... 1090
Fehlerbehebung bei Datenbankverbindungen ... 1093
Ressourcen ... 1096
Bereitstellung von Node.js ... 1097
QuickStart für Node.js ... 1098
Ihr AWS Konto ... 1098
Voraussetzungen ... 1100
Schritt 1: Erstellen Sie eine Node.js -Anwendung ... 1101
Schritt 2: Führen Sie Ihre Anwendung lokal aus ... 1102
Schritt 3: Stellen Sie Ihre Anwendung Node.js mit der EB CLI bereit ... 1102
Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus ... 1103
Schritt 5: Bereinigen ... 887
AWS Ressourcen für Ihre Anwendung ... 1104
Nächste Schritte ... 1105
Mit der Konsole bereitstellen ... 1105
Entwicklungsumgebung ... 1105

Installieren von Node.js ... 1106
Bestätigen der npm-Installation ... 1106
Installieren Sie das AWS SDK für Node.js ... 1106
Installieren des Express-Generators ... 1106
Einrichten eines Express-Frameworks und eines Servers ... 1108
Die Node.js-Plattform ... 1108
Konfiguration der Node.js-Umgebung ... 1110
Node.js-Konfigurations-Namespace ... 1112
Procfile ... 1116
Konfigurieren von Abhängigkeiten ... 1117
npm Shrinkwrap-Datei ... 1122
Proxykonfiguration ... 1122
Beispielanwendungen und Tutorials ... 1126
Starten einer Umgebung mit einer Beispielanwendung von Node.js ... 1127
Nächste Schritte ... 1128
Tutorial – Express ... 1128
Voraussetzungen ... 1128
Erstellen einer Elastic Beanstalk-Umgebung ... 1129
Aktualisieren der Anwendung zur Nutzung von Express ... 1132
Aktualisieren der Anwendung, um Amazon RDS zu nutzen ... 1135
Bereinigen ... 1140
Tutorial – Express mit Clustering ... 1141
Voraussetzungen ... 1141
Erstellen einer Elastic Beanstalk-Umgebung ... 1142
Aktualisieren der Anwendung zur Nutzung von Express ... 1144
Bereinigen ... 1159
Tutorial – Node.js mit DynamoDB ... 1159
Voraussetzungen ... 1160
Erstellen einer Elastic Beanstalk-Umgebung ... 1160
Hinzufügen von Berechtigungen zu den Instances Ihrer Umgebung ... 1163
Bereitstellen der Beispielanwendung ... 1164
Erstellen einer DynamoDB-Tabelle ... 1167
Aktualisieren der Konfigurationsdateien der Anwendung ... 1167
Konfigurieren der Umgebung für hohe Verfügbarkeit ... 1169
Bereinigen ... 1170
Nächste Schritte ... 1171

Hinzufügen einer Datenbank ... 1171
Hinzufügen einer DB-Instance zu Ihrer Umgebung ... 1172
Herunterladen eines Treibers ... 1173
Herstellen einer Verbindung zu einer Datenbank ... 1174
Ressourcen ... 1175
Bereitstellen von PHP ... 1176
QuickStart für PHP ... 1176
Ihr AWS Konto ... 1177
Voraussetzungen ... 1179
Schritt 1: Erstellen Sie eine PHP-Anwendung ... 1179
Schritt 2: Führen Sie Ihre Anwendung lokal aus ... 1180
Schritt 3: Initialisieren und implementieren Sie Ihre PHP-Anwendung ... 1180
Schritt 4: Durchsuchen Sie Ihre Cloud-Anwendung ... 1181
Schritt 5: Aktualisieren Sie Ihre Anwendung und stellen Sie sie erneut bereit ... 1181
Bereinigen ... 1182
Nächste Schritte ... 1183
PHP-Plattform ... 1183
Das AWS SDK for PHP installieren ... 1184
Überlegungen zu PHP 8.1 auf Amazon Linux 2 ... 1184
Konfigurieren Ihrer PHP-Umgebung ... 1187
Namespaces für die Konfiguration ... 1189
Installieren von Abhängigkeiten ... 1190
Aktualisieren von Composer ... 1192
Erweitern von "php.ini" ... 1193
Fortschrittliche -Beispiele ... 1194
Hinzufügen einer Datenbank ... 1194
Tutorial – Laravel ... 1198
Tutorial – CakePHP ... 1209
Tutorial – Symfony ... 1217
Tutorial – HA-Produktion ... 1224
Lernprogramm — HA WordPress ... 1236
Tutorial – Drupal mit hoher Verfügbarkeit ... 1254
Python bereitstellen ... 1274
QuickStart für Python ... 1274
Ihr AWS Konto ... 1275
Voraussetzungen ... 1277

Schritt 1: Erstellen Sie eine Python-Anwendung 1279
Schritt 2: Führen Sie Ihre Anwendung lokal aus 1280
Schritt 3: Stellen Sie Ihre Python-Anwendung mit der EB CLI bereit 1280
Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus 1281
Schritt 5: Bereinigen 887
AWS Ressourcen für Ihre Anwendung 1282
Nächste Schritte 1282
Mit der Konsole bereitstellen 1283
Entwicklungsumgebung 1283
Voraussetzungen 1284
Verwenden einer virtuellen Umgebung 1285
Konfigurieren eines Python-Projekts für Elastic Beanstalk 1286
Die Python-Plattform 1287
Konfigurieren Ihrer Python-Umgebung 1288
Namespaces der Python-Konfiguration 1290
`python3`Die ausführbare Datei 1292
Procfile 1293
Angeben von Abhängigkeiten 1294
Tutorial – Flask 1296
Voraussetzungen 1297
Einrichten einer virtuellen Python-Umgebung mit Flask 1297
Erstellen einer Flask-Anwendung 1299
Bereitstellen Ihrer Website mit der EB-CLI 1301
Bereinigen 1305
Nächste Schritte 1305
Tutorial – Django 1305
Voraussetzungen 1306
Einrichten einer virtuellen Python-Umgebung und Installieren von Django 1306
Erstellen eines Django-Projekts 1308
Konfigurieren der Django-Anwendung für Elastic Beanstalk 1310
Bereitstellen Ihrer Website mit der EB-CLI 1312
Aktualisieren der Anwendung 1315
Bereinigen 1319
Nächste Schritte 1319
Hinzufügen einer Datenbank 1320
Hinzufügen einer DB-Instance zu Ihrer Umgebung 1320

Herunterladen eines Treibers ... 1321
Herstellen einer Verbindung zu einer Datenbank ... 1322
Ressourcen ... 1323
Bereitstellung von Ruby ... 1324
Entwicklungsumgebung ... 1324
Installieren von Ruby ... 1325
Das AWS SDK for Ruby installieren ... 1326
Installieren einer IDE oder eines Texteditors ... 1326
Die Ruby-Plattform ... 1327
Konfigurieren Ihrer Ruby-Umgebung ... 1328
Namespaces der Ruby-Konfigurationen ... 1331
Gemfile ... 1332
Procfile ... 1333
Tutorial – Rails ... 1334
Voraussetzungen ... 1335
Grundkenntnisse zu Elastic Beanstalk ... 1335
Starten einer Elastic Beanstalk-Umgebung ... 1336
Installieren von Rails und Generieren einer Website ... 1338
Konfigurieren von Rails-Einstellungen ... 1342
Bereitstellen der Anwendung ... 1343
Bereinigen ... 1343
Nächste Schritte ... 1344
Tutorial – Sinatra ... 1344
Voraussetzungen ... 1344
Starten einer Elastic Beanstalk-Umgebung ... 1345
Schreiben einer einfachen Sinatra-Website ... 1347
Bereitstellen der Anwendung ... 1348
Bereinigen ... 1349
Nächste Schritte ... 1349
Hinzufügen einer Datenbank ... 1350
Hinzufügen einer DB-Instance zu Ihrer Umgebung ... 1350
Herunterladen eines Adapters ... 1352
Herstellen einer Verbindung zu einer Datenbank ... 1352
Bereitstellung mit Docker ... 1353
Filialen der Docker-Plattform ... 1353
Stillgelegte Plattformzweige, die auf Amazon Linux AMI (AL1) laufen ... 1354

Docker-Plattformzweig 1355
QuickStart für Docker 1355
QuickStart für Docker Compose 1363
Docker-Image-Konfiguration 1374
ECS-verwalteter Plattformzweig 1381
Überblick über die von ECS verwaltete Docker-Plattform 1382
Von Elastic Beanstalk erstellte Amazon ECS-Ressourcen 1382
`Dockerrun.aws.json`v2-Datei 1383
Docker-Images 1384
Fehlgeschlagene Container-Bereitstellungen 1384
Erweiterung der ECS-basierten Docker-Plattformen 1385
Von ECS verwaltete Docker-Konfiguration für Elastic Beanstalk 1385
Tutorial – ECS-verwaltete Docker 1398
Migration zu ECS, das auf 0.23 läuft AL2 1408
Authentifizierung mit Image-Repositorys 1410
Amazon ECR-Repository 1410
AWS Secrets Manager 1412
Dockerrun.aws.json-Datei 1414
Öffentliche Bilder von Amazon ECR 1416
Konfiguration der Umgebung 1417
Konfigurieren von Software in Docker-Umgebungen 1418
Verweisen auf Umgebungsvariablen in Containern 1420
Verwenden der Interpolationsfunktion für Umgebungsvariablen mit Docker Compose 1422
Generieren von Protokollen für verbesserte Gesundheitsberichte mit Docker Compose 1423
Benutzerdefinierte Protokollierung für Docker-Container mit Docker Compose 1423
Docker-Images 1425
Konfigurieren von verwalteten Aktualisierungen für Docker-Umgebungen 1425
Docker-Konfigurations-Namespaces 1426
Docker-Konfiguration auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2) 1427
Ältere Plattformen 1429
(Legacy-System) Migration zu Docker auf Amazon Linux 2 1429
(Ältere) Docker-Container GlassFish 1435
Umgebungen überwachen 1445
Überwachungskonsole 1445
Überwachungsdiagramme 1446
Anpassen der Überwachungskonsole 1447

Überwachung des Zustands mit CLI ... 1448
Lesen der Ausgabe ... 1452
Interaktive Ansicht des Zustands ... 1454
Optionen für die Interaktive Ansicht des Zustands ... 1456
Grundlegende Zustandsberichte ... 1456
Zustandsfarben ... 1457
Zustandsprüfungen von Elastic Load Balancing ... 1458
Zustandsprüfungen für Umgebungen mit einer einzelnen Instance oder Worker-Ebene ... 1459
Zusätzliche Prüfungen ... 1459
CloudWatch Amazon-Metriken ... 1460
Erweiterte Zustandsberichte und -überwachung ... 1461
Der Elastic-Beanstalk-Integritäts-Agent ... 1464
Faktoren bei der Bestimmung des Instance- und Umgebungszustands ... 1465
Anpassung der Regel für die Zustandsprüfung ... 1467
Rollen in erweiterten Zustandsberichten ... 1468
Erweiterte Zustandsautorisierung ... 1468
Ereignisse in erweiterten Zustandsberichten ... 1470
Verhalten der erweiterten Zustandsberichte bei Aktualisierungen, Bereitstellungen und Skalierung ... 1470
Aktivieren der erweiterten Zustandsberichte ... 1471
Konsole für die Integritätsüberwachung ... 1475
Farben und Status in Zustandsangaben ... 1480
Instance-Metriken ... 1483
Regeln für den erweiterten Zustand ... 1486
CloudWatch ... 1492
API-Benutzer ... 1501
Format der Protokolle der erweiterten Zustandsberichte ... 1503
Benachrichtigungen und Fehlerbehebung ... 1507
KI-gestützte Umgebungsanalyse ... 1509
Alarme verwalten ... 1513
Anzeigen des Änderungsverlaufs ... 1516
Anzeigen von Ereignissen ... 1518
Konsole ... 1518
Befehlszeile ... 1518
Überwachen von Instances ... 1519
Anzeigen von Instance-Protokollen ... 1521

Speicherort der Protokolle auf Amazon EC2-Instances ... 1523
Speicherort der Protokolle in Amazon S3 ... 1524
Protokollrotations-Einstellungen auf Linux ... 1525
Erweitern der Standardkonfiguration für Protokollaufgaben ... 1525
Protokolldateien nach Amazon CloudWatch Logs streamen ... 1529
Bereitstellungsprotokolle ... 1530
Unterstützte Vorgänge ... 1531
Inhalte protokollieren ... 1531
Konsole ... 1532
Instanz-Protokolldateien ... 1534
Dienste integrieren AWS ... 1536
Übersicht über die Architektur ... 1536
CloudFront ... 1537
CloudTrail ... 1538
Informationen zu Elastic Beanstalk in CloudTrail ... 1538
Grundlegendes zu den Elastic Beanstalk-Einträgen in der Protokolldatei ... 1539
CloudWatch ... 1540
CloudWatch Logs ... 1541
Voraussetzungen für das Streaming von CloudWatch Instanzprotokollen in Logs ... 1542
So richtet Elastic Beanstalk Logs ein CloudWatch ... 1543
Die Streaming-Instanz protokolliert in CloudWatch Logs ... 1548
Fehlerbehebung bei der Integration CloudWatch von Logs ... 1551
Streamen des Umgebungszustands ... 1552
EventBridge ... 1556
Überwachen Sie eine Elastic Beanstalk Beanstalk-Ressource mit EventBridge ... 1557
Beispiel für Elastic-Beanstalk-Ereignisse ... 1560
Beispiel für Elastic Beanstalk-Ereignisse ... 1562
Elastic Beanstalk-Ereignisfeldzuordnung ... 1563
AWS Config ... 1566
Einrichten AWS Config ... 1567
Konfiguration für AWS Config die Aufzeichnung von Elastic Beanstalk Beanstalk-Ressourcen ... 1567
Elastic Beanstalk Beanstalk-Konfigurationsdetails in der Konsole anzeigen AWS Config ... 1569
Evaluierung von Elastic Beanstalk Beanstalk-Ressourcen mithilfe von Regeln AWS Config ... 1572
DynamoDB ... 1573

ElastiCache 1574
Amazon EFS 1575
Konfigurationsdateien 1575
Verschlüsselte Dateisysteme 1576
Beispielanwendungen 1577
Bereinigen von Dateisystemen 1577
IAM 1578
Instance-Profile 1578
Servicerollen 1584
Verwenden von servicegebundenen Rollen 1592
Benutzerrichtlinien 1605
ARN-Format 1613
Ressourcen und Bedingungen 1615
Tag-basierte Zugriffskontrolle 1661
Beispiel für verwaltete Richtlinien 1666
Beispiel für ressourcenspezifische Richtlinien 1670
Umgebungsübergreifender S3-Bucket-Zugriff 1682
Amazon RDS 1684
Amazon RDS in Standard-VPC 1686
Amazon RDS-Anmeldeinformationen und Secrets Manager 1692
Bereinigen einer externen Amazon RDS-Instance 1693
Amazon S3 1693
Der Elastic Beanstalk Amazon S3 S3-Kundenkonto-Bucket 1693
Inhalt des Elastic Beanstalk Amazon S3 S3-Kundenkonto-Buckets 1694
Löschen von Objekten im Elastic Beanstalk Amazon S3-Bucket 1695
Löschen des Elastic Beanstalk Amazon S3-Buckets 1696
Parameterspeicher für Secrets Manager und Systems Manager 1697
Abrufen von Geheimnissen in Umgebungsvariablen 1698
Erforderliche IAM-Berechtigungen 1708
Verwenden von Secrets Manager und Systems Manager Parameter Store 1711
Fehlerbehebung bei Geheimnissen und Umgebungsvariablen 1713
Amazon VPC 1715
Öffentliche VPC 1717
Öffentliche/private VPC 1718
Private VPC 1718
Bastion-Hosts 1721

Amazon RDS ... 1726
VPC-Endpunkte ... 1734
VPC-Endpunktrichtlinien ... 1738
Sicherheit ... 1749
Datenschutz ... 1750
Datenverschlüsselung ... 1751
Datenschutz zwischen Netzwerken ... 1752
Identity and Access Management ... 1752
AWS verwaltete Richtlinien ... 1753
Protokollierung und Überwachung ... 1769
Erweiterte Integritätsberichte ... 1769
EC2 Amazon-Instanzprotokolle ... 1769
Umgebungsbenachrichtigungen ... 1770
CloudWatch Amazon-Alarme ... 1770
AWS CloudTrail Logs ... 1770
AWS X-Ray Debuggen ... 1770
Compliance-Validierung ... 1771
Ausfallsicherheit ... 1772
Sicherheit der Infrastruktur ... 1772
Modell der geteilten Verantwortung ... 1773
Bewährte Methoden für die Gewährleistung der Sicherheit ... 1773
Bewährte Methoden für vorbeugende Sicherheitsmaßnahmen ... 1773
Bewährte Methoden für aufdeckende Sicherheitsmaßnahmen ... 1775
Berechtigungen ... 1777
Erforderliche Rollen ... 1777
Optionale Richtlinien und Rollen ... 1777
Service – -Rolle ... 1778
`AWSElasticBeanstalkEnhancedHealth` ... 1585
`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` ... 1585
Instance-Profil ... 1780
Verwaltete Richtlinien ... 1781
Ein EC2-Instanzprofil erstellen ... 1781
Richtlinie für Benutzer: ... 1781
Tutorials und Beispiele ... 1783
IIS-Anwendungen migrieren ... 1786
Voraussetzungen ... 1787

Glossar zur Migration 1789
Windows-, IIS- und .NET-Begriffe 1790
Allgemeine Geschäftsbedingungen von Elastic Beanstalk 1794
Python-Begriffe 1798
Grundlegende IIS-Migrationen 1800
Erkunden Sie Ihre IIS-Umgebung 1800
Vorbereitung der Migration 1802
Ihre erste Migration 1802
Steuerung der Migration 1803
Überwachen des Fortschritts 1803
Überprüfung der Migration 1804
Verwaltung von Migrationsartefakten 1805
Netzwerkkonfiguration 1805
VPC-Konfiguration 1806
Bereitstellungen an mehreren Standorten mit Portkonfigurationen 1810
Gemeinsame Konfiguration und Abhängigkeiten 1812
Best Practices 1813
Fehlerbehebung 1814
Security configuration (Sicherheitskonfiguration) 1814
Konfiguration des Instanzprofils 1815
Verwaltung von Servicerollen 1815
Sicherheitsgruppenkonfiguration 1816
Integration von SSL-Zertifikaten 1817
Windows-Authentifizierung 1818
Bewährte Methoden und Problembehebung 1819
Zuordnung der Migration von IIS zu Elastic Beanstalk 1821
IIS-Sites und -Anwendungen in Elastic Beanstalk 1821
Verwaltung virtueller Verzeichnisse und Anwendungspfade 1823
URL-Umschreibung und Routing von Anwendungsanfragen (ARR) 1824
Struktur des Migrationsartefakts 1824
Fortgeschrittene Migrationsszenarien 1826
Migrationen an mehreren Standorten mit Application Request Routing (ARR) 1826
Migrationen an mehreren Standorten ohne ARR mithilfe von hostbasiertem Routing 1830
Verwaltung virtueller Verzeichnisse 1834
Benutzerdefinierte Einstellungen für den Anwendungspool 1838
Bereitstellung früherer Versionen 1840

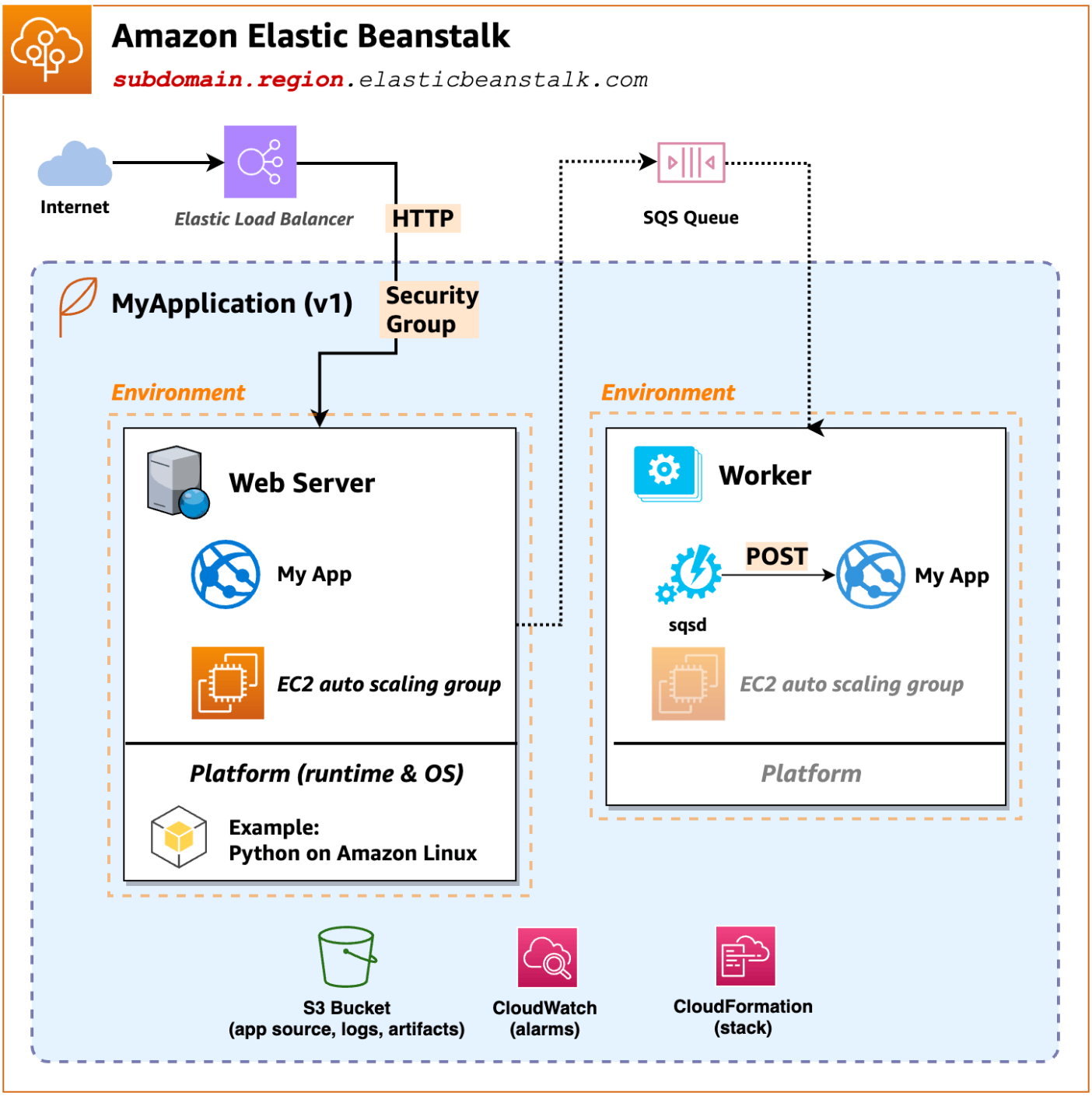
Problembehebung und Diagnose 1841
Ein EC2 Schlüsselpaar mit Ihrer Umgebung verknüpfen 1841
Auf Protokolle zugreifen 1842
Zugreifen auf clientseitige Artefakte 1842
Überwachung des Zustands der Umgebung 1843
EC2 Leistungsoptimierung 1843
Konfiguration des EBS-Volumes 1844
Häufige Probleme und Lösungen 1844
Supportanfragen 1845
Migrationsoptionen: EB CLI im Vergleich zu MGN 1846
Wann sollten die einzelnen Migrationsoptionen verwendet werden 1847
Vergleich der Migrations-Workflows 1848
Schlussfolgerung 1848
Fehlerbehebung 1850
Mithilfe von KI-gestützter Umgebungsanalyse 1851
Verwenden des Systems-Manager-Tools 1851
General guidance (Allgemeine Anleitung) 1853
Umgebungsvariablen für Geheimnisse 1854
Umgebungserstellung 1855
Bereitstellungen 1856
Gesundheit 1856
Mit zwei Stacks konfigurierte Load Balancer 1858
Konfiguration 1859
Docker 1859
Häufig gestellte Fragen 1860
Beheben von Bereitstellungsfehlern 1862
Fehler bei der Bereitstellung 1863
Ressourcen 1866
Beispielanwendungen 1867
AWS SDK für Java 1868
AWS SDK für .NET 1868
AWS Toolkit for Visual Studio 1868
AWS SDK für JavaScript in Node.js 1869
AWS SDK für PHP 1869
AWS SDK für Python (Boto) 1869
AWS SDK für Ruby 1869

Archivierter Inhalt 1871
Benutzerdefinierte Plattformen (im Ruhestand) 1871
Dokumentverlauf 1889
..... mdcccxcvi

Was ist AWS Elastic Beanstalk?

Mit Elastic Beanstalk können Sie Webanwendungen auf einer Vielzahl unterstützter Plattformen in der AWS Cloud bereitstellen. Sie erstellen und implementieren Ihre Anwendungen. Elastic Beanstalk stellt EC2 Amazon-Instances bereit, konfiguriert den Lastenausgleich, richtet die Zustandsüberwachung ein und skaliert Ihre Umgebung dynamisch.

Zusätzlich zu Webserver-Umgebungen bietet Elastic Beanstalk auch Worker-Umgebungen, mit denen Sie Nachrichten aus einer Amazon SQS SQS-Warteschlange verarbeiten können, was für asynchrone oder lang andauernde Aufgaben nützlich ist. Weitere Informationen finden Sie unter Elastic-Beanstalk-Worker-Umgebungen.



Unterstützte Plattformen

Elastic Beanstalk unterstützt Anwendungen, die in Go, Java, .NET, Node.js, PHP, Python und Ruby entwickelt wurden. Elastic Beanstalk unterstützt auch Docker-Container, in denen Sie Ihre eigene Programmiersprache und Anwendungsabhängigkeiten wählen können. Wenn Sie Ihre Anwendung

bereitstellen, erstellt Elastic Beanstalk die ausgewählte unterstützte Plattformversion und stellt eine oder mehrere AWS Ressourcen, wie EC2 Amazon-Instances, in Ihrem AWS Konto bereit, um Ihre Anwendung auszuführen.

Sie können mit Elastic Beanstalk über die Elastic Beanstalk-Konsole, die AWS Command Line Interface (AWS CLI), oder die EB CLI, ein speziell für Elastic Beanstalk entwickeltes Befehlszeilentool auf hoher Ebene, interagieren.

Sie können die meisten Bereitstellungsaufgaben, wie z. B. die Änderung der Größe Ihrer Flotte von EC2 Amazon-Instances oder die Überwachung Ihrer Anwendung, direkt über die Elastic Beanstalk Beanstalk-Weboberfläche (Konsole) ausführen.

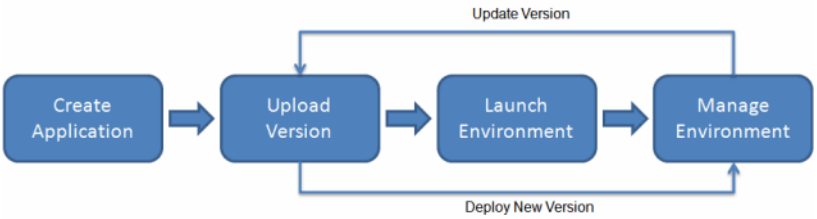
Weitere Informationen zur Bereitstellung einer Beispiel-Webanwendung mit Elastic Beanstalk finden Sie unter. Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können

Arbeitsablauf für die Anwendungsbereitstellung

Um Elastic Beanstalk zu verwenden, erstellen Sie eine Anwendung und laden dann Ihr Anwendungsquellpaket auf Elastic Beanstalk hoch. Als Nächstes geben Sie Informationen über die Anwendung an, und Elastic Beanstalk startet automatisch eine Umgebung und erstellt und konfiguriert die AWS Ressourcen, die für die Ausführung Ihres Codes erforderlich sind.

Nachdem Sie Ihre Anwendung erstellt und bereitgestellt haben und Ihre Umgebung gestartet ist, können Sie Ihre Umgebung verwalten und neue Anwendungsversionen bereitstellen. Informationen über die Anwendung — einschließlich Metriken, Ereignissen und Umgebungsstatus — werden über die Elastic Beanstalk Beanstalk-Konsole und die Befehlszeilenschnittstellen zur Verfügung gestellt. APIs

Das folgende Diagramm veranschaulicht den Elastic Beanstalk Beanstalk-Workflow:



Preisgestaltung

Für die Nutzung von Elastic Beanstalk fallen keine zusätzlichen Gebühren an. Sie zahlen nur für die zugrunde liegenden AWS Ressourcen, die Ihre Anwendung verbraucht. Einzelheiten zur Preisgestaltung finden Sie unter Elastic Beanstalk Service-Detailseite .

Nächste Schritte

Wir empfehlen das Tutorial,Erste Schritte-Tutorial, um mit der Verwendung von Elastic Beanstalk zu beginnen. Das Tutorial führt Sie Schritt für Schritt durch das Erstellen, Anzeigen und Aktualisieren einer Elastic Beanstalk Beanstalk-Beispielanwendung.

Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können

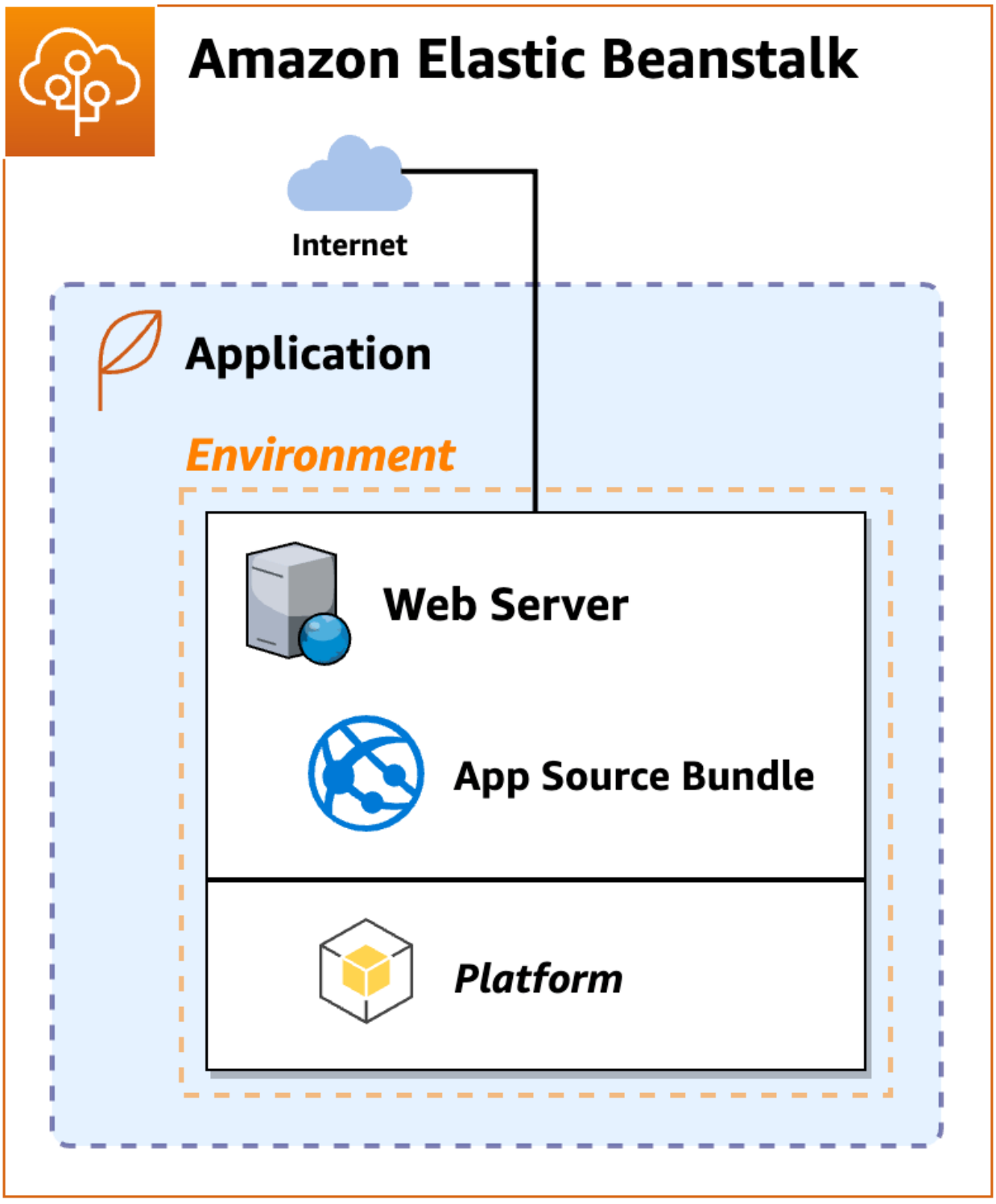
Mit Elastic Beanstalk können Sie Webanwendungen und Services bereitstellen, überwachen und skalieren. In der Regel entwickeln Sie Ihren Code lokal und stellen ihn dann auf Amazon EC2-Server-Instances bereit. Diese Instances, auch Umgebungen genannt, laufen auf Plattformen, die über die AWS Konsole oder die Befehlszeile aktualisiert werden können.

Zu Beginn empfehlen wir, eine vorgefertigte Beispielanwendung direkt von der Konsole aus bereitzustellen. Anschließend erfahren Sie, wie Sie lokal entwickeln und über die Befehlszeile im the section called “QuickStart für PHP” bereitstellen.

Die Nutzung von Elastic Beanstalk ist kostenlos, aber für AWS Ressourcen, die Sie im Laufe dieses Tutorials erstellen, fallen Standardgebühren an, bis Sie sie am Ende löschen. Die Gesamtgebühren betragen in der Regel weniger als einen Dollar. Weitere Informationen zum Minimieren von Gebühren finden Sie unter Kostenloses Kontingent für AWS.

Nach Abschluss dieses Tutorials werden Sie die Grundlagen der Erstellung, Konfiguration, Bereitstellung, Aktualisierung und Überwachung einer Elastic Beanstalk Beanstalk-Anwendung mit Umgebungen verstehen, die auf Amazon EC2 EC2-Instances ausgeführt werden.

Geschätzte Dauer: 35-45 Minuten


Amazon Elastic Beanstalk
Internet
Application
Environment
Web Server
App Source Bundle
Platform

Was werden Sie bauen

Ihre erste Elastic Beanstalk Beanstalk-Anwendung wird aus einer einzigen Amazon EC2 EC2-Umgebung bestehen, in der das PHP-Beispiel auf einer verwalteten PHP-Plattform ausgeführt wird.

Elastic Beanstalk Beanstalk-Anwendung

: Eine Elastic Beanstalk-Anwendung ist ein Container für Elastic Beanstalk Beanstalk-Komponenten, einschließlich Umgebungen, in denen Ihr Anwendungscode auf Plattformen ausgeführt wird, die von Elastic Beanstalk bereitgestellt und verwaltet werden, oder in benutzerdefinierten Containern, die Sie bereitstellen.

Umgebung

: Eine Elastic Beanstalk Beanstalk-Umgebung ist eine Sammlung von AWS Ressourcen, die zusammen ausgeführt werden, einschließlich einer Amazon EC2 EC2-Instance. Wenn Sie eine Umgebung erstellen, stellt Elastic Beanstalk Ihrem AWS Konto die erforderlichen Ressourcen zur Verfügung.

Plattform

: Eine Plattform ist eine Kombination aus einem Betriebssystem, einer Programmiersprache, einer Runtime, einem Webserver, einem Anwendungsserver und zusätzlichen Elastic Beanstalk Beanstalk-Komponenten. Elastic Beanstalk bietet verwaltete Plattformen, oder Sie können Ihre eigene Plattform in einem Container bereitstellen.

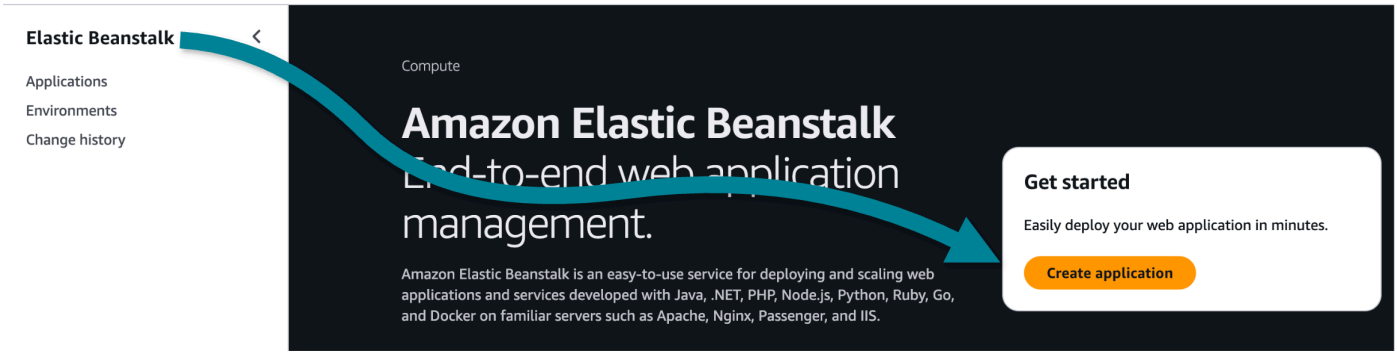
Elastic Beanstalk unterstützt Plattformen für verschiedene Programmiersprachen, Anwendungsserver und Docker-Container. Wenn Sie eine Umgebung erstellen, müssen Sie die Plattform auswählen. Sie können die Plattform aktualisieren, aber Sie können die Plattform nicht für eine Umgebung ändern.

> **Plattformen wechseln**
>
> Wenn Sie die Programmiersprache ändern müssen, müssen Sie eine neue Umgebung auf einer anderen Plattform erstellen und zu dieser wechseln.

Schritt 1 — Erstellen Sie eine Anwendung

Um Ihre Beispielanwendung zu erstellen, verwenden Sie den Konsolenassistenten Create application (Eine Anwendung erstellen). Er erstellt eine Elastic Beanstalk-Anwendung und startet eine Umgebung darin.

Zur Erinnerung: Eine Umgebung ist eine Sammlung von AWS Ressourcen, die zur Ausführung Ihres Anwendungscodes erforderlich sind.



So erstellen Sie eine Anwendung

1. In der Elastic-Beanstalk-Konsole öffnen.
2. Wählen Sie Create application aus.
3. Geben Sie für Application name (Anwendungsname) den Text **`getting-started-app`** ein.

Die Konsole bietet einen sechsstufigen Prozess zum Erstellen einer Anwendung und zum Konfigurieren einer Umgebung. Für diesen Schnellstart müssen Sie sich nur auf die ersten beiden Schritte konzentrieren. Anschließend können Sie überspringen, um Ihre Anwendung und Umgebung zu überprüfen und zu erstellen.

Um eine Umgebung zu konfigurieren

1. Geben Sie im Feld Umgebungsinformationen für Umgebungsname Folgendes ein:**`gs-app-web-env`**.
2. Wählen Sie unter Plattform die PHP-Plattform aus.
3. Akzeptieren Sie für Anwendungscode und Voreinstellungen die Standardeinstellungen (Beispielanwendung und Einzelinstanz) und wählen Sie dann Weiter.

Um den Servicezugriff zu konfigurieren

Als Nächstes benötigen Sie zwei Rollen. Eine Service-Rolle ermöglicht es Elastic Beanstalk, Ihre EC2-Instances zu überwachen und die Plattform Ihrer Umgebung zu aktualisieren. Eine EC2-Instance-Profilrolle ermöglicht Aufgaben wie das Schreiben von Logs und die Interaktion mit anderen Services.

Um die Service-Rolle zu erstellen oder auszuwählen

1. Wenn Sie zuvor eine Servicerolle erstellt haben und eine bestehende auswählen möchten, wählen Sie den Wert aus der Dropdownliste Servicerolle aus und überspringen Sie die restlichen Schritte, um eine Servicerolle zu erstellen.
2. Wenn keine Werte für die Servicerolle aufgeführt sind oder Sie eine neue erstellen möchten, fahren Sie mit den nächsten Schritten fort.
3. Wählen Sie für Servicerolle die Option Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS -Service aus.
5. Wählen Sie als Anwendungsfall Elastic Beanstalk — Environment.
6. Wählen Sie Weiter aus.
7. Vergewissern Sie sich, dass die Berechtigungsrichtlinien Folgendes beinhalten, und wählen Sie dann Weiter aus:

   - `AWSElasticBeanstalkEnhancedHealth`
   - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`
8. Wählen Sie Rolle erstellen aus.
9. Kehren Sie zur Registerkarte Dienstzugriff konfigurieren zurück, aktualisieren Sie die Liste und wählen Sie dann die neu erstellte Servicerolle aus.

Um ein EC2-Instanzprofil zu erstellen oder auszuwählen

1. Wenn Sie zuvor ein EC2-Instanzprofil erstellt haben und ein vorhandenes auswählen möchten, wählen Sie den Wert aus der Dropdownliste für das EC2-Instanzprofil aus und überspringen Sie die restlichen Schritte, um ein EC2-Instanzprofil zu erstellen.
2. Wenn keine Werte für das EC2-Instanzprofil aufgeführt sind oder Sie ein neues erstellen möchten, fahren Sie mit den nächsten Schritten fort.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS -Service aus.

5. Wählen Sie als Anwendungsfall Elastic Beanstalk — Compute.
6. Wählen Sie Weiter aus.
7. Vergewissern Sie sich, dass die Berechtigungsrichtlinien Folgendes beinhalten, und wählen Sie dann Weiter aus:

   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
8. Wählen Sie Rolle erstellen aus.
9. Kehren Sie zur Registerkarte Dienstzugriff konfigurieren zurück, aktualisieren Sie die Liste und wählen Sie dann das neu erstellte EC2-Instanzprofil aus.

Um die Konfiguration und Erstellung Ihrer Anwendung abzuschließen

1. Überspringen Sie das EC2-Schlüsselpaar.

   Wir zeigen Ihnen weitere Möglichkeiten, über die Konsole eine Verbindung zu Ihren Amazon EC2 EC2-Instances herzustellen.
2. Wählen Sie „Zur Überprüfung springen“, um mehrere optionale Schritte durchzugehen.

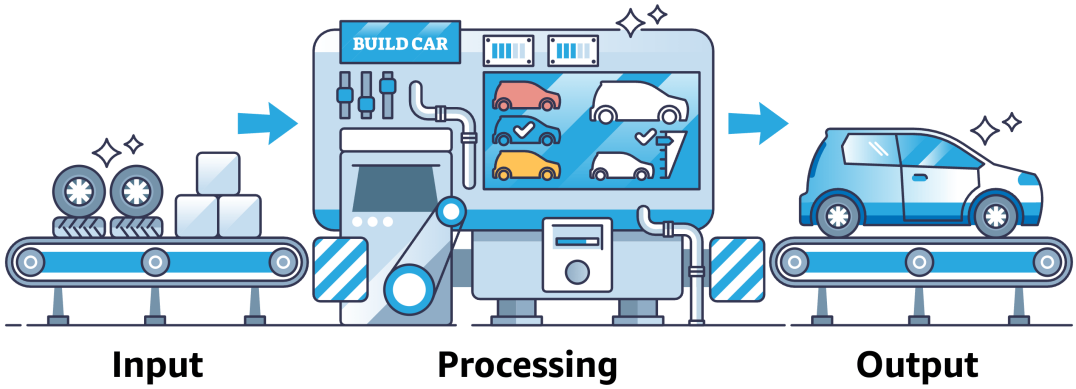
   Optionale Schritte: Netzwerk, Datenbanken, Skalierungsparameter, erweiterte Konfiguration für Updates, Überwachung und Protokollierung.
3. Wählen Sie auf der Seite „Überprüfen“, auf der eine Zusammenfassung Ihrer Optionen angezeigt wird, „Senden“ aus.

> **Herzlichen Glückwunsch!**
>
> Sie haben eine Anwendung erstellt und eine Umgebung konfiguriert! Jetzt müssen Sie warten, bis die Ressourcen bereitgestellt sind.

Schritt 2 — Stellen Sie Ihre Anwendung bereit

Wenn Sie eine Anwendung erstellen, richtet Elastic Beanstalk die Umgebungen für Sie ein. Sie müssen sich nur zurücklehnen und warten.

Die erste Bereitstellung kann bis zu fünf Minuten dauern, bis die Ressourcen erstellt sind. Aktualisierungen nehmen weniger Zeit in Anspruch, da nur Änderungen in Ihrem Stack bereitgestellt werden.



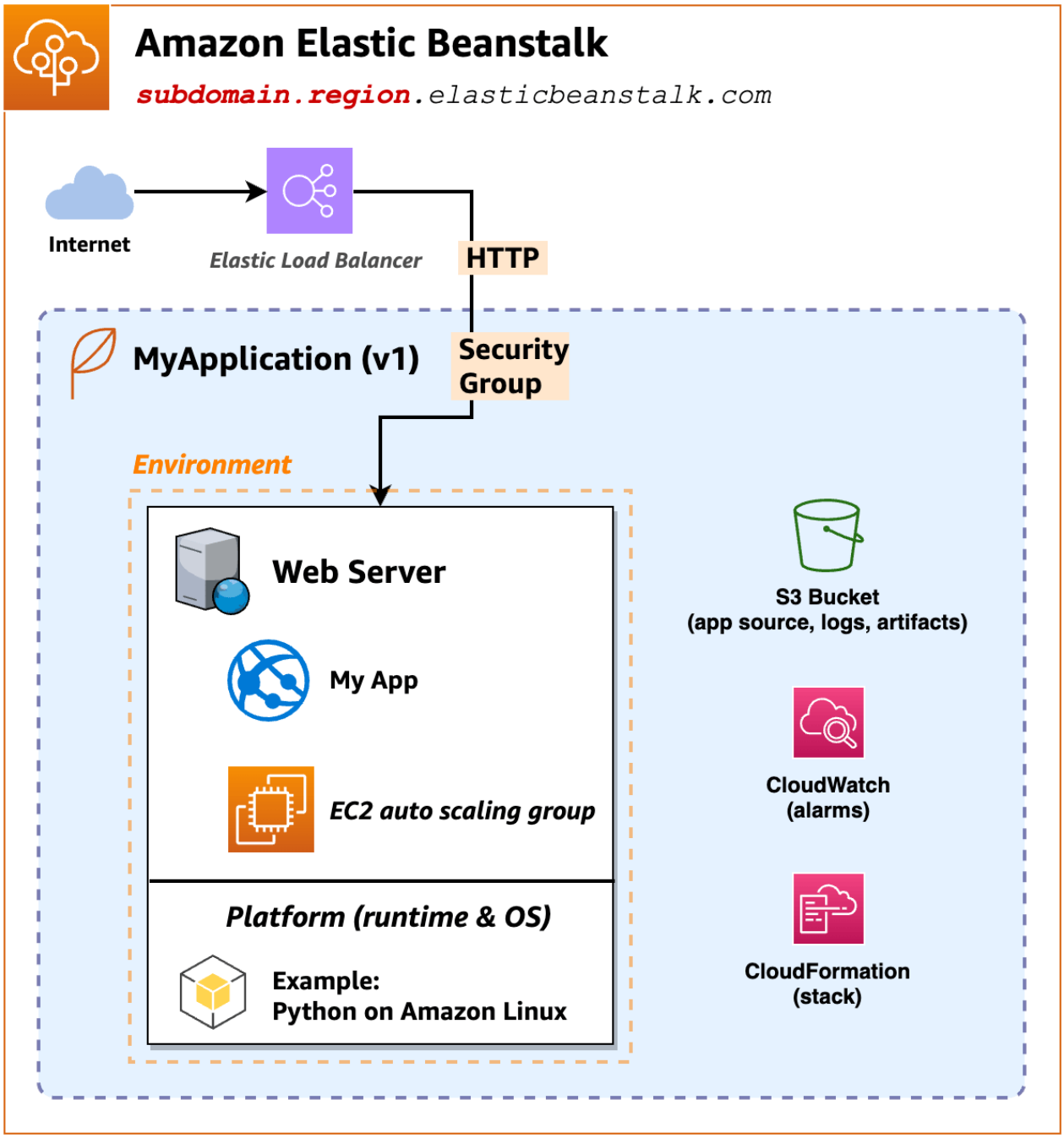
Wenn Sie die Beispielanwendung erstellen, erstellt Elastic Beanstalk die folgenden -Ressourcen:

- EC2-Instance — Eine virtuelle Amazon EC2-Maschine, die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Auf jeder Plattform wird ein anderer Satz von Software, Konfigurationsdateien und Skripten ausgeführt, um eine bestimmte Sprachversion, ein Framework, einen Webcontainer oder eine Kombination davon zu unterstützen. Die meisten Plattformen verwenden entweder Apache oder Nginx als Reverse-Proxy, um Web-Traffic an Ihre Web-App weiterzuleiten, statische Ressourcen bereitzustellen und Zugriffs- und Fehlerprotokolle zu generieren. Sie können eine Verbindung zu Ihren Amazon EC2 EC2-Instances herstellen, um die Konfiguration und die Protokolle einzusehen.
- Instance-Sicherheitsgruppe — Eine Amazon EC2-Sicherheitsgruppe wird erstellt, um eingehende Anfragen auf Port 80 zuzulassen, sodass eingehender Datenverkehr auf einem Load Balancer Ihre Web-App erreichen kann.
- Amazon S3 S3-Bucket — Ein Speicherort für Ihren Quellcode, Ihre Protokolle und andere Artefakte.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme werden erstellt, um die Auslastung Ihrer Instances zu überwachen und sie nach Bedarf zu erhöhen oder zu reduzieren.
- CloudFormation Stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung bereitzustellen und Konfigurationsänderungen vorzunehmen. Sie können die Vorlage für die Ressourcendefinition in der CloudFormation Konsole einsehen.

- Domainname — Ein Domainname, der im folgenden Format an Ihre Web-App weitergeleitet wird:*subdomain*.*region*.elasticbeanstalk.com.

Elastic Beanstalk erstellt Ihre Anwendung, startet eine Umgebung, erstellt eine Anwendungsversion und stellt dann Ihren Code in der Umgebung bereit. Während des Vorgangs verfolgt die Konsole den Fortschritt und zeigt den Ereignisstatus auf der Registerkarte Ereignisse an.



Nachdem alle Ressourcen bereitgestellt wurden, sollte sich der Zustand der Umgebung auf ändern. **0k**

> Ihre Bewerbung ist fertig!
>
> Wenn Sie sehen, dass sich der Zustand Ihrer Anwendung auf geändert hat**0k**, können Sie die Website Ihrer Webanwendung aufrufen.

Schritt 3 — Erkunden Sie die Elastic Beanstalk Beanstalk-Umgebung

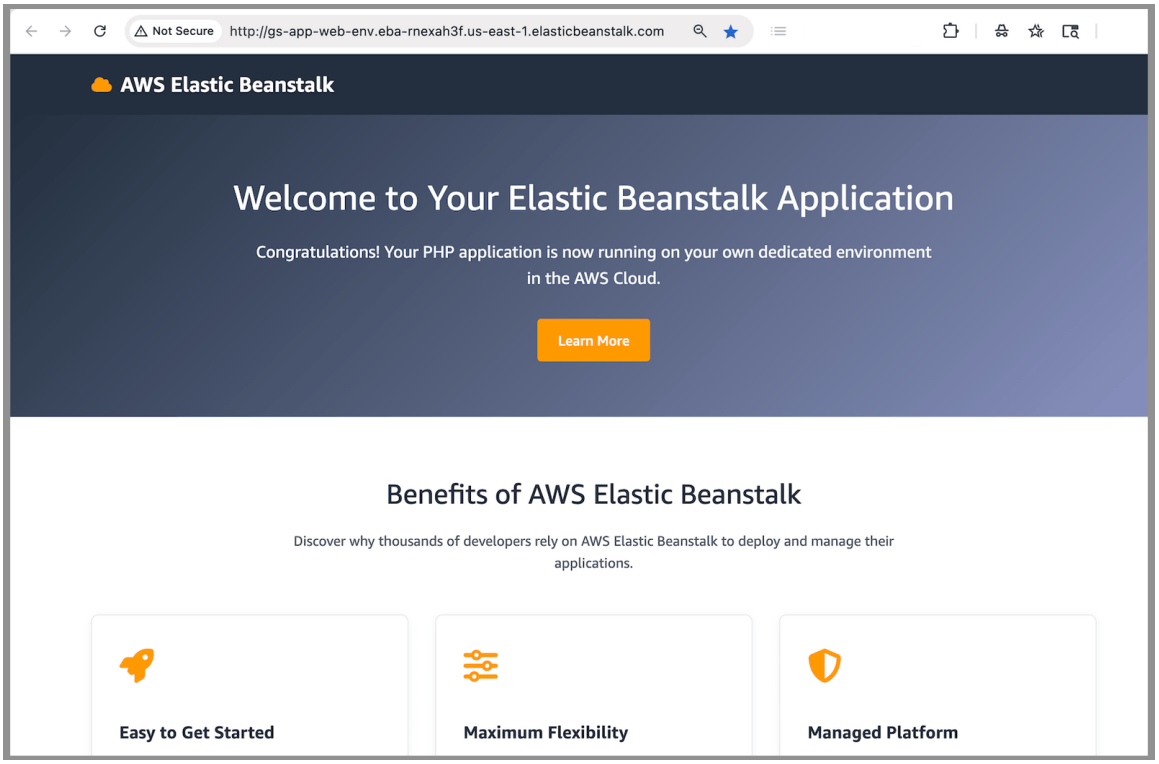
Sie beginnen mit der Erkundung Ihrer bereitgestellten Anwendungsumgebung auf der Seite mit der Umgebungsübersicht in der Konsole.

Um sich die Umgebung und Ihre Anwendung anzusehen

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie Gehe zur Umgebung, um Ihre Anwendung zu durchsuchen!

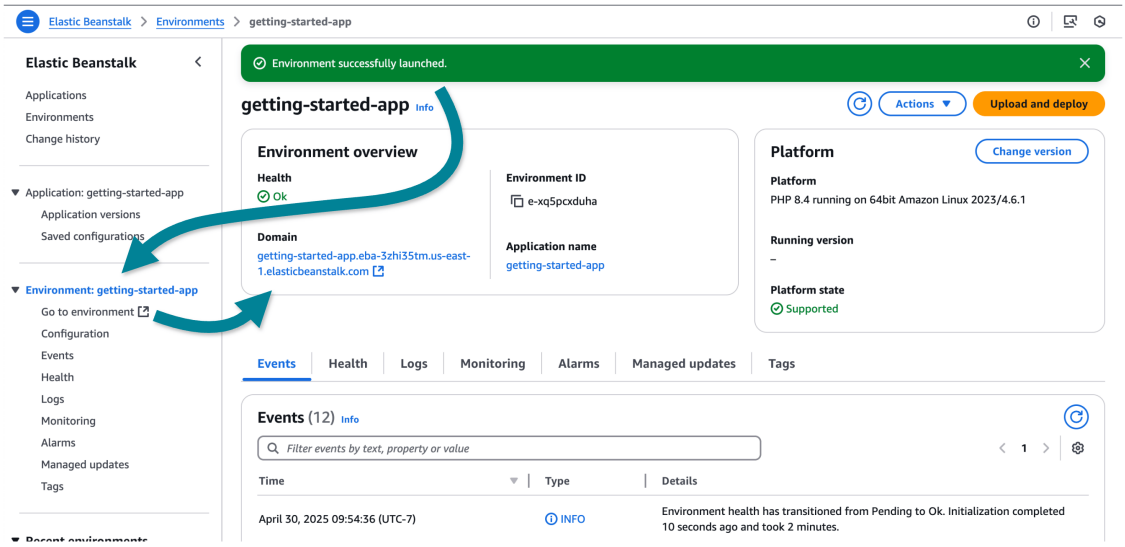
   (Sie können auch den URL-Link wählen, der für Domain aufgeführt ist, um Ihre Anwendung zu durchsuchen.)

   Die Verbindung erfolgt über HTTP (nicht HTTPS), sodass in Ihrem Browser möglicherweise eine Warnung angezeigt wird.



Zurück in der Elastic Beanstalk Beanstalk-Konsole wird im oberen Bereich die Umgebungsübersicht mit Informationen auf oberster Ebene zu Ihrer Umgebung angezeigt, darunter Name, Domain-URL, aktueller Integritätsstatus, laufende Version und die Plattform, auf der die Anwendung ausgeführt wird. Die laufende Version und Plattform sind für die Fehlerbehebung Ihrer aktuell bereitgestellten Anwendung unerlässlich.

Nach dem Übersichtsbereich sehen Sie auf der Registerkarte Ereignisse die aktuellen Umgebungsaktivitäten.



Während Elastic Beanstalk Ihre AWS Ressourcen erstellt und Ihre Anwendung startet, befindet sich die Umgebung in einem `Pending` Zustand. Statusmeldungen über Startereignisse werden kontinuierlich zur Liste der Ereignisse hinzugefügt.

Die Domäne der Umgebung ist die URL für Ihre bereitgestellte Webanwendung. Im linken Navigationsbereich gelangen Sie mit Gehe zur Umgebung auch zu Ihrer Domain. In ähnlicher Weise enthält der linke Navigationsbereich Links, die den verschiedenen Tabs entsprechen.

Beachten Sie den Link Konfiguration im linken Navigationsbereich, der eine Zusammenfassung der Werte der Umgebungskonfigurationsoptionen, gruppiert nach Kategorien, anzeigt.

> **Einstellungen für die Umgebungskonfiguration**
>
> Beachten Sie den Link Konfiguration im linken Navigationsbereich. Sie können detaillierte Umgebungseinstellungen wie Servicerollen, Netzwerke, Datenbanken, Skalierung, verwaltete Plattformaktualisierungen, Speicher, Systemüberwachung, fortlaufende Bereitstellung, Protokollierung und mehr anzeigen und bearbeiten!

Die verschiedenen Registerkarten enthalten detaillierte Informationen zu Ihrer Umgebung:

- Ereignisse — Zeigt eine aktuelle Liste mit Informationen und Fehlermeldungen des Elastic Beanstalk-Service und anderer Services für Ressourcen in Ihrer Umgebung an.
- Zustand — Zeigt den Status und detaillierte Gesundheitsinformationen für die Amazon EC2 EC2-Instances an, auf denen Ihre Anwendung ausgeführt wird.
- Logs (Protokolle) – Rufen Sie Protokolle von Amazon EC2 in Ihrer Umgebung ab und laden Sie sie herunter. Sie können vollständige Protokolle oder aktuelle Aktivitäten abrufen. Die abgerufenen Protokolle sind 15 Minuten lang verfügbar.
- Überwachung — Zeigt Statistiken für die Umgebung an, z. B. durchschnittliche Latenz und CPU-Auslastung.
- Alarme — Alarme, die für Umgebungsmetriken konfiguriert sind, anzeigen und bearbeiten.
- Verwaltete Updates — Hier finden Sie Informationen zu bevorstehenden und abgeschlossenen Updates der verwalteten Plattform und zum Austausch von Instanzen.
- Tags — Zeigen Sie Schlüssel-Wert-Paare an und bearbeiten Sie sie, die auf Ihre Umgebung angewendet werden.

> **Note**
>
> Links im Navigationsbereich der Konsole zeigen die entsprechende Registerkarte an.

Fehlerbehebung mit Protokollen

Um unerwartetes Verhalten zu beheben oder Bereitstellungen zu debuggen, sollten Sie die Protokolle in Ihren Umgebungen überprüfen.

Sie können 100 Zeilen aller Protokolldateien auf der Registerkarte Logs in der Elastic Beanstalk Beanstalk-Konsole anfordern. Alternativ können Sie sich direkt mit der Amazon EC2 EC2-Instance verbinden und die Protokolle in Echtzeit verfolgen.

Um die Logs anzufordern (Elastic Beanstalk Beanstalk-Konsole)

1. Navigieren Sie in der Elastic Beanstalk Beanstalk-Konsole zu Ihrer Umgebung.
2. Wählen Sie den Tab Logs oder klicken Sie mit der linken Navigationsleiste auf Logs anfordern.
3. Wählen Sie Letzte 100 Zeilen aus.
4. Nachdem die Protokolle erstellt wurden, wählen Sie den Link Herunterladen, um die Protokolle im Browser anzuzeigen.

Suchen Sie in den Protokollen nach dem Protokoll und notieren Sie sich das Verzeichnis für das Nginx-Zugriffsprotokoll.

Fügen Sie eine Richtlinie hinzu, um Verbindungen zu Amazon EC2 zu ermöglichen

Bevor Sie eine Verbindung herstellen können, müssen Sie eine Richtlinie hinzufügen, die Verbindungen zu Amazon EC2 mit Session Manager ermöglicht.

1. Navigieren Sie zur IAM-Konsole.
2. Suchen Sie die **`aws-elasticbeanstalk-ec2-role`** Rolle und wählen Sie sie aus.
3. Wählen Sie „Erlaubnis hinzufügen“ und dann „Richtlinien anhängen“.
4. Suchen Sie nach einer Standardrichtlinie, die mit dem folgenden Text beginnt:**`AmazonSSMManagedEC2Instance`**, und fügen Sie sie dann der Rolle hinzu.

So stellen Sie mit Session Manager eine Verbindung zu Ihrem Amazon EC2 her

1. Navigieren Sie zur Amazon-EC2-Konsole.
2. Wählen Sie Instances und dann Ihre `gs-app-web-env` Instance aus.
3. Wählen Sie Connect und dann Session Manager.
4. Wählen Sie Connect aus.

Nachdem Sie eine Verbindung zur Instanz hergestellt haben, starten Sie eine Bash-Shell und verfolgen Sie die Protokolle:

1. Führen Sie den Befehl **`bash`** aus.
2. Führen Sie den Befehl **`cd /var/log/nginx`** aus.
3. Führen Sie den Befehl **`tail -f access.log`** aus.
4. Rufen Sie in Ihrem Browser die URL der Anwendungsdomäne auf. Aktualisieren.

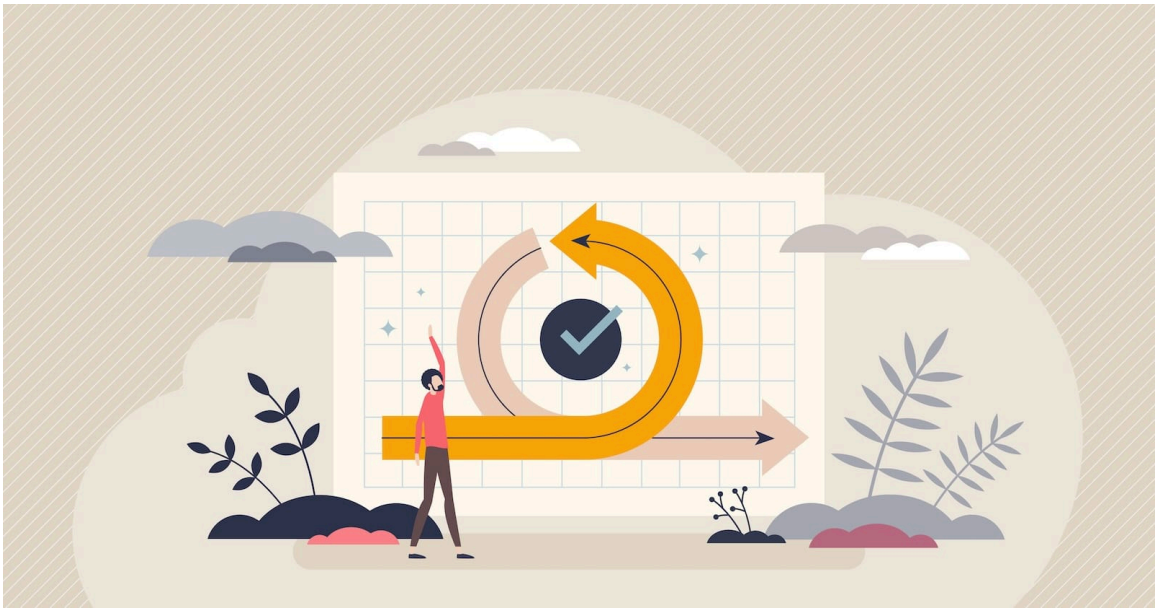
> Herzlichen Glückwunsch, Sie sind verbunden!
>
> Sie sollten jedes Mal, wenn Sie die Seite aktualisieren, Protokolleinträge in Ihrem Instance-Update sehen.

> Die Verbindungstaste funktioniert nicht?
>
> Wenn die Verbindungsschaltfläche nicht verfügbar ist, kehren Sie zu IAM zurück und überprüfen Sie, ob Sie der Rolle die erforderliche Richtlinie hinzugefügt haben.

Schritt 4 — Aktualisieren Sie Ihre Anwendung

Schließlich werden Sie Ihre Anwendung aktualisieren wollen. Sie können jederzeit eine neue Version bereitstellen, sofern nicht andere Aktualisierungsvorgänge in der Umgebung ausgeführt werden.

Die Anwendungsversion, mit der Sie dieses Tutorial gestartet haben, heißt Beispielanwendung.



So aktualisieren Sie die Anwendungsversion

1. Laden Sie die folgende PHP-Beispielanwendung herunter:

   PHP — php-v2.zip

2. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
3. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
4. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
5. Wählen Sie Datei auswählen aus und laden Sie dann das Quellpaket für die Beispielanwendung hoch, das Sie heruntergeladen haben.

   Die Konsole füllt das Versionslabel automatisch mit einer neuen eindeutigen Bezeichnung aus und erhöht dabei automatisch eine nachgestellte Ganzzahl. Wenn Sie Ihr eigenes Versionslabel wählen, stellen Sie sicher, dass es eindeutig ist.
6. Wählen Sie Bereitstellen.

Während Elastic Beanstalk Ihre Datei auf Ihren Amazon EC2 EC2-Instances bereitstellt, können Sie den Bereitstellungsstatus auf der Umgebungsübersichtsseite einsehen. Während die Anwendungsversion aktualisiert wird, ist der Gesundheitsstatus der Umgebung grau. Nach

abgeschlossener Bereitstellung führt Elastic Beanstalk eine Zustandsprüfung für die Anwendung durch. Wenn die Anwendung auf die Zustandsprüfung reagiert, wird sie als fehlerfrei betrachtet und der Status ändert sich in grün. Die Umgebungs-Übersicht zeigt die neue Running Version (Ausgeführte Version) an – den Namen, den Sie als Version label (Versionsbezeichnung) angegeben haben.

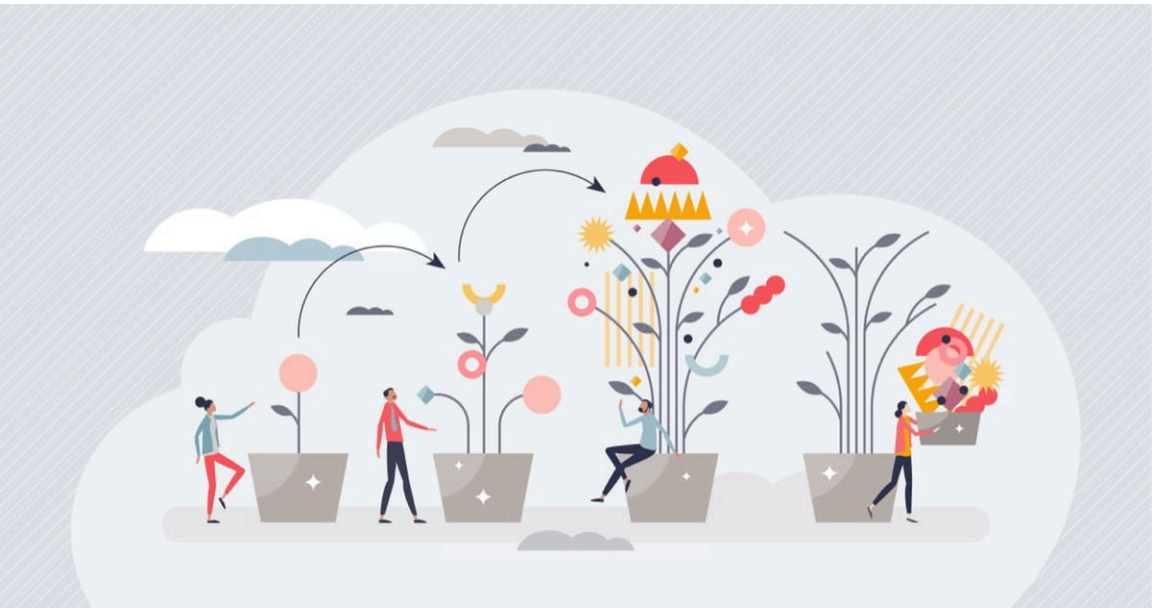
Elastic Beanstalk lädt auch Ihre neue Anwendungsversion hoch und fügt sie der Tabelle der Anwendungsversionen hinzu. Wählen Sie im Navigationsbereich unter getting-started-appAnwendungsversionen aus, um die Tabelle anzuzeigen.

> **Aktualisierung erfolgreich!**
>
> Nach dem Aktualisieren Ihres Browsers sollte eine aktualisierte „v2“ -Meldung angezeigt werden.
> Wenn Sie die Quelle selbst bearbeiten möchten, entpacken Sie das Quellpaket, bearbeiten Sie es und komprimieren Sie es dann erneut. Verwenden Sie auf macOS den folgenden Befehl in Ihrem PHP-Verzeichnis mit dem -X, um zusätzliche Dateiattribute auszuschließen:
> `zip -X -r ../php-v2.zip .`

Schritt 5 — Skalieren Sie Ihre Anwendung

Sie können Ihre Umgebung konfigurieren, damit sie besser auf die Anwendung ausgerichtet ist. Beispielsweise können Sie bei einer rechenintensiven Anwendung den Amazon Elastic Compute Cloud (Amazon EC2)-Instance-Typ für die Anwendungsausführung ändern. Um Konfigurationsänderungen anzuwenden, führt Elastic Beanstalk eine Umgebungsaktualisierung durch.

Einige Konfigurationsänderungen lassen sich einfach und schnell durchführen. Bei einigen Änderungen müssen AWS Ressourcen gelöscht und neu erstellt werden, was mehrere Minuten dauern kann. Wenn Sie die Konfigurationseinstellungen ändern, warnt Elastic Beanstalk Sie über potenzielle Anwendungsausfallzeiten.



Erhöhen Sie die Kapazitätseinstellungen

In diesem Beispiel einer Konfigurationsänderung bearbeiten Sie die Kapazitätseinstellungen Ihrer Umgebung. Sie konfigurieren eine skalierbare Umgebung mit Lastenausgleich, die zwischen zwei und vier Amazon EC2-Instances in ihrer Auto Scaling-Gruppe hat. Dann überprüfen Sie, ob die Änderung durchgeführt wurde. Elastic Beanstalk erstellt eine zusätzliche Amazon EC2-Instance, die der einzelnen Instance hinzugefügt wird, die ursprünglich erstellt wurde. Anschließend ordnet Elastic Beanstalk beide Instances dem Load Balancer der Umgebung zu. Dadurch wird die Reaktionsfähigkeit Ihrer Anwendung verbessert und die Verfügbarkeit erhöht.

So ändern Sie die Kapazität Ihrer Umgebung

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Instance traffic and scaling (Instance, Datenverkehr und Skalierung) die Option Edit (Bearbeiten) aus.
5. Blenden Sie den Abschnitt Instances (Instances) aus, damit Sie den Abschnitt Capacity (Kapazität) leichter sehen können. Ändern Sie unter Auto Scaling Group (Auto-Scaling-Gruppe) die Option Environment type (Umgebungstyp) zu Load balanced (Load Balancer).
6. Ändern Sie in der Zeile Instances die Werte Min in **2** und Max in. **4**

7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

   Wenn Sie gewarnt werden, dass das Update alle Ihre aktuellen Instances ersetzen wird. Wählen Sie Bestätigen.

Das Aktualisieren der Umgebung kann einige Minuten dauern. In der Liste der Ereignisse sollten mehrere Aktualisierungen angezeigt werden. Halten Sie Ausschau nach dem Ereignis Die neue Konfiguration wurde erfolgreich in der Umgebung bereitgestellt.

Überprüfen Sie die erhöhte Kapazität

Nachdem das Umgebungsupdate abgeschlossen und die Umgebung bereit ist, hat Elastic Beanstalk automatisch eine zweite Instance gestartet, um Ihre neue Mindestkapazitätseinstellung zu erfüllen.

So überprüfen Sie die erhöhte Kapazität

1. Wählen Sie Health entweder in der Tab-Liste oder im linken Navigationsbereich aus.
2. Lesen Sie den Abschnitt „Verbesserter Instanzstatus“.

> Sie haben gerade hochskaliert!
>
> Mit zwei Amazon EC2 EC2-Instances hat sich die Kapazität Ihrer Umgebung verdoppelt, und das hat nur wenige Minuten gedauert.

Aufräumen Ihrer Elastic Beanstalk Beanstalk-Umgebung

Um sicherzustellen, dass Ihnen keine Dienste in Rechnung gestellt werden, die Sie nicht nutzen, löschen Sie alle Anwendungsversionen und beenden Sie Umgebungen. Dadurch werden auch die AWS Ressourcen gelöscht, die die Umgebung für Sie erstellt hat.



So löschen Sie die Anwendung und alle zugehörigen Ressourcen

1. Alle Anwendungsversionen löschen.
   a. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
   b. Wählen Sie im Navigationsbereich Applications und anschließend. getting-started-app
   c. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen)aus.
   d. Wählen Sie auf der Seite Application versions (Anwendungsversionen) alle zu löschenden Anwendungsversionen aus.
   e. Wählen Sie Aktionen und anschließend Löschen aus.
   f. Aktivieren Sie Delete versions from Amazon S3 (Versionen von Amazon S3 löschen).
   g. Wählen Sie Delete (Löschen) und dann Done (Fertig).
2. Die Umgebung beenden.
   a. Wählen Sie getting-started-appim Navigationsbereich die Option und anschließend in der Umgebungsliste die Option GettingStartedApp-env aus.

b. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).

c. Bestätigen Sie, dass Sie GettingStartedApp-env beenden möchten, indem Sie den Umgebungsnamen eingeben, und wählen Sie dann Terminate aus.

3. Löschen Sie die getting-started-app-Anwendung.

a. Wählen Sie im Navigationsbereich die. getting-started-app

b. Wählen Sie Actions (Aktionen) und anschließend Delete Application (Anwendung löschen).

c. Bestätigen Sie, dass Sie löschen möchten, getting-started-appindem Sie den Namen der Anwendung eingeben, und wählen Sie dann Löschen aus.

Herzlichen Glückwunsch!

Sie haben erfolgreich eine Beispielanwendung in der AWS Cloud bereitgestellt, eine neue Version hochgeladen, ihre Konfiguration geändert, um eine zweite Auto Scaling Scaling-Instanz hinzuzufügen, und Ihre AWS Ressourcen bereinigt!

Nächste Schritte

Um zu erfahren, wie Sie das **eb** Befehlszeilentool verwenden können, um die Bereitstellung Ihres Codes in Elastic Beanstalk zu automatisieren, empfehlen wir Ihnen, mit dem fortzufahren. the section called “QuickStart für PHP”

Als Nächstes möchten Sie vielleicht nachlesen, wie Sie eine HTTPS-Verbindung einrichten, siehe. the section called “HTTPS”

Einrichtung der EB-Befehlszeilenschnittstelle (EB CLI) zur Verwaltung von Elastic Beanstalk

Die EB CLI ist eine Befehlszeilenschnittstelle, die interaktive Befehle zum Erstellen, Aktualisieren und Überwachen von Umgebungen bereitstellt AWS Elastic Beanstalk. Das EB CLI Open-Source-Projekt ist auf Github: aws/aws-elastic-beanstalk-cli

Nachdem Sie die EB-CLI installiert und ein Projektverzeichnis konfiguriert haben, können Sie Umgebungen mit nur einem Befehl erstellen:

```
$ eb create my-beanstalk-environment
```

Wir empfehlen die Installation mit dem Setup-Skript. Wie das geht, erfahren Sie unterthe section called “Installieren Sie EB CLI”.

Das AWS CLI bietet direkten Zugriff auf Elastic APIs Beanstalk auf niedriger Ebene. Es ist zwar leistungsstark, aber auch ausführlich und wird der EB-CLI weniger bevorzugt. Zum Erstellen einer Umgebung mit dem AWS CLI sind beispielsweise die folgenden Befehle erforderlich:

```
$ aws elasticbeanstalk check-dns-availability \
    --cname-prefix my-cname
$ aws elasticbeanstalk create-application-version \
    --application-name my-application \
    --version-label v1 \
    --source-bundle S3Bucket=amzn-s3-demo-bucket,S3Key=php-proxy-sample.zip
$ aws elasticbeanstalk create-environment \
    --cname-prefix my-cname \
    --application-name my-app \
    --version-label v1 \
    --environment-name my-env \
    --solution-stack-name "64bit Amazon Linux 2023 v4.5.0 running Ruby 3.4"
```

EB CLI mit Setup-Skript installieren (empfohlen)

> **Wir empfehlen das Installationsskript**
>
> Wir empfehlen, das Installationsskript zu verwenden, um die EB-CLI und ihre Abhängigkeiten einzurichten und mögliche Konflikte mit anderen Python-Paketen zu vermeiden.

Voraussetzungen: Git, Python und virtualenv

Um das Installationsskript herunterzuladen und zu verwenden

1. Klonen Sie das Repository

   ```
   git clone https://github.com/aws/aws-elastic-beanstalk-cli-setup.git
   ```

2. Installieren oder aktualisieren Sie die EB-CLI.

   macOS//Linux in Bash oder Zsh

   ```
   python ./aws-elastic-beanstalk-cli-setup/scripts/ebcli_installer.py
   ```

   Windows im PowerShell oder Befehlsfenster

   ```
   python .\aws-elastic-beanstalk-cli-setup\scripts\ebcli_installer.py
   ```

3. Überprüfen Sie, ob die EB CLI ordnungsgemäß installiert wurde.

   ```
   $ eb --version
   EB CLI 3.21.0 (Python 3.12)
   ```

Vollständige Installationsanweisungen finden Sie im aws/aws-elastic-beanstalk-cli-setupRepository unter GitHub.

Manuelle Installation der EB-CLI

Sie können die EB-CLI unter Linux, macOS und Windows mit dem `pip` Paketmanager für Python installieren, der die Installation, Aktualisierung und Entfernung von Python-Paketen und deren Abhängigkeiten ermöglicht.

> **Wir empfehlen das Installationsskript**
>
> Wir empfehlen die Verwendung vonInstallieren Sie EB CLI, um die EB-CLI einzurichten und Abhängigkeitskonflikte zu vermeiden.

Voraussetzung: Sie müssen eine unterstützte Version von Python installiert haben. Sie können es von der Python-Download-Seite auf der Python-Website herunterladen.

So installieren Sie die EB-CLI (manuell)

1. Führen Sie den folgenden Befehl aus.

   ```
   $ pip install awsebcli --upgrade --user
   ```

   Die Option `--upgrade` weist `pip` an, alle bereits installierten Anforderungen zu aktualisieren. Die Option `--user` bewirkt, dass `pip` das Programm in ein Unterverzeichnis Ihres Benutzerverzeichnisses installiert, damit keine vom Betriebssystem verwendeten Bibliotheken geändert werden.

   > **Beheben von -Problemen**
   >
   > Falls bei der Installation der EB CLI mit `pip` Probleme auftreten, können Sie die EB CLI in einer virtuellen Umgebung installieren, um das Tool und seine Abhängigkeiten zu isolieren, oder eine Python-Version verwenden, die Sie normalerweise nicht nutzen.

2. Fügen Sie den Pfad der ausführbaren Datei zu Ihrer `PATH`-Variable hinzu:

   - Unter Linux und macOS:

     Linux – `~/.local/bin`

     macOS – `~/Library/Python/3.12/bin`

So ändern Sie die PATH-Variable (Linux, Unix oder macOS):

a. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`.
- Zsh – `.zshrc`
- Tcsh – `.tcshrc`, `.cshrc` oder `.login`.

b. Fügen Sie dem Profilskript einen Exportbefehl hinzu. Im folgenden Beispiel wird der Pfad, der durch dargestellt wird*LOCAL_PATH*, zur aktuellen PATH Variablen hinzugefügt.

```
export PATH=LOCAL_PATH:$PATH
```

c. Laden Sie das im ersten Schritt beschriebene Profilskript in die aktuelle Sitzung. Im folgenden Beispiel wird das Profilskript geladen, das durch dargestellt wird*PROFILE_SCRIPT*.

```
$ source ~/PROFILE_SCRIPT
```

- Unter Windows:

  Python 3.12 — `%USERPROFILE%\AppData\Roaming\Python\Python312\Scripts`

  Python frühere Versionen – `%USERPROFILE%\AppData\Roaming\Python\Scripts`

  So ändern Sie Ihre PATH-Variable (Windows):

  a. Betätigen Sie die Windows-Taste und geben Sie anschließend ei **`environment variables`**.

  b. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).

  c. Wählen Sie PATH (PFAD) und Edit (Bearbeiten) aus.

  d. Fügen Sie im Feld Variable value (Variablenwert) Pfade hinzu, getrennt durch Semikola. Beispiel: *`C:\item1\path`***`;C:\item2\path`**

e. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.

f. Schließen Sie alle laufenden Eingabeaufforderungsfenster und öffnen Sie dann erneut ein Eingabeaufforderungsfenster.

3. Überprüfen Sie, ob die EB CLI korrekt installiert wurde, indem Sie eb --version ausführen.

```
$ eb --version
EB CLI 3.21.0 (Python 3.12)
```

Durch regelmäßige Aktualisierungen der EB CLI werden Funktionalitäten zur Unterstützung der neuesten Elastic Beanstalk-Funktionen hinzugefügt. Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version der EB CLI zu aktualisieren.

```
$ pip install awsebcli --upgrade --user
```

Wenn Sie die EB CLI deinstallieren müssen, verwenden Sie `pip uninstall`.

```
$ pip uninstall awsebcli
```

Installieren der EB-CLI in einer virtuellen Umgebung

Sie können Versionsanforderungskonflikte mit anderen `pip`-Paketen vermeiden, indem Sie die EB CLI in einer virtuellen Umgebung installieren.

So installieren Sie die EB CLI in einer virtuellen Umgebung:

1. Installieren Sie zuerst mit. `virtualenv pip`

```
$ pip install --user virtualenv
```

2. Erstellen Sie eine virtuelle Umgebung.

```
$ virtualenv ~/eb-ve
```

Mithilfe der Option `-p` können Sie eine andere als die standardmäßige ausführbare Python-Datei verwenden.

```
$ virtualenv -p /usr/bin/python3.12 ~/eb-ve
```

3. Aktivieren Sie die virtuelle Umgebung.

   Linux, Unix oder macOS

   ```
   $ source ~/eb-ve/bin/activate
   ```

   Windows

   ```
   $ %USERPROFILE%\eb-ve\Scripts\activate
   ```

4. Installieren Sie die EB CLI.

   ```
   (eb-ve)$ pip install awsebcli --upgrade
   ```

5. Überprüfen Sie, ob die EB CLI ordnungsgemäß installiert wurde.

   ```
   $ eb --version
   EB CLI 3.23.0 (Python 3.12)
   ```

Sie können den Befehl `deactivate` verwenden, um die virtuelle Umgebung zu beenden. Führen Sie den Aktivierungsbefehl bei jedem Start einer neuen Sitzung erneut aus.

Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version zu aktualisieren.

```
(eb-ve)$ pip install awsebcli --upgrade
```

Installieren der EB-CLI mit Homebrew

In der Regel ist die neueste EB CLI-Version für `Homebrew` einige Tage nach der Veröffentlichung für `pip` verfügbar.

> ⚠ Wir empfehlen das Installationsskript
>
> Wir empfehlen die Verwendung von Installieren Sie EB CLI, um die EB-CLI einzurichten und Abhängigkeitskonflikte zu vermeiden.

So installieren Sie die EB CLI mit **`Homebrew`**:

1. Stellen Sie sicher, dass Sie über die neueste Version von `Homebrew` verfügen.

```
$ brew update
```

2. Führen Sie `brew install awsebcli`.

```
$ brew install awsebcli
```

3. Überprüfen Sie, ob die EB CLI ordnungsgemäß installiert wurde.

```
$ eb --version
EB CLI 3.21.0 (Python 3.12)
```

Konfigurieren der EB CLI

Nach der Installation der EB-CLI können Sie Ihr Projektverzeichnis und die EB-CLI konfigurieren, indem Sie Folgendes eb init ausführen. Das folgende Beispiel zeigt die Konfigurationsschritte eb init bei der ersten Ausführung in einem Projektordner mit dem Name`eb`.

So initialisieren Sie ein EB CLI-Projekt

1. Zunächst werden Sie von der EB CLI aufgefordert, eine Region auszuwählen. Wählen Sie Ihre bevorzugte Region.

```
~/eb $ eb init
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
(default is 3): 3
```

2. Wenn Sie dazu aufgefordert werden, geben Sie Ihren Zugriffsschlüssel und Ihren geheimen Schlüssel ein, damit die EB CLI Ressourcen für Sie verwalten kann. Zugriffsschlüssel werden in

der AWS Identity and Access Management Konsole erstellt. Wenn Sie keine Schlüssel haben, finden Sie weitere Informationen unter Wie erhalte ich Sicherheitsanmeldeinformationen? im Allgemeine Amazon Web Services-Referenz.

```
You have not yet set up your credentials or your credentials are incorrect.
You must provide your credentials.
(aws-access-id): AKIAJOUAASEXAMPLE
(aws-secret-key): 5ZRIrtTM4ciIAvd4EXAMPLEDtm+PiPSzpoK
```

3. Eine Elastic Beanstalk Beanstalk-Anwendung ist eine Ressource, die eine Reihe von Anwendungsversionen (Quelle), Umgebungen und gespeicherten Konfigurationen enthält, die einer einzelnen Webanwendung zugeordnet sind. Jedes Mal, wenn Sie Ihren Quellcode mit der EB CLI in Elastic Beanstalk bereitstellen, wird eine neue Anwendungsversion erstellt und der Liste hinzugefügt.

```
Select an application to use
1) [ Create new Application ]
(default is 1): 1
```

4. Der Standard-Anwendungsname ist der Name des Ordners, in dem Sie eb init ausführen. Geben Sie einen beliebigen Namen ein, der Ihr Projekt beschreibt.

```
Enter Application Name
(default is "eb"): eb
Application eb has been created.
```

5. Wählen Sie eine Plattform, die mit der Sprache oder dem Framework, in der bzw. in dem Ihre Webanwendung entwickelt wurde, übereinstimmt. Wenn Sie noch keine Anwendung entwickelt haben, wählen Sie eine Plattform, die Sie interessiert. Wir zeigen Ihnen in Kürze, wie Sie eine Beispielanwendung starten. Sie können diese Einstellung zu einem späteren Zeitpunkt jederzeit ändern.

```
Select a platform.
1) .NET Core on Linux
2) .NET on Windows Server
3) Docker
4) Go
5) Java
6) Node.js
7) PHP         <== select platform by number
8) Packer
```

```
9) Python
10) Ruby
11) Tomcat
(make a selection):7
```

6. Wählen Sie einen bestimmten Plattformzweig aus.

```
Select a platform branch.
1) PHP 8.4 running on 64bit Amazon Linux 2023
2) PHP 8.3 running on 64bit Amazon Linux 2023
3) PHP 8.2 running on 64bit Amazon Linux 2023
4) PHP 8.1 running on 64bit Amazon Linux 2023
5) PHP 8.1 running on 64bit Amazon Linux 2
(default is 1):1
```

7. Wählen Sie Yes (Ja), um den Instances in Ihrer Elastic Beanstalk-Umgebung ein SSH-Schlüsselpaar zuzuweisen. Auf diese Weise können Sie eine direkte Verbindung für die Fehlerbehebung herstellen.

```
Do you want to set up SSH for your instances?
(y/n): y
```

8. Wählen Sie ein vorhandenes Schlüsselpaar oder erstellen Sie ein neues. Damit mit eb init ein neues Schlüsselpaar erstellt werden kann, müssen Sie ssh-keygen auf Ihrem lokalen Computer installiert haben und über die Befehlszeile darauf zugreifen können. Die EB CLI registriert das neue key pair EC2 für Sie bei Amazon und speichert den privaten Schlüssel lokal in einem Ordner mit dem Namen `.ssh` in Ihrem Benutzerverzeichnis.

```
Select a keypair.
1) [ Create new KeyPair ]
(default is 1): 1
```

Ihre EB CLI-Installation ist nun konfiguriert und einsatzbereit.

Erweiterte Konfiguration

- Ignorieren von Dateien, die .ebignore verwenden
- Verwenden von benannten Profilen
- Bereitstellen eines Artefakts anstelle des Projektordners
- Konfigurationseinstellungen und Vorrang

- Instance-Metadaten

Ignorieren von Dateien, die .ebignore verwenden

Sie können die EB CLI anweisen, bestimmte Dateien in Ihrem Projektverzeichnis zu ignorieren, indem Sie dem Verzeichnis die Datei `.ebignore` hinzufügen. Diese Datei funktioniert wie eine `.gitignore`-Datei. Bei der Bereitstellung Ihres Projektverzeichnisses für Elastic Beanstalk und der Erstellung einer neuen Anwendungsversion schließt die EB CLI keine Dateien ein, die als `.ebignore` im von ihr erstellten Quell-Bundle angegeben werden.

Wenn keine `.ebignore` vorhanden ist, jedoch eine `.gitignore`, ignoriert die EB CLI die Dateien, die in `.gitignore` angegeben sind. Wenn `.ebignore` vorhanden ist, liest die EB CLI `.gitignore` nicht.

Wenn `.ebignore` vorhanden ist, verwendet die EB CLI keine git-Befehle zum Erstellen Ihres Quell-Bundles. Das bedeutet, EB-CLI ignoriert die in `.ebignore` angegebenen Dateien und bindet alle anderen Dateien ein. Insbesondere bindet sie nicht festgeschriebene Quelldateien ein.

> **Note**
>
> Unter Windows bewirkt das Hinzufügen von `.ebignore`, dass die EB CLI symbolischen Links folgt und die verknüpfte Datei bei der Erstellung eines Quell-Bundle einschließt. Das Problem ist bekannt und wird in einem der nächsten Updates behoben.

Verwenden von benannten Profilen

Wenn Sie Ihre Anmeldeinformationen in einer `credentials`- oder `config`-Datei als benanntes Profil speichern, können Sie die `--profile`-Option zur expliziten Angabe eines Profils verwenden. Mit dem folgenden Befehl wird beispielsweise eine neue Anwendung mit dem `user2`-Profil erstellt.

```
$ eb init --profile user2
```

Sie können auch das Standard-Profil ändern, indem Sie die `AWS_EB_PROFILE`-Umgebungsvariable festlegen. Wenn diese Variable festgelegt ist, liest die EB CLI Anmeldeinformationen aus dem angegebenen Profil anstatt `default` oder eb-cli.

Linux, macOS oder Unix

```
$ export AWS_EB_PROFILE=user2
```

Windows

```
> set AWS_EB_PROFILE=user2
```

Bereitstellen eines Artefakts anstelle des Projektordners

Sie können die EB CLI anweisen, eine ZIP- oder WAR-Datei bereitzustellen, die Sie als Teil eines separaten Erstellungsprozess generieren, indem Sie die folgenden Zeilen zu `.elasticbeanstalk/config.yml` in Ihrem Projektordner hinzufügen.

```
deploy:
  artifact: path/to/buildartifact.zip
```

Wenn Sie die EB CLI in Ihrem Git-Repository konfigurieren und das Artifact nicht in Ihrer Quelle bereitstellen, verwenden Sie die `--staged`-Option zur Bereitstellung des neuesten Builds.

```
~/eb$ eb deploy --staged
```

Konfigurationseinstellungen und Vorrang

Die EB CLI verwendet eine Anbieterkette, um an verschiedenen Stellen nach AWS Anmeldeinformationen zu suchen, einschließlich System- oder Benutzerumgebungsvariablen und lokalen AWS Konfigurationsdateien.

Die EB CLI sucht in der folgenden Reihenfolge nach Anmeldeinformationen und Konfigurationseinstellungen:

1. Befehlszeilenoptionen – Geben Sie ein benanntes Profil mit `--profile` an, um die Standardeinstellungen zu überschreiben.
2. Umgebungsvariablen – `AWS_ACCESS_KEY_ID` und `AWS_SECRET_ACCESS_KEY`.
3. Die AWS Anmeldeinformationsdatei — Befindet sich `~/.aws/credentials` auf Linux- und OS X-Systemen oder `C:\Users\USERNAME\.aws\credentials` unter Windows-Systemen. Diese Datei kann zusätzlich zu einem Standardprofil mehrere benannte Profile enthalten.
4. Die AWS CLI -Konfigurationsdatei – Gespeichert in `~/.aws/config` auf Linux- und OS X-Systemen oder in `C:\Users\USERNAME\.aws\config` auf Windows-Systemen.

   Diese Datei kann ein Standardprofil, benannte Profile und jeweils AWS CLI spezifische Konfigurationsparameter enthalten.
5. Bestehende EB CLI-Konfigurationsdatei – Gespeichert in `~/.elasticbeanstalk/config` auf Linux- und OS X-Systemen oder in `C:\Users\USERNAME\.elasticbeanstalk\config` auf Windows-Systemen.
6. Anmeldeinformationen für das Instance-Profil — Diese Anmeldeinformationen können auf EC2 Amazon-Instances mit einer zugewiesenen Instance-Rolle verwendet werden und werden über den EC2 Amazon-Metadatenservice bereitgestellt. Das Instance-Profil muss über die Berechtigung zur Nutzung von Elastic Beanstalk verfügen.

Wenn die Datei mit den Anmeldeinformationen ein benanntes Profil mit dem Namen "eb-cli" enthält, zieht die EB CLI dieses Profil dem Standard-Profil vor. Wenn keine Profile gefunden werden oder ein Profil gefunden wird, dieses jedoch nicht über die Berechtigung zur Nutzung von Elastic Beanstalk verfügt, fordert die EB CLI Sie auf, Schlüssel einzugeben.

Instance-Metadaten

Um die EB-CLI von einer EC2 Amazon-Instance aus zu verwenden, erstellen Sie eine Rolle, die Zugriff auf die benötigten Ressourcen hat, und weisen Sie diese Rolle der Instance zu, wenn sie gestartet wird. Starten Sie die Instance und installieren Sie die EB CLI mit `pip`.

```
~$ sudo pip install awsebcli
```

`pip` ist auf Amazon Linux vorinstalliert.

Die EB CLI liest die Anmeldeinformationen aus den Instance-Metadaten. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf AWS Ressourcen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden.

Verwenden der EB CLI mit Git

Die EB-CLI bietet eine Git-Integration, sodass Sie Code-Branches bestimmten Umgebungen in Elastic Beanstalk zuordnen können, um Ihre Deployments zu organisieren.

So installieren Sie Git und initialisieren Ihr Git-Repository

1. Laden Sie die neueste Version von Git unter http://git-scm.com herunter.
2. Initialisieren Sie Ihr Git-Repository, indem Sie den folgenden Befehl eingeben:

```
~/eb$ git init
```

EB CLI wird nun erkennen, dass Ihre Anwendung mit Git eingerichtet wurde.

3. Wenn Sie eb init noch nicht ausgeführt haben, tun Sie dies jetzt:

```
~/eb$ eb init
```

Zuordnen von Git-Verzweigungen zu Elastic Beanstalk-Umgebungen

Sie können jede Verzweigung Ihres Codes einer anderen Umgebung zuordnen. Wenn Sie eine Verzweigung abschließen, werden Änderungen für die zugeordnete Umgebung bereitgestellt. Sie können beispielsweise Folgendes eingeben, um Ihre Produktionsumgebung Ihrem Mainline-Zweig, und eine separate Entwicklungsumgebung Ihrer Entwicklungsverzweigung zuzuordnen:

```
~/eb$ git checkout mainline
~/eb$ eb use prod
~/eb$ git checkout develop
~/eb$ eb use dev
```

Bereitstellen von Änderungen

Standardmäßig stellt die EB CLI den neuesten Commit in der aktuellen Verzweigung bereit, wobei die Commit-ID und Nachricht als die Bezeichnung bzw. Beschreibung der Anwendungsversion verwendet wird. Wenn Sie ohne Commit in Ihrer Umgebung bereitstellen möchten, können Sie die `--staged`-Option verwenden, um Änderungen bereitzustellen, die zum Staging-Bereich hinzugefügt wurden.

So stellen Sie Änderungen ohne Commit bereit

1. Fügen Sie neue und geänderte Dateien zum Staging-Bereich hinzu:

```
~/eb$ git add .
```

2. Bereitstellen von zwischengespeicherten Änderungen mit eb deploy:

```
~/eb$ eb deploy --staged
```

Wenn Sie die EB CLI so konfiguriert haben, dass ein Artefakt bereitgestellt wird, und Sie das Artefakt nicht in Ihrem git-Repository bereitstellen, verwenden Sie die `--staged`-Option zur Bereitstellung des neuesten Builds.

Verwenden von Git-Submodulen

Einige Code-Projekte profitieren davon, dass sie Git-Submodul-Repositorys innerhalb des Top-Level-Repository haben. Wenn Sie Ihren Code mithilfe von eb create oder eb deploy bereitstellen, kann die EB CLI Submodule in die ZIP-Datei der Anwendungsversion einfügen und sie mit den Rest des Codes hochladen.

Sie können die Einbeziehung von Submodulen kontrollieren, indem Sie die Option `include_git_submodules` im Abschnitt `global` der EB CLI-Konfigurationsdatei `.elasticbeanstalk/config.yml` in Ihrem Projektordner verwenden.

Um die Submodule einzubeziehen, legen Sie diese Option auf `true` fest:

```
global:
  include_git_submodules: true
```

Wenn die Option `include_git_submodules` fehlt oder auf `false` festgelegt ist, bezieht EB CLI keine Submodule in die hochgeladene ZIP-Datei ein.

Weitere Informationen über Git-Submodule finden Sie unter Git Tools - Submodules.

> **Standardverhalten**
>
> Wenn Sie eb init zum Konfigurieren Ihres Projekts ausführen, fügt die EB CLI die Option `include_git_submodules` hinzu und legt sie auf `true` fest. Auf diese Weise wird sichergestellt, dass alle Submodule, die Sie in Ihrem Projekt haben, in Ihre Bereitstellungen einbezogen werden.
> Die EB CLI hat die Einbeziehung von Submodulen nicht immer unterstützt. Um eine versehentliche und unerwünschte Änderung an vorher bereits vorhandenen Projekten zu verhindern, haben wir Submodul-Support hinzugefügt. Die EB CLI bezieht keine Submodule ein, wenn die Option `include_git_submodules` fehlt. Wenn Sie eines dieser vorhandenen Projekte haben und Sie Submodule in Ihren Bereitstellungen einbeziehen möchten, fügen Sie die Option hinzu und legen Sie sie auf `true` fest, wie in diesem Abschnitt erklärt.

> **CodeCommit Verhalten**
>
> Die Integration von Elastic Beanstalk mit unterstützt derzeit CodeCommit keine Submodule. Wenn Sie für Ihre Umgebung die Integration mit aktiviert haben CodeCommit, sind Submodule nicht in Ihren Bereitstellungen enthalten.

Zuweisen von Git-Tags zu Ihrer Anwendungsversion

Sie können ein Git-Tag als Versionsbezeichnung verwenden, um zu identifizieren, welche Anwendungsversion in Ihrer Umgebung ausgeführt wird. Geben Sie zum Beispiel Folgendes ein:

```
~/eb$ git tag -a v1.0 -m "My version 1.0"
```

EB-CLI-Befehlsreferenz

Mit der Elastic-Beanstalk-Befehlszeilenschnittstelle (EB-CLI) können Sie zahlreiche Vorgänge für die Bereitstellung und Verwaltung Ihrer Elastic-Beanstalk-Anwendungen und -Umgebungen ausführen. Die EB-CLI kann in Git integriert werden, sodass Sie Anwendungsquellcode mit der Git-Quellüberwachung bereitstellen können. Weitere Informationen erhalten Sie unter Einrichtung der EB-Befehlszeilenschnittstelle (EB CLI) zur Verwaltung von Elastic Beanstalk und Verwenden der EB CLI mit Git.

Befehle

- Allgemeine Optionen
- eb abort
- eb appversion
- eb clone
- eb codesource
- eb config
- eb console
- eb create
- eb deploy
- eb events
- eb health
- eb init
- eb labs
- eb list
- eb local
- eb logs
- eb migrate
- eb open
- eb platform
- eb printenv

- eb restore
- eb scale
- eb setenv
- eb ssh
- eb status
- eb swap
- eb tags
- eb terminate
- eb upgrade
- eb use

Allgemeine Optionen

Sie können die folgenden Optionen mit allen EB-CLI-Befehlen verwenden.

Name	Beschreibung
`--debug`	Ausdrucken von Informationen für das Debugging.
`-h`, `--help`	Anzeigen der Hilfenachricht. Typ: Zeichenfolge Standard: keiner
`--no-verify-ssl`	Überspringen der SSL-Zertifikatprüfung. Verwenden Sie diese Option, wenn bei der Nutzung der CLI mit einem Proxy Probleme auftreten.
`--profile`	Verwenden Sie ein bestimmtes Profil aus Ihrer AWS Anmeldein formationsdatei.
`--quiet`	Unterdrücken der gesamten Ausgabe des Befehls.
`--region`	Verwenden der angegebenen Region.

Name	Beschreibung
`-v`, `--verbose`	Anzeigen von ausführlichen Informationen.

eb abort

Description

Bricht ein Upgrade ab, während die Umgebungskonfigurationsänderungen für die Instances noch ausgeführt werden.

> **Note**
>
> Falls für mehr als zwei Umgebungen eine Aktualisierung ausgeführt wird, werden Sie aufgefordert, den Namen der Umgebung anzugeben, für die Änderungen zurückgesetzt werden sollen.

Syntax

eb abort

eb abort ***environment-name***

Optionen

Name	Beschreibung
Allgemeine Optionen	

Ausgabe

Mit dem Befehl wird eine Liste der Umgebungen angezeigt, die gerade aktualisiert werden. Zudem werden Sie aufgefordert, die abzubrechende Aktualisierung auszuwählen. Sofern derzeit nur eine Umgebung aktualisiert wird, muss kein Umgebungsname angegeben werden. Bei einer erfolgreichen

Ausführung werden mit diesem Befehl die Umgebungskonfigurationsänderungen zurückgesetzt. Der Rollbackvorgang wird fortgesetzt, bis alle Instances der Umgebung wieder die vorherige Umgebungskonfiguration aufweisen oder bis im Rollback Fehler auftreten.

Beispiel

Mit dem folgenden Beispiel wird die Plattformaktualisierung abgebrochen.

```
$ eb abort
Aborting update to environment "tmp-dev".
<list of events>
```

eb appversion

Description

Der EB-CLI-Befehl `appversion` verwaltet Ihre Elastic-Beanstalk -Anwendungsversionen. Sie können eine neue Version der Anwendung erstellen, ohne sie bereitzustellen, eine Version der Anwendung löschen oder die Anwendungsversions-Lebenszyklus-Richtlinie erstellen. Wenn Sie den Befehl ohne Optionen aufrufen, wird dieser in den interaktiven Modus versetzt.

Verwenden Sie die `--create`-Option zum Erstellen einer neuen Version der Anwendung.

Verwenden Sie die `--delete`-Option zum Löschen einer Version der Anwendung.

Verwenden Sie die `lifecycle`-Option zum Anzeigen oder Erstellen der Anwendungsversions-Lebenszyklus-Richtlinie. Weitere Informationen finden Sie unter the section called “Lebenszyklus der Version”.

Syntax

eb appversion

eb appversion [-c | --create]

eb appversion [-d | --delete] ***version-label***

eb appversion lifecycle [-p | --print]

Optionen

Name	Beschreibung Typ: Zeichenfolge
-a *application-name* oder --Anwendungsname *application-name*	Der Name der Anwendung. Wenn eine Anwendung mit dem angegebenen Namen nicht gefunden wird, erstellt die EB-CLI eine Anwendungsversion für eine neue Anwendung. Gilt nur für die Option `--create`. Typ: Zeichenfolge
-c oder --create	Erstellen Sie eine neue Version der Anwendung.
-d *version-label* oder --löschen *version-label*	Löscht die Version der Anwendung, die beschriftet *version-label* ist.
`-l` *version_label* oder `--label` *version_label*	Geben Sie eine Bezeichnung für die Version an, die die EB-CLI erstellt. Wenn Sie diese Option nicht verwenden, generiert die EB-CLI eine neue eindeutige Bezeichnung. Wenn Sie eine Versionsbezeichnung angeben, sollten Sie sicherstellen, dass sie eindeutig ist. Gilt nur für die Option `--create`. Typ: Zeichenfolge
Lebenszyklus	Rufen Sie den Standard-Editor zum Erstellen einer neuen Anwendungsversions-Lebenszyklus-Richtlinie auf. Verwenden Sie diese Richtlinie, um zu vermeiden, dass die Grenze für das Anwendungsversionskontingent erreicht wird.

Name	Beschreibung Typ: Zeichenfolge
lifecycle -p oder lifecycle --print	Zeigen Sie die aktuelle Anwendungs-Lebenszyklus-Richtlinie an.

`-m "`*`version_description`*` "`
 	
oder
 	
`--message "`*`version_description`*`  "`
	Die Beschreibung für die Anwendungsversion. Sie ist in doppelten Anführungszeichen eingeschlossen.
	
	Gilt nur für die Option `--create`.
	
	Typ: Zeichenfolge

`-p`
 	
oder
 	
`--process`
	Vorverarbeiten und Validieren des Umgebungsmanifests und der Konfigurationsdateien im Quell-Bundle. Das Validieren von Konfigurationsdateien kann Probleme identifizieren. Wir empfehlen dies vor der Bereitstellung der Anwendungsversion in der Umgebung zu tun.
	
	Gilt nur für die Option `--create`.

`--source codecommit/` *`repository-name`*`/`*`branch-name`*	CodeCommit Repository und Zweig. Gilt nur für die Option `--create`.
`--staged`	Verwenden Sie anstelle des HEAD-Commit die zum Erstellen der Anwendungsversion im Git-Index bereitgestellten Dateien. Gilt nur für die Option `--create`.
`--timeout` *`minutes`*	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt. Gilt nur für die Option `--create`.

Name	Beschreibung Typ: Zeichenfolge
Allgemeine Optionen	

Interaktives Verwenden des Befehls

Wenn Sie den Befehl ohne Argumente verwenden, werden die Anwendungsversionen in der Ausgabe angezeigt. Sie sind in umgekehrter chronologischer Reihenfolge aufgeführt, wobei die letzte Version zuerst aufgeführt ist. Beispiele dazu, wie der Bildschirm aussieht, finden Sie im Abschnitt Examples (Beispiele). Beachten Sie, dass die Statuszeile unten angezeigt wird. In der Statuszeile werden kontextbezogene Informationen angezeigt.

Drücken Sie d zum Löschen einer Anwendungsversion, drücken Sie `l` zum Verwalten der Lebenszyklus-Richtlinie für Ihre Anwendung oder drücken Sie q, um den Vorgang zu beenden, ohne Änderungen vorzunehmen.

> **Note**
>
> Wenn die Version in einer beliebigen Umgebung bereitgestellt wird, können Sie diese Version nicht löschen.

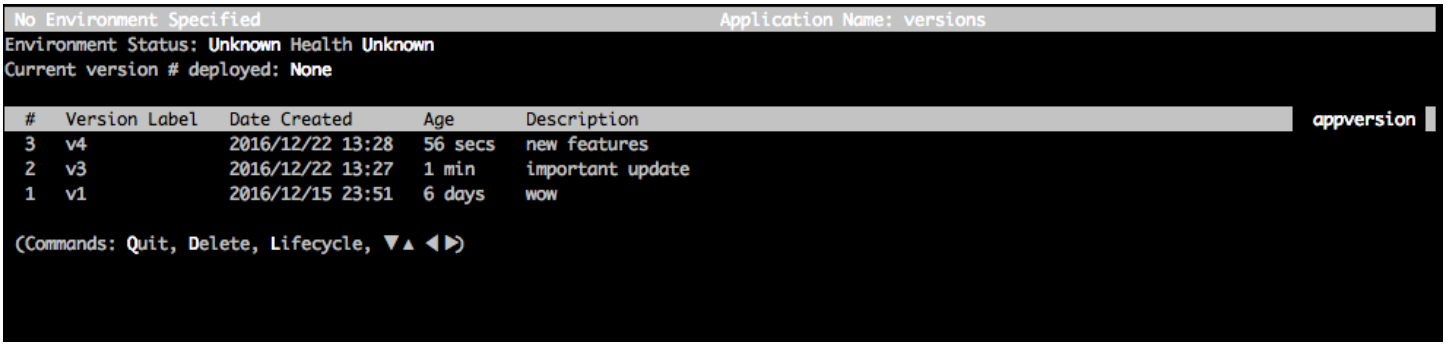
Ausgabe

Der Befehl mit der Option `--create` zeigt eine Meldung an, in der bestätigt wird, dass die Anwendungsversion erstellt wurde.

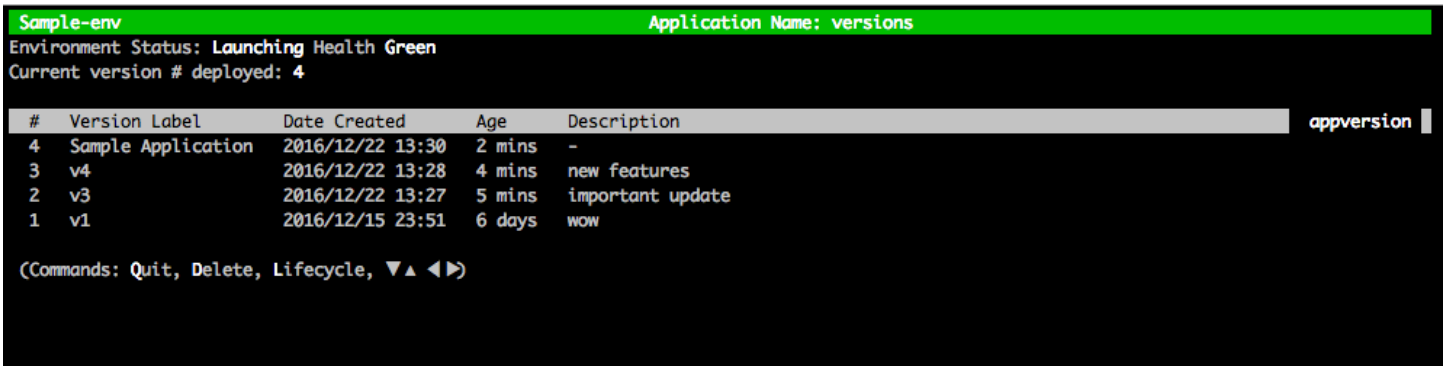
Der Befehl mit der `--delete` *`version-label`* Option zeigt eine Meldung an, die bestätigt, dass die Anwendungsversion gelöscht wurde.

Beispiele

Das folgende Beispiel zeigt das interaktive Fenster für eine Anwendung ohne Bereitstellungen.



Das folgende Beispiel zeigt das interaktive Fenster für eine Anwendung mit der vierten Version, wobei die Versionsbezeichnung Sample Application (Beispielanwendung) bereitgestellt ist.



Das folgende Beispiel zeigt die Ausgabe eines eb appversion lifecycle -p Befehls, wobei die Konto-ID des Benutzers angegeben *ACCOUNT-ID* ist:

```
Application details for: lifecycle
  Region: sa-east-1
  Description: Application created from the EB CLI using "eb init"
  Date Created: 2016/12/20 02:48 UTC
  Date Updated: 2016/12/20 02:48 UTC
  Application Versions: ['Sample Application']
  Resource Lifecycle Config(s):
    VersionLifecycleConfig:
      MaxCountRule:
        DeleteSourceFromS3: False
        Enabled: False
        MaxCount: 200
      MaxAgeRule:
        DeleteSourceFromS3: False
        Enabled: False
        MaxAgeInDays: 180
    ServiceRole: arn:aws:iam::ACCOUNT-ID:role/aws-elasticbeanstalk-service-role
```

eb clone

Description

Klont eine Umgebung in einer neuen Umgebung, damit beide identische Umgebungseinstellungen haben.

> **Note**
>
> Standardmäßig erstellt der eb clone-Befehl die Klonumgebung mit dem neuesten Lösungs-Stack, unabhängig von der Lösungs-Stack-Version der Umgebung, aus der Sie den Klon erstellen. Sie können dies unterdrücken, indem Sie die `--exact`-Option einschließen, wenn Sie den Befehl ausführen.

> **Important**
>
> Bei geklonten Elastic Beanstalk Beanstalk-Umgebungen werden die Sicherheitsgruppen für eingehenden Zugriff nicht übertragen, sodass die Umgebung für den gesamten Internetverkehr geöffnet bleibt. Sie müssen die Sicherheitsgruppen für eingehende Zugriffe für die geklonte Umgebung neu einrichten.
> Sie können sehen, welche Ressourcen möglicherweise nicht geklont wurden, indem Sie den Drift-Status Ihrer Umgebungskonfiguration überprüfen. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter Drift auf einem gesamten CloudFormation Stack erkennen.

Syntax

eb clone

eb clone ***environment-name***

Optionen

Name	Beschreibung
`-n` *`string`* oder `--clone_name` *`string`*	Gewünschter Name für die geklonte Umgebung.
`-c` *`string`* oder `--cname` *`string`*	Gewünschtes CNAME-Präfix für die geklonte Umgebung.
`--envvars`	Umgebungseigenschaften in einer durch Kommas getrennten Liste im Format *`name`* =. *`value`* Typ: Zeichenfolge Einschränkungen: • Schlüssel-Wert-Paare müssen durch Kommas getrennt werden. • Schlüssel und Werte können alphabetische Zeichen in jeder beliebigen Sprache, numerische Zeichen, Leerzeichen, unsichtbare Trennzeichen und die folgenden Symbole sein: _ . : / + \ - @ • Schlüssel können bis zu 128 Zeichen enthalten. Werte können bis zu 256 Zeichen enthalten. • Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt. • Werte können nicht mit dem Umgebungsnamen übereinst immen. • Werte können nicht `aws:` oder `elasticbeanstalk:` enthalten.

Name	Beschreibung
	• Die Größe aller Umgebungseigenschaften darf nicht mehr als 4096 Byte sein.
`--exact`	Verhindert, dass Elastic Beanstalk die Lösungs-Stack-Version für die neue Klonumgebung auf die neueste verfügbare Version aktualisiert (für die Plattform der ursprünglichen Umgebung).
`--scale` *`number`*	Die Anzahl der Instances, die in der Klonumgebung ausgeführt werden sollen, wenn diese gestartet wird.
`--tags` *`name`*`=`*`value`*	Tags für die Ressourcen in Ihrer Umgebung in einer durch Kommas getrennten Liste im Format =. *`name value`*
`--timeout`	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.
Allgemeine Optionen	

Ausgabe

Wenn der Befehl erfolgreich war, wird eine Umgebung erstellt, die dieselben Einstellungen wie die ursprüngliche Umgebung hat oder Änderungen aufweist, wie durch eb clone-Optionen angegeben.

Beispiel

Im folgenden Beispiel wird die angegebene Umgebung geklont.

```
$ eb clone
Enter name for Environment Clone
(default is tmp-dev-clone):
Enter DNS CNAME prefix
(default is tmp-dev-clone):
Environment details for: tmp-dev-clone
  Application name: tmp
  Region: us-west-2
  Deployed Version: app-141029_144740
  Environment ID: e-vjvrqnn5pv
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
```

```
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev-clone.elasticbeanstalk.com
  Updated: 2014-10-29 22:00:23.008000+00:00
Printing Status:
2018-07-11 21:04:20    INFO: createEnvironment is starting.
2018-07-11 21:04:21    INFO: Using elasticbeanstalk-us-west-2-888888888888 as Amazon S3
 storage bucket for environment data.
...
2018-07-11 21:07:10    INFO: Successfully launched environment: tmp-dev-clone
```

eb codesource

Description

Konfiguriert die EB-CLI für die Bereitstellung aus einem CodeCommit Repository oder deaktiviert die CodeCommit Integration und lädt das Quellpaket von Ihrem lokalen Computer hoch.

> **Note**
>
> In einigen AWS Regionen ist das nicht möglich. CodeCommit Die Integration zwischen Elastic Beanstalk und CodeCommit funktioniert in diesen Regionen nicht.
> Informationen zu den in den einzelnen Regionen angebotenen AWS Diensten finden Sie in der Regionstabelle.

Syntax

eb codesource

eb codesource codecommit

eb codesource local

Optionen

Name	Beschreibung
Allgemeine Optionen	

Ausgabe

eb codesourcefordert Sie auf, zwischen CodeCommit Integrations- und Standardbereitstellungen zu wählen.

Mit eb codesource codecommit wird die interaktive Repository-Konfiguration für die CodeCommit - Integration gestartet.

Mit eb codesource local wird die ursprüngliche Konfiguration angezeigt und die CodeCommit-Integration wird deaktiviert.

Beispiele

Wird verwendeteb codesource codecommit, um die CodeCommit Integration für den aktuellen Zweig zu konfigurieren.

```
~/my-app$ eb codesource codecommit
Select a repository
1) my-repo
2) my-app
3) [ Create new Repository ]
(default is 1): 1

Select a branch
1) mainline
2) test
3) [ Create new Branch with local HEAD ]
(default is 1): 1
```

Wird verwendeteb codesource local, um die CodeCommit Integration für den aktuellen Zweig zu deaktivieren.

```
~/my-app$ eb codesource local
Current CodeCommit setup:
  Repository: my-app
  Branch: mainline
Default set to use local sources
```

eb config

Description

Verwaltet die aktiven Konfigurationseinstellungen und gespeicherten Konfigurationen der Umgebung. Mit diesem Befehl können Sie die gespeicherten Konfigurationen der Umgebung hochladen, herunterladen oder auflisten. Sie können sie auch zum Herunterladen, Anzeigen oder Aktualisieren der aktiven Konfigurationseinstellungen verwenden.

Wenn das Stammverzeichnis eine `platform.yaml`-Datei enthält, die eine benutzerdefinierte Plattform angibt, ändert dieser Befehl auch die Builder-Konfigurationseinstellungen. Dies geschieht basierend auf den in festgelegten Werte `platform.yaml`.

> **Note**
>
> eb config zeigt keine Umgebungseigenschaften. Zum Festlegen von Umgebungseigenschaften, die Sie in Ihrer Anwendung lesen können, verwenden Sie stattdessen eb setenv.

Syntax

Im Folgenden sind Teile der Syntax aufgeführt, die für den Befehl eb config verwendet wird, um mit den aktiven Konfigurationseinstellungen der Umgebung zu arbeiten. Konkrete Beispiele finden Sie im Abschnitt Beispiele weiter unten in diesem Thema.

- eb config – Zeigt die aktiven Konfigurationseinstellungen der Umgebung in einem Texteditor an, den Sie als EDITOR-Umgebungsvariable konfiguriert haben. Wenn Sie Änderungen in der Datei speichern und den Editor schließen, wird die Umgebung mit den in der Datei gespeicherten Optionseinstellungen aktualisiert.

  > **Note**
  >
  > Wenn Sie keine EDITOR-Umgebungsvariable konfiguriert haben, zeigt die EB-CLI Ihre Optionseinstellungen im Standard-Editor für YAML-Dateien an.

- eb config ***environment-name***— Zeigt die Konfiguration für die benannte Umgebung an und aktualisiert diese. Die Konfiguration wird entweder in einem von Ihnen konfigurierten Texteditor oder im Standard-Editor für YAML-Dateien angezeigt.
- eb config save – Speichert die aktiven Konfigurationseinstellungen für die aktuelle Umgebung in `.elasticbeanstalk/saved_configs/` mit dem Dateinamen `[configuration-name].cfg.yml`. Standardmäßig speichert die EB-CLI die Konfigurationseinstellungen mit einem auf der Umgebung *`configuration-name`* basierenden Namen. Sie können einen anderen Konfigurationsnamen angeben, indem Sie die `--cfg`-Option in den gewünschten Konfigurationsnamen einschließen, wenn Sie den Befehl ausführen.

  Sie können Ihre gespeicherte Konfiguration unter Verwendung der `--tags`-Option markieren.
- eb config **`--display`**– Schreibt die aktiven Konfigurationseinstellungen einer Umgebung in stdout statt in eine Datei. Standardmäßig werden dadurch die Konfigurationseinstellungen für das Terminal angezeigt.
- eb config **`--update`** ***`configuration_string`*** **`|`** ***`file_path`***— Aktualisiert die aktiven Konfigurationseinstellungen für die aktuelle Umgebung mit den Informationen, die in *`configuration_string`* oder in der durch identifizierten Datei angegeben sind*`file_path`*.

> **Note**
>
> Die Optionen `--display` und `--update` bieten Flexibilität beim programmgesteuerten Lesen und Überarbeiten der Konfigurationseinstellungen einer Umgebung.

In der folgenden Tabelle wird die Syntax für den eb config-Befehl für die Arbeit mit gespeicherten Konfigurationen beschrieben. Beispiele finden Sie im Abschnitt Beispiele weiter unten in diesem Thema.

- eb config get ***`config-name`*** – Lädt die benannte gespeicherte Konfiguration von Amazon-S3 herunter.

- eb config delete ***`config-name`*** – Löscht die benannte gespeicherte Konfiguration von Amazon-S3. Löscht sie auch lokal, wenn Sie sie bereits heruntergeladen haben.
- eb config list – Listet die Konfigurationen auf, die Sie in Amazon-S3 gespeichert haben.
- eb config put ***`filename`*** – Lädt die benannte gespeicherte Konfiguration in einen Amazon-S3-Bucket hoch. Sie *`filename`* müssen die Dateierweiterung haben`.cfg.yml`. Um den Dateinamen

ohne Pfad anzugeben, können Sie die Datei im Ordner `.elasticbeanstalk` oder im Ordner `.elasticbeanstalk/saved_configs/` speichern, bevor Sie den Befehl ausführen. Alternativ können Sie das angeben, *`filename`* indem Sie den vollständigen Pfad angeben.

Optionen

Name	Beschreibung
`--cfg` *`config-name`*	Der Name, der für eine gespeicherte Konfiguration verwendet werden soll. Diese Option funktioniert nur mit dem Befehl eb config save.

`-d`
 	
oder
 	
`--display`
	Zeigt die aktiven Konfigurationseinstellungen für die aktuelle Umgebung an (schreibt in stdout).
	
	Verwenden Sie sie mit der Option `--format`, um anzugeben, dass die Ausgabe in JSON oder YAML erfolgen soll. Wenn Sie keine Angaben machen, erfolgt die Ausgabe im YAML-Format.
	
	Diese Option kann nur ausgeführt werden, wenn Sie den Befehl eb config ohne einen der anderen Unterbefehle verwenden.

`-f` *`format_type`* oder `--format` *`format_type`*	Gibt das Anzeigeformat an. Gültige Werte sind JSON und YAML. Die Standardeinstellung ist YAML. Diese Option funktioniert nur mit der Option `--display` .
`--tags` *`key1`*`=`*`value1`*`[,`*`ke`	Tags zum Hinzufügen zu Ihrer gespeicherten Konfiguration. Wenn Sie in der Liste Tags angeben, dann geben Sie diese als Schlüssel-Wert-Paare an und trennen Sie jedes mit einem Komma. Weitere Informationen finden Sie unter Markieren von gespeicherten Konfigurationen. Diese Option funktioniert nur mit dem Befehl eb config save.

Name	Beschreibung
`--timeout` *timeout*	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.

Name	Beschreibung

`-u configuration_string | file_path`
 	
oder
 	
`--update configuration_string | file_path`
	Aktualisiert die aktiven Konfigurationseinstellungen für die aktuelle Umgebung.
	
	Diese Option kann nur ausgeführt werden, wenn Sie den Befehl eb config ohne einen der anderen Unterbefehle verwenden.
	
	Der Parameter *configuration_string | file_path* hat den Typ Zeichenfolge. Die Zeichenfolge enthält die Liste der Namespaces und die entsprechenden Optionen zum Hinzufügen, Aktualisieren oder Entfernen aus den Konfigurationseinstellungen für die Umgebung. Alternativ kann die Eingabezeichenfolge eine Datei darstellen, die dieselben Informationen enthält.
	
	Um einen Dateinamen anzugeben, muss die Eingabezeichenfolge dem Format entsprechen "file://<*path*><*filename*>". Um den Dateinamen ohne *path* anzugeben, speichern Sie die Datei in dem Ordner, in dem Sie den Befehl ausführen. Alternativ legen Sie den Dateinamen fest, indem Sie den vollständigen Pfad angeben.
	
	Die Konfigurationsinformationen müssen die folgenden Bedingungen erfüllen. Mindestens einer der Abschnitte, OptionSettings oder OptionsToRemove, ist erforderlich. Verwenden Sie OptionSettings, um Optionen hinzuzufügen oder zu ändern. Verwenden Sie OptionsToRemove, um Optionen aus einem Namespace zu entfernen. Konkrete Beispiele finden Sie im Abschnitt Beispiele weiter unten in diesem Thema.
	
	Example
	
	YAML-Format
	
	`OptionSettings:`
	`  namespace1:`
	`    option-name-1: option-value-1`
	`    option-name-2: option-value-2`

Name	Beschreibung

	...
	OptionsToRemove:
	  namespace1:
	  option-name-1
	  option-name-2
	  ...
	
	Example
	
	JSON-Format
	
	{
	  "OptionSettings": {
	  "namespace1": {
	  "option-name-1": " *option-value-1* ",
	  "option-name-2": " *option-value-2* ",
	  ...
	  }
	  },
	  "OptionsToRemove": {
	  "namespace1": {
	  "option-name-1",
	  "option-name-2",
	  ...
	  }
	  }
	}

Allgemeine Optionen	

Ausgabe

Wenn der Befehl eb config oder eb config ***environment-name*** erfolgreich ohne hinzugefügte Unterbefehle oder Optionen ausgeführt wird, zeigt der Befehl Ihre aktuellen Optionseinstellungen im Text-Editor, die Sie als EDITOR-Umgebungsvariable konfiguriert haben. Wenn Sie keine EDITOR-Umgebungsvariable konfiguriert haben, zeigt die EB-CLI Ihre Optionseinstellungen im Standard-Editor für YAML-Dateien an.

Wenn Sie Änderungen in der Datei speichern und den Editor schließen, wird die Umgebung mit den in der Datei gespeicherten Optionseinstellungen aktualisiert. Die folgende Ausgabe wird angezeigt, um das Konfigurationsupdate zu bestätigen.

```
$ eb config myApp-dev
    Printing Status:
    2021-05-19 18:09:45    INFO    Environment update is starting.
    2021-05-19 18:09:55    INFO    Updating environment myApp-dev's configuration
 settings.
    2021-05-19 18:11:20    INFO    Successfully deployed new configuration to
 environment.
```

Wenn der Befehl erfolgreich mit der Option `--display` ausgeführt wird, werden die Konfigurationseinstellungen für die aktuelle Umgebung angezeigt (schreibt instdout).

Wenn der Befehl erfolgreich mit dem `get`-Parameter ausgeführt wird, zeigt der Befehl den Speicherort der lokalen Kopie, die Sie heruntergeladen haben.

Wenn der Befehl erfolgreich mit dem `save`-Parameter ausgeführt wird, zeigt der Befehl den Speicherort der gespeicherten Datei.

Beispiele

In diesem Abschnitt wird beschrieben, wie Sie den Text-Editor ändern, den Sie zum Anzeigen und Bearbeiten Ihrer Optionseinstellungen verwenden.

Für Linux und UNIX wird im folgenden Beispiel der Editor auf vim geändert:

```
$ export EDITOR=vim
```

Für Linux und UNIX wird im folgenden Beispiel der Editor auf das geändert, was unter installiert is `/usr/bin/kate`.

```
$ export EDITOR=/usr/bin/kate
```

Für Windows wird im folgenden Beispiel der Editor auf Notepad++ geändert:

```
> set EDITOR="C:\Program Files\Notepad++\Notepad++.exe
```

Dieser Abschnitt zeigt Beispiele für den eb config-Befehl, wenn er mit Unterbefehlen ausgeführt wird.

Im folgenden Beispiel wird die gespeicherte Konfiguration mit dem Namen gelösch `app-tmp`.

```
$ eb config delete app-tmp
```

Im folgenden Beispiel wird die gespeicherte Konfiguration mit dem Namen "app-tmp" aus Ihrem Amazon-S3-Bucket heruntergeladen.

```
$ eb config get app-tmp
```

Im folgenden Beispiel werden die Namen der gespeicherten Konfigurationen aufgelistet, die in Ihrem Amazon-S3-Bucket gespeichert sind.

```
$ eb config list
```

Im folgenden Beispiel wird die lokale Kopie der gespeicherten Konfiguration mit dem Namen "app-tmp" in Ihren Amazon-S3-Bucket hochgeladen.

```
$ eb config put app-tmp
```

Im folgenden Beispiel werden Konfigurationseinstellungen aus der aktuell laufenden Umgebung gespeichert. Wenn Sie keinen Namen für die gespeicherte Konfiguration angeben, benennt Elastic Beanstalk die Konfigurationsdatei entsprechend des Umgebungsnamens. Beispiel: Eine Umgebung mit dem Namen tmp-dev würde als `tmp-dev.cfg.yml` bezeichnet werden. Elastic Beanstalk speichert die Datei im Ordner `/.elasticbeanstalk/saved_configs/`.

```
$ eb config save
```

Im folgenden Beispiel wird die `--cfg`-Option zum Speichern der Konfigurationseinstellungen aus der Umgebung "tmp-dev" in eine Datei mit dem Namen `v1-app-tmp.cfg.yml` gespeichert wird. Elastic Beanstalk speichert die Datei im Ordner `/.elasticbeanstalk/saved_configs/`. Wenn Sie keinen Umgebungsnamen angeben, speichert Elastic-Beanstalk-Konfigurationseinstellungen aus der aktuell laufenden Umgebung.

```
$ eb config save tmp-dev --cfg v1-app-tmp
```

Dieser Abschnitt zeigt Beispiele für den eb config-Befehl, wenn er ohne Unterbefehle ausgeführt wird.

Der folgende Befehl zeigt die Optionseinstellungen Ihrer aktuellen Umgebung in einem Texteditor an.

```
$ eb config
```

Der folgende Befehl zeigt die Optionseinstellungen für die my-env-Umgebung in einem Texteditor an.

```
$ eb config my-env
```

Im folgenden Beispiel werden die Optionseinstellungen für die aktuelle Umgebung angezeigt. Die Ausgabe erfolgt im YAML-Format, da mit der Option `--format` kein bestimmtes Format angegeben wurde.

```
$ eb config --display
```

Im folgenden Beispiel werden die Optionseinstellungen für Ihre aktuelle Umgebung mit den Angaben in der Datei namens aktualisier `example.txt`. Die Datei ist entweder im YAML- oder JSON-Format. Die EB-CLI erkennt das Dateiformat automatisch.

- Die Option MinSize ist für den Namespace auf 1 festgeleg `aws:autoscaling:asg`.
- Die Batch-Größe für den Namespace `aws:elasticbeanstalk:command` ist auf 30 % festgelegt.
- Es entfernt die Optionseinstellung IdleTimeout: Keine aus dem NamespaceAWSEBV2LoadBalancer.`aws:elbv2:loadbalancer`.

```
$ eb config --update "file://example.txt"
```

Example- Dateiname: example.txt – YAML-Format

```
OptionSettings:
  'aws:elasticbeanstalk:command':
    BatchSize: '30'
    BatchSizeType: Percentage
  'aws:autoscaling:asg':
    MinSize: '1'
OptionsToRemove:
  'AWSEBV2LoadBalancer.aws:elbv2:loadbalancer':
    IdleTimeout
```

Example- Dateiname: example.txt – JSON-Format

```
{
```

```
    "OptionSettings": {
        "aws:elasticbeanstalk:command": {
            "BatchSize": "30",
            "BatchSizeType": "Percentage"
        },
        "aws:autoscaling:asg": {
            "MinSize": "1"
        }
    },
    "OptionsToRemove": {
        "AWSEBV2LoadBalancer.aws:elbv2:loadbalancer": {
            "IdleTimeout"
        }
    }
}
```

Im folgenden Beispiel werden die Optionseinstellungen für die aktuelle Umgebung aktualisiert. Der Befehl legt die Minsize-Option für den Namespace `aws:autoscaling:asg` auf 1 fest.

> **Note**
>
> Diese Beispiele sind spezifisch für Windows PowerShell. Sie umgehen literale Vorkommen des Anführungszeichens ("), indem sie ihm einen Schrägstrich (\) voranstellen. Verschiedene Betriebssysteme und Befehlszeilenumgebungen haben möglicherweise unterschiedliche Escape-Sequenzen. Aus diesem Grund empfehlen wir, die in den vorherigen Beispielen gezeigte Dateioption zu verwenden. Die Angabe der Konfigurationsoptionen in einer Datei erfordert keine Escape-Zeichen und ist für verschiedene Betriebssysteme konsistent.

Das folgende Beispiel verwendet das JSON-Format. Die EB-CLI erkennt, ob ein JSON- oder YAML-Format vorliegt.

```
PS C:\Users\myUser\EB_apps\myApp-env>eb config --update '{\"OptionSettings\":
{\"aws:autoscaling:asg\":{\"MaxSize\":\"1\"}}}'
```

Das folgende Beispiel verwendet das YAML-Format. Um die YAML-Zeichenfolge im richtigen Format einzugeben, beinhaltet der Befehl Leerzeichen und end-of-line Zeilenumbrüche, die in einer YAML-Datei erforderlich sind.

- Schließen Sie jede Zeile mit der „Enter“- oder „Return“-Taste ab.
- Beginnen Sie die zweite Zeile mit zwei Leerzeichen und die dritte Zeile mit vier Leerzeichen.

```
PS C:\Users\myUser\EB_apps\myApp-env>eb config --update 'OptionSettings:
>>  aws:autoscaling:asg:
>>    MinSize: \"1\"'
```

eb console

Description

Öffnet einen Browser zum Anzeigen des Umgebungskonfigurations-Dashboards in der Elastic-Beanstalk-Managementkonsole.

Wenn das Stammverzeichnis eine `platform.yaml`-Datei enthält, die eine benutzerdefinierte Plattform angibt, zeigt dieser Befehl auch die Architekten-Umgebungskonfiguration, wie in `platform.yaml` angegeben, in der Elastic-Beanstalk-Managementkonsole an.

Syntax

eb console

eb console ***environment-name***

Optionen

Name	Beschreibung
Allgemeine Optionen	

eb create

Description

Erstellt eine neue Umgebung und stellt eine Anwendungsversion darin bereit.

> **Note**
>
> - Um den Befehl eb create für eine .NET-Anwendung zu verwenden, müssen Sie ein Bereitstellungspaket wie in Erstellen eines Quell-Bundles für eine .NET-Anwendung beschrieben erstellen und die CLI-Konfiguration zum Bereitstellen des Pakets als Artefakt gemäß der Beschreibung in Bereitstellen eines Artefakts anstelle des Projektordners einrichten.
> - Zum Erstellen von Umgebungen mit der EB-CLI ist eine Service-Rolle erforderlich. Eine Servicerolle generieren Sie, indem Sie eine Umgebung in der Elastic-Beanstalk-Konsole erstellen. Falls keine Servicerolle vorhanden ist, versucht die EB-CLI eine Rolle zu erstellen, wenn Sie ausführen `eb create`.

Sie können die Anwendungsversion aus unterschiedlichen Quellen bereitstellen:

- Standardmäßig: die Anwendung wird aus dem Quellcode im lokalen Projektverzeichnis bereitgestellt.
- Mit der Option `--version` kann die Anwendung aus einer Anwendungsversion bereitgestellt werden, die bereits in Ihrer Anwendung vorhanden ist.
- Wenn Ihr Projektverzeichnis keinen Anwendungscode enthält oder wenn Sie die Option `--sample` verwenden: Bereitgestellt aus einer Beispielanwendung, die für die Plattform Ihrer Umgebung spezifisch ist.

Syntax

eb create

eb create ***environment-name***

Ein Umgebungsname muss zwischen 4 und 40 Zeichen lang sein. Der Name darf nur Buchstaben, Zahlen und Bindestriche (-) enthalten. Ein Umgebungsname darf nicht mit einem Bindestrich beginnen oder enden.

Wenn Sie einen Umgebungsnamen in den Befehl aufnehmen, fordert die EB-CLI Sie nicht auf, eine Auswahl zu treffen oder eine Servicerolle zu erstellen.

Wenn Sie den Befehl ohne Umgebungsnamensargument ausführen, wird er in einem interaktiven Ablauf ausgeführt und Sie werden aufgefordert, Werte für einige Einstellungen auszuwählen. Falls

Sie eine Beispielanwendung bereitstellen, werden Sie in diesem interaktiven Ablauf von der EB-CLI auch gefragt, ob Sie diese Beispielanwendung in Ihr lokales Projektverzeichnis herunterladen möchten. Durch Herunterladen können Sie die EB-CLI später mit der neuen Umgebung verwenden, um Vorgänge auszuführen, die den Code der Anwendung erfordern, z. B. eb deploy.

Einige interaktive Flow-Aufforderungen werden nur unter bestimmten Bedingungen angezeigt. Wenn Sie beispielsweise einen Application Load Balancer verwenden möchten und Ihr Konto mindestens einen freigebbaren Application Load Balancer enthält, zeigt Elastic Beanstalk eine Aufforderung an, in der Sie gefragt werden, ob Sie einen freigegebenen Load Balancer verwenden möchten. Wenn in Ihrem Konto kein gemeinsam nutzbarer Application Load Balancer vorhanden ist, wird diese Eingabeaufforderung nicht angezeigt.

Optionen

Keine dieser Optionen ist erforderlich. Wenn Sie eb create ohne Optionen ausführen, werden Sie von der EB-CLI aufgefordert, einen Wert für jede Einstellung einzugeben oder auszuwählen.

Name	Beschreibung

`-d`
 	
oder
 	
`--branch_default`
	Legen Sie die Umgebung als Standard-Umgebung für das aktuelle Repository fest.

`--cfg` *`config-name`*	Verwenden Sie Plattformeinstellungen aus einer gespeicherten Konfiguration in `.elasticbeanstalk/saved_configs/` oder Ihrem Amazon-S3-Bucket. Geben Sie den Namen der Datei ohne die `.cfg.yml`-Erweiterung an.
`-c` *`subdomain-name`* oder `--cname` *`subdomain-name`*	Der Name der Subdomäne für ein Präfix des CNAME DNS-Eintrags, der auf Ihre Website weiterleitet. Typ: Zeichenfolge Standard: Der Umgebungsname
`-db`	Fügt eine Datenbank an die Umgebung an. Wenn Sie eb create mit der Option `--database` , jedoch ohne die

Name	Beschreibung
oder `--database`	Optionen `--database.username` und `--database.password` ausführen, werden Sie von EB-CLI aufgefordert, den Master-Benutzernamen und das Master-Passwort der Datenbank einzugeben.
`-db.engine` *`engine`* oder `--database.engine` *`engine`*	Der Typ der Datenbank-Engine. Wenn Sie eb create mit dieser Option ausführen, startet EB-CLI die Umgebung mit einer angefügten Datenbank. Dies ist auch dann der Fall, wenn Sie den Befehl nicht mit der Option `--database` ausgeführt haben. Typ: Zeichenfolge Zulässige Werte: `mysql`, `oracle-se1` , `postgres`, `sqlserver-ex` , `sqlserver-web` , `sqlserver-se`
`-db.i` *`instance_type`* oder `--database.instance` *`instance_type`*	Der Typ der EC2 Amazon-Instance, die für die Datenbank verwendet werden soll. Wenn Sie eb create mit dieser Option ausführen, startet EB-CLI die Umgebung mit einer angefügten Datenbank. Dies ist auch dann der Fall, wenn Sie den Befehl nicht mit der Option `--database` ausgeführt haben. Typ: Zeichenfolge Zulässige Werte: Amazon RDS unterstützt einen Standardsatz DB-Instances. Beachten Sie bei der Auswahl der geeigneten DB-Instance diese spezifischen Aspekte. Weitere Informationen finden Sie unter Erstellen einer DB-Instance im Amazon RDS-Benutzerhandbuch.

Name	Beschreibung
`-db.pass` *password* oder `--database.password` *password*	Das Passwort für die Datenbank. Wenn Sie eb create mit dieser Option ausführen, startet EB-CLI die Umgebung mit einer angefügten Datenbank. Dies ist auch dann der Fall, wenn Sie den Befehl nicht mit der Option `--database` ausgeführt haben.
`-db.size` *number_of_gigabytes* oder `--database.size` *number_of_gigabytes*	Die Anzahl der Gigabyte (GB) für die Datenbank speicherung. Wenn Sie eb create mit dieser Option ausführen, startet EB-CLI die Umgebung mit einer angefügten Datenbank. Dies ist auch dann der Fall, wenn Sie den Befehl nicht mit der Option `--database` ausgeführt haben. Typ: Zahl Zulässige Werte: • MySQL – `5` bis `1024`. Der Standardwert ist `5`. • Postgres – `5` bis `1024`. Der Standardwert ist `5`. • Oracle – `10` bis `1024`. Der Standardwert ist `10`. • Microsoft SQL Server Express Edition – `30`. • Microsoft SQL Server Web Edition – `30`. • Microsoft SQL Server Standard Edition – `200`.
`-db.user` *username* oder `--database.username` *username*	Der Benutzername für die Datenbank. Wenn Sie eb create mit dieser Option ausführen, startet EB-CLI die Umgebung mit einer angefügten Datenbank, auch wenn Sie den Befehl nicht mit der `--database` -Option ausgeführt haben. Wenn Sie eb create mit der `--database` -Option, jedoch ohne die `--database.username` - und `--database.password` -Optionen ausführen, werden Sie von EB-CLI aufgefordert, den Benutzernamen und das Passwort der Master-Datenbank einzugeben.

Name	Beschreibung
`-db.version` *`version`* oder `--database.version` *`version`*	Wird verwendet, um die Version des Datenbankmoduls anzugeben. Wenn diese Markierung vorhanden ist, startet die Umgebung mit einer Datenbank mit der angegeben en Versionsnummer, auch wenn die `--database` Markierung nicht vorhanden ist.
`--elb-type` *`type`*	Der Load Balancer-Typ. Typ: Zeichenfolge Zulässige Werte: `classic`, `application` , `network` Standard: `application`

`-es`
 	
oder
 	
`--enable-spot`
	Aktivieren Sie Spot-Instance-Anforderungen für Ihre Umgebung. Weitere Informationen finden Sie unter Auto Scaling-Gruppe.
	
	Verwandte Optionen:
	
	• `--instance-types`
	• `--on-demand-base-capacity`
	• `--on-demand-above-base-capacity`
	• `--spot-max-price`

`--env-group-suffix` *`groupname`*	Der Gruppenname, der an den Umgebungsnamen angehängt werden soll. Nur zur Verwendung mit Compose-Umgebungen.
`--envvars`	Umgebungseigenschaften in einer durch Kommas getrennten Liste im Format *`name`* =. *`value`* Informationen zu den Beschränkungen finden Sie unter Konfiguration von Umgebungseigenschaften (Umgebungsvariablen).

Name	Beschreibung
`-ip` *`profile_name`* oder `--instance_profile` *`profile_name`*	Das Instanzprofil mit der IAM-Rolle mit den temporären Sicherheitsanmeldedaten, die Ihre Anwendung für den Zugriff auf Ressourcen benötigt. AWS

`-it`
 	
oder
 	
`--instance-types` *`type1`*`[,`*`type2`* `...]`
	Eine durch Kommas getrennte Liste von EC2 Amazon-Instance-Typen, die Ihre Umgebung verwenden soll. Wenn Sie diese Option nicht angeben, stellt Elastic Beanstalk Standard-Instance-Typen bereit.
	
	Weitere Informationen erhalten Sie unter EC2 Amazon-Instanzen und Auto Scaling-Gruppe.
	
	Important
	Die EB-CLI wendet diese Option nur auf Spot-Instances an. Es sei denn, diese Option wird mit dem `--enable-spot` verwenden, ignoriert die EB-CLI diese Option. Um einen Instanztyp für eine On-Demand-Instance anzugeben, verwenden Sie dazu die`--intance-type` (keine „s“) Option.

Name	Beschreibung

`-i`
 	
oder
 	
`--instance_type`
	Der EC2 Amazon-Instance-Typ, den Ihre Umgebung verwenden soll. Wenn Sie diese Option nicht angeben, stellt Elastic Beanstalk einen Standard-Instance-Typ bereit.
	
	Weitere Informationen finden Sie unter EC2 Amazon-Instanzen.
	
	Important
	Die EB-CLI wendet diese Option nur auf On-Demand-Instances an. Verwenden Sie diese Option nicht mit dem `--enable-spot` , da die EB-CLI sie ignoriert, wenn Sie dies tun. Um Exemplartypen für eine Spot-Instance anzugeben , verwenden Sie die Option`--intance-types` (mit der Option „s“).

`-k` *`key_name`* oder `--keyname` *`key_name`*	Der Name des EC2 Amazon-Schlüsselpaars, das mit dem Secure Shell (SSH) -Client verwendet werden soll, um sich sicher bei den EC2 Amazon-Instances anzumelden, auf denen Ihre Elastic Beanstalk Beanstalk -Anwendung ausgeführt wird. Wenn Sie diese Option mit dem eb create Befehl bereitstellen, überschreibt der Wert alle Schlüsselnamen, die Sie mit eb initangegeben haben. Gültige Werte: Ein vorhandener Schlüsselname, der bei Amazon registriert ist EC2

Name	Beschreibung
`-im` *number-of-instances* oder `--min-instances` *number-of-instances*	Die Mindestanzahl von EC2 Amazon-Instances, über die Ihre Umgebung verfügen muss. Typ: Zahl (Ganzzahl) Standard: `1` Gültige Werte: `1` bis `10000`
`-ix` *number-of-instances* oder `--max-instances` *number-of-instances*	Die maximale Anzahl von EC2 Amazon-Instances, die Sie Ihrer Umgebung zur Verfügung stellen. Typ: Zahl (Ganzzahl) Standard: `4` Gültige Werte: `1` bis `10000`
`--modules` *component-a component-b*	Liste der zu erstellenden Komponentenumgebungen. Nur zur Verwendung mit Compose-Umgebungen.

`-sb`
 	
oder
 	
`--on-demand-base-capacity`
	Die Mindestanzahl von On-Demand-Instances, die von der Auto Scaling-Gruppe bereitgestellt werden, bevor Spot-Instances bei der Skalierung Ihrer Umgebung in Betracht gezogen werden.
	
	Diese Option kann nur mit der Option `--enable-spot` angegeben werden. Weitere Informationen finden Sie unter Auto Scaling-Gruppe.
	
	Typ: Zahl (Ganzzahl)
	
	Standard: `0`
	
	Gültige Werte: `0` bis `--max-instances` (wenn nicht vorhanden: `MaxSize`-Option im `aws:autoscaling:asg` Namespace)

Name	Beschreibung

`-sp`
 	
oder
 	
`--on-demand-above-base-capacity`
	Der Anteil der On-Demand-Instances als Teil der zusätzlichen Kapazität, den Ihre Auto-Scaling-Gruppe über die durch die `--on-demand-base-capacity` -Option angegebene Anzahl von Instances hinaus bereitstellt.
	
	Diese Option kann nur mit der Option `--enable-spot` angegeben werden. Weitere Details finden Sie unter Auto Scaling-Gruppe.
	
	Typ: Zahl (Ganzzahl)
	
	Standard: `0` für eine Einzel-Instance-Umgebung; `70` für eine lastverteilte Umgebung
	
	Gültige Werte: `0` bis `100`

Name	Beschreibung
`-p` *`platform-version`* oder `--platform` *`platform-version`*	Die zu verwendende Plattformversion . Sie können eine Plattform, Name und Version einer Plattform, einen Plattformzweig, einen Lösungs-Stack-Namen oder einen Lösungs-Stack-ARN angeben. Zum Beispiel: • `php`, `PHP`, `node.js` – Die aktuelle Plattformversion für die angegebene Plattform • `php-7.2`, `"PHP 7.2"` – Die empfohlene (typische rweise neueste) PHP 7.2-Plattformversion • `"PHP 7.2 running on 64bit Amazon Linux"` – Die empfohlene (typischerweise neueste) PHP-Platt formversion in diesem Plattformzweig • `"64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1"` – Die PHP-Plattformversion, die von diesem Lösungs-Stack-Namen angegeben wird • `"arn:aws:elasticbeanstalk:us-east-2::platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3"` – Die PHP-Plattformversi on, die durch diesen Lösungs-Stack-ARN angegeben wird Verwenden Sie `eb platform list`, um eine Liste der verfügbaren Konfigurationen anzuzeigen. Wenn Sie die `--platform` -Option angeben, wird der durch `eb init` bereitgestellte Wert überschrieben.

`-pr`
 	
oder
 	
`--process`
	Vorverarbeiten und Validieren des Umgebungsmanifests und der Konfigurationsdateien im Quell-Bundle. Durch die Validierung der Konfigurationsdateien können Probleme vor der Bereitstellung der Anwendungsversion für eine Umgebung identifiziert werden.

Name	Beschreibung
`-r` *region* oder `--region` *region*	Die AWS Region, in der Sie die Anwendung bereitstellen möchten. Eine Liste der Werte, die Sie für diese Option angeben können, finden Sie unter AWS Elastic Beanstalk - Endpunkte und -Kontingente im Allgemeine AWS-Referenz.
`--sample`	Stellen Sie die Beispielanwendung für die neue Umgebung anstelle von Code in Ihrem Repository bereit.
`--scale` *number-of-instances*	Starten mit der angegebenen Anzahl von Instances
`--service-role` *servicerole*	Weisen Sie der Umgebung eine nicht standardmäßige Servicerolle zu. **Note** Geben Sie keine ARN ein. Geben Sie nur den Rollennamen ein. Elastic Beanstalk fügt den Rollennamen mit den richtigen Werten als Präfix hinzu, um den resultierenden ARN intern zu erstellen.

Name	Beschreibung
`-ls` *load-balancer* oder `--shared-lb` *load-balancer*	Konfigurieren Sie die Umgebung für die Verwendung eines freigegebenen Load Balancer. Geben Sie den Namen oder ARN eines freigebbaren Load Balancers in Ihrem Konto an. Dabei muss es sich um einen Application Load Balancer handeln, den Sie ausdrücklich erstellt haben. Es darf sich nicht um eine Lastenverteilung handeln, der von einer anderen Elastic-Beanstalk-Umgebung erstellt wurde. Weitere Informationen finden Sie unter Geteile Application Load Balancer. Parameterbeispiele: • `FrontEndLB` – Ein Load Balancer-Name. • `arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/FrontEndLB/0dbf78d8ad96abbc` – Ein Application-Load-Balancer-ARN. Sie können diese Option nur mit angebe `--elb-type application` . Wenn Sie diese Option ohne Angabe von `--shared-lb` angeben, erstellt Elastic Beanstalk einen dedizierten Load Balancer für die Umgebung.

Name	Beschreibung
`-lp` *`port`* oder `--shared-lb-port` *`port`*	Der standardmäßige Listener-Port des gemeinsam genutzten Load Balancers für diese Umgebung. Elastic Beanstalk fügt eine Listener-Regel hinzu, die den gesamten Datenverkehr von diesem Listener an den Standardumgebungsprozess weiterleitet. Weitere Informationen finden Sie unter Geteile Application Load Balancer. Typ: Zahl (Ganzzahl) Standard: `80` Gültige Werte: Jede Ganzzahl, die einen Listener-Port des freigegebenen Load Balancer darstellt.
`--single`	Erstellen Sie die Umgebung mit einer einzigen EC2 Amazon-Instance und ohne Load Balancer. **Warning** Eine Umgebung mit einer einzelnen Instance ist nicht geeignet für die Produktion. Falls die Instance während der Bereitstellung instabil wird, oder Elastic Beanstalk beendet wird und die Instance während eines Konfigurations-Upd ate neu gestartet wird, steht Ihre Anwendung möglicherweise längere Zeit nicht zur Verfügung . Umgebungen mit einer einzelnen Instance können Sie für die Entwicklung, das Testen oder ein Staging nutzen. Für die Produktion verwenden Sie Umgebungen mit Load Balancing.

Name	Beschreibung

`-sm`
 	
oder
 	
`--spot-max-price`
	Der Höchstpreis pro Einheitsstunde in USD, den Sie für eine Spot-Instance zu zahlen bereit sind.
	
	Diese Option kann nur mit der Option `--enable-spot` angegeben werden. Weitere Details finden Sie unter Auto Scaling-Gruppe.
	
	Typ: Zahl (Gleitkomma)
	
	Standard: Der On-Demand-Preis für jeden Instance-Typ. Der Wert der Option ist in diesem Fall `null`.
	
	Gültige Werte: `0.001` bis `20.0`
	
	Empfehlungen zu Höchstpreisoptionen für Spot-Instances finden Sie in der Preisentwicklung für Spot-Instances im EC2 Amazon-Benutzerhandbuch.

`--tags` *`key1=value1[,key2=va`*	Markieren Sie die Ressourcen in Ihrer Umgebung. Tags werden als durch Komma getrennte Liste von `key=value` -Paaren angegeben. Weitere Informationen finden Sie unter Tagging-Umgebungen.

`-t worker`
 	
oder
 	
`--tier worker`
	Erstellen Sie eine Worker-Umgebung. Lassen Sie diese Option zum Erstellen einer Web-Server-Umgebung aus.

`--timeout` *`minutes`*	Legen Sie Anzahl der Minuten fest, bevor ein Timeout für den Befehl auftritt.

Name	Beschreibung
`--version` *version_label*	Gibt die Anwendungsversion an, die Sie für die Umgebung bereitgestellt haben möchten, im lokalen Projektverzeichnis an, anstelle des Anwendungsquellcodes. Typ: Zeichenfolge Zulässige Werte: Eine vorhandene Anwendungsversionsbezeichnung
`--vpc`	Konfigurieren Sie eine VPC für die Umgebung. Wenn Sie diese Option einschließen, werden Sie von der EB-CLI aufgefordert, alle erforderlichen Einstellungen vor dem Start der Umgebung einzugeben.
`--vpc.dbsubnets` *subnet1,subnet2*	Gibt Subnetze für Datenbank-Instances in einer VPC an. Erforderlich, wenn `--vpc.id` angegeben ist.
`--vpc.ec2subnets` *subnet1,subnet2*	Spezifiziert Subnetze für EC2 Amazon-Instances in einer VPC. Erforderlich, wenn `--vpc.id` angegeben ist.
`--vpc.elbpublic`	Startet den Elastic-Load-Balancing-Lastenverteilung in einem öffentlichen Subnetz in Ihrer VPC. Diese Option können Sie nicht mit den Optionen `--tier worker` oder `--single` angeben.
`--vpc.elbsubnets` *subnet1,subnet2*	Gibt Subnetze für den Elastic-Load-Balancing-Last enverteilung in einer VPC an. Diese Option können Sie nicht mit den Optionen `--tier worker` oder `--single` angeben.
`--vpc.id` *ID*	Startet Ihre Umgebung in der angegeben VPC.

Name	Beschreibung
`--vpc.publicip`	Startet Ihre EC2 Amazon-Instances in einem öffentlichen Subnetz in Ihrer VPC. Diese Option können Sie nicht mit der Option `--tier worker` angeben.
`--vpc.securitygroups` *securitygroup1,securitygroup2*	Gibt die Sicherheitsgruppe an. IDs Erforderlich, wenn `--vpc.id` angegeben ist.
Allgemeine Optionen	

Ausgabe

Wenn der Befehl erfolgreich war, werden Sie aufgefordert, Fragen zu beantworten und gibt den Status des erstellten Vorgangs zurück. Wenn es Probleme beim Starten gibt, können Sie über die eb events-Operation weitere Details erhalten.

Wenn Sie die CodeBuild Unterstützung in Ihrer Anwendung aktiviert haben, eb create werden Informationen angezeigt, CodeBuild während Ihr Code erstellt wird. Informationen zur CodeBuild Unterstützung in Elastic Beanstalk finden Sie unter. Verwenden der EB-CLI mit AWS CodeBuild

Beispiele

Im folgenden Beispiel wird eine Umgebung im interaktiven Modus erstellt.

```
$ eb create
Enter Environment Name
(default is tmp-dev): ENTER
Enter DNS CNAME prefix
(default is tmp-dev): ENTER
Select a load balancer type
1) classic
2) application
3) network
(default is 2): ENTER
Environment details for: tmp-dev
```

```
  Application name: tmp
  Region: us-east-2
  Deployed Version: app-141029_145448
  Environment ID: e-um3yfrzq22
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev.elasticbeanstalk.com
  Updated: 2014-10-29 21:54:51.063000+00:00
Printing Status:
...
```

Im folgenden Beispiel wird auch eine Umgebung im interaktiven Modus erstellt. In diesem Beispiel enthält das Projektverzeichnis keinen Anwendungscode. Der Befehl stellt eine Beispielanwendung bereit und lädt diese in Ihr lokales Projektverzeichnis herunter.

```
$ eb create
Enter Environment Name
(default is tmp-dev): ENTER
Enter DNS CNAME prefix
(default is tmp-dev): ENTER
Select a load balancer type
1) classic
2) application
3) network
(default is 2): ENTER
NOTE: The current directory does not contain any source code. Elastic Beanstalk is
 launching the sample application instead.
Do you want to download the sample application into the current directory?
(Y/n): ENTER
INFO: Downloading sample application to the current directory.
INFO: Download complete.
Environment details for: tmp-dev
  Application name: tmp
  Region: us-east-2
  Deployed Version: Sample Application
  Environment ID: e-um3yfrzq22
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev.elasticbeanstalk.com
  Updated: 2017-11-08 21:54:51.063000+00:00
Printing Status:
...
```

Mithilfe den folgenden Befehlen können Sie eine Umgebung erstellen, ohne eine Eingabeaufforderung anzuzeigen.

```
$ eb create dev-env
Creating application version archive "app-160312_014028".
Uploading test/app-160312_014028.zip to S3. This may take a while.
Upload Complete.
Application test has been created.
Environment details for: dev-env
  Application name: test
  Region: us-east-2
  Deployed Version: app-160312_014028
  Environment ID: e-6fgpkjxyyi
  Platform: 64bit Amazon Linux 2015.09 v2.0.8 running PHP 5.6
  Tier: WebServer-Standard
  CNAME: UNKNOWN
  Updated: 2016-03-12 01:40:33.614000+00:00
Printing Status:
...
```

Mit dem folgenden Befehl wird eine Umgebung in einer benutzerdefinierten VPC erstellt.

```
$ eb create dev-vpc --vpc.id vpc-0ce8dd99 --vpc.elbsubnets subnet-
b356d7c6,subnet-02f74b0c --vpc.ec2subnets subnet-0bb7f0cd,subnet-3b6697c1 --
vpc.securitygroup sg-70cff265
Creating application version archive "app-160312_014309".
Uploading test/app-160312_014309.zip to S3. This may take a while.
Upload Complete.
Environment details for: dev-vpc
  Application name: test
  Region: us-east-2
  Deployed Version: app-160312_014309
  Environment ID: e-pqkcip3mns
  Platform: 64bit Amazon Linux 2015.09 v2.0.8 running Java 8
  Tier: WebServer-Standard
  CNAME: UNKNOWN
  Updated: 2016-03-12 01:43:14.057000+00:00
Printing Status:
...
```

eb deploy

Description

Stellt das Quell-Bundle der Anwendung aus dem initialisierten Projektverzeichnis für die laufende Anwendung bereit.

Wenn Git installiert ist, verwendet EB-CLI den `git archive`-Befehl zum Erstellen einer `.zip`-Datei aus dem Inhalt des letzten `git commit`-Befehls.

Wenn jedoch `.ebignore` in Ihrem Projektverzeichnis vorhanden ist, verwendet die EB-CLI keine git-Befehle und -Semantik, um Ihr Quell-Bundle zu erstellen. Das bedeutet, EB-CLI ignoriert die in `.ebignore` angegebenen Dateien und bindet alle anderen Dateien ein. Insbesondere bindet sie nicht festgeschriebene Quelldateien ein.

> **Note**
>
> Sie können die EB-CLI so konfigurieren, dass sie ein Artefakt aus Ihrem Erstellungsprozess bereitstellt, anstatt eine ZIP-Datei Ihres Projektordners zu erstellen. Details dazu finden Sie unter Bereitstellen eines Artefakts anstelle des Projektordners.

Syntax

eb deploy

eb deploy ***environment-name***

Optionen

Name	Beschreibung
`-l` *`version_label`* oder `--label` *`version_label`*	Geben Sie eine Bezeichnung für die Version an, die die EB-CLI erstellt. Wenn die Bezeichnung bereits verwendet wurde, stellt die EB-CLI die frühere Version mit dieser Bezeichnung erneut bereit. Typ: Zeichenfolge

Name	Beschreibung
`--env-group-suffix` *groupname*	Gruppenname, der an den Umgebungsnamen angefügt wird. Nur zur Verwendung mit Compose-Umgebungen.

`-m "`*version_description*` "`
 	
oder
 	
`--message "`*version_description*` "`
	Die Beschreibung für die Anwendungsversion, in doppelte Anführungszeichen gesetzt.
	
	Typ: Zeichenfolge

`--modules` *component-a component-b*	Liste der zu aktualisierenden Komponenten. Nur zur Verwendung mit Compose-Umgebungen.

`-p`
 	
oder
 	
`--process`
	Vorverarbeiten und Validieren des Umgebungsmanifests und der Konfigurationsdateien im Quell-Bundle. Durch die Validierung der Konfigurationsdateien können Probleme vor der Bereitstellung der Anwendungsversion für eine Umgebung identifiziert werden.

`--source codecommit/` *repository-name*`/`*branch-name*	CodeCommit Endlager und Filiale.
`--staged`	Bereitstellen von Dateien, die im Git-Index und nicht im HEAD-Commit zwischengespeichert sind.
`--timeout` *minutes*	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.
`--version` *version_label*	Eine vorhandene Anwendungsversion zum Bereitstellen. Typ: Zeichenfolge
Allgemeine Optionen	

Ausgabe

Bei Erfolg gibt der Befehl den Status des `deploy`-Vorgangs aus.

Wenn Sie die CodeBuild Unterstützung in Ihrer Anwendung aktiviert haben, eb deploy werden Informationen angezeigt, CodeBuild während Ihr Code erstellt wird. Informationen zur CodeBuild Unterstützung in Elastic Beanstalk finden Sie unter. Verwenden der EB-CLI mit AWS CodeBuild

Beispiel

Im folgenden Beispiel wird die aktuelle Anwendung bereitgestellt.

```
$ eb deploy
2018-07-11 21:05:22    INFO: Environment update is starting.
2018-07-11 21:05:27    INFO: Deploying new version to instance(s).
2018-07-11 21:05:53    INFO: New application version was deployed to running EC2
 instances.
2018-07-11 21:05:53    INFO: Environment update completed successfully.
```

eb events

Description

Gibt die aktuellen Ereignisse für die Umgebung zurück.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, gibt dieser Befehl auch die aktuellen Ereignisse für die Builder-Umgebung zurück.

Syntax

eb events

eb events ***environment-name***

Optionen

Name	Beschreibung
`-f`	Streamt Ereignisse. Wenn Sie den Vorgang abbrechen möchten, drücken Sie STRG+C.

Name	Beschreibung
oder `--follow`	

Ausgabe

Bei Erfolg gibt der Befehl die aktuellen Ereignisse zurück.

Beispiel

Das folgende Beispiel gibt die aktuellen Ereignisse zurück.

```
$ eb events
2014-10-29 21:55:39     INFO    createEnvironment is starting.
2014-10-29 21:55:40     INFO    Using elasticbeanstalk-us-west-2-111122223333 as Amazon
 S3 storage bucket for environment data.
2014-10-29 21:55:57     INFO    Created load balancer named: awseb-e-r-AWSEBLoa-
NSKUOK5X6Z9J
2014-10-29 21:56:16     INFO    Created security group named: awseb-e-rxgrhjr9bx-stack-
AWSEBSecurityGroup-1UUHU5LZ20ZY7
2014-10-29 21:57:18     INFO    Waiting for EC2 instances to launch. This may take a
 few minutes.
2014-10-29 21:57:18     INFO    Created Auto Scaling group named: awseb-e-rxgrhjr9bx-
stack-AWSEBAutoScalingGroup-1TE320ZCJ9RPD
2014-10-29 21:57:22     INFO    Created Auto Scaling group policy named:
 arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:2cced9e6-859b-421a-
be63-8ab34771155a:autoScalingGroupName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingGroup-1TE320ZCJ9RPD:policyName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingScaleUpPolicy-1I2ZSNVU4APRY
2014-10-29 21:57:22     INFO    Created Auto Scaling group policy named:
 arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:1f08b863-
bf65-415a-b584-b7fa3a69a0d5:autoScalingGroupName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingGroup-1TE320ZCJ9RPD:policyName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingScaleDownPolicy-1E3G7PZKZPSOG
2014-10-29 21:57:25     INFO    Created CloudWatch alarm named: awseb-e-rxgrhjr9bx-
stack-AWSEBCloudwatchAlarmLow-VF5EJ549FZBL
2014-10-29 21:57:25     INFO    Created CloudWatch alarm named: awseb-e-rxgrhjr9bx-
stack-AWSEBCloudwatchAlarmHigh-LA9YEW3O6WJO
2014-10-29 21:58:50     INFO    Added EC2 instance 'i-c7ee492d' to Auto ScalingGroup
 'awseb-e-rxgrhjr9bx-stack-AWSEBAutoScalingGroup-1TE320ZCJ9RPD'.
```

```
2014-10-29 21:58:53     INFO    Successfully launched environment: tmp-dev
2014-10-29 21:59:14     INFO    Environment health has been set to GREEN
2014-10-29 21:59:43     INFO    Adding instance 'i-c7ee492d' to your environment.
```

eb health

Description

Gibt den aktuellen Zustand für die Umgebung zurück.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, gibt dieser Befehl auch den aktuellen Zustand für die Builder-Umgebung zurück.

Syntax

eb health

eb health ***environment-name***

Optionen

Name	Beschreibung

`-r`
oder
`--refresh`
	Zeigen Sie Zustandsinformationen interaktiv an und aktualisieren alle 10 Sekunden, wenn neue Informationen gemeldet werden.

`--mono`	Keine Anzeigefarbe in der Ausgabe.

Ausgabe

Bei Erfolg gibt der Befehl den aktuellen Zustand zurück.

Beispiel

Das folgende Beispiel gibt die aktuellsten Zustandsinformationen für eine Linux-Umgebung zurück.

```
~/project $ eb health
```

```
 elasticBeanstalkExa-env                                Ok
 2015-07-08 23:13:20
WebServer
 Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        5        0        0        0        0        0        0

  instance-id   status     cause
                                      health
    Overall     Ok
  i-d581497d    Ok
  i-d481497c    Ok
  i-136e00c0    Ok
  i-126e00c1    Ok
  i-8b2cf575    Ok

  instance-id   r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75     p50
   p10                                requests
    Overall     671.8   100.0    0.0    0.0    0.0    0.003    0.002    0.001   0.001
0.000
 i-d581497d    143.0    1430      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-d481497c    128.8    1288      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-136e00c0    125.4    1254      0      0      0    0.004    0.002    0.001   0.001
0.000
 i-126e00c1    133.4    1334      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-8b2cf575    141.2    1412      0      0      0    0.003    0.002    0.001   0.001
0.000

 instance-id   type       az   running     load 1  load 5      user%  nice%  system%
idle%   iowait%                            cpu
 i-d581497d    t2.micro   1a   12 mins        0.0    0.04        6.2    0.0      1.0
92.5       0.1
 i-d481497c    t2.micro   1a   12 mins       0.01    0.09        5.9    0.0      1.6
92.4       0.1
 i-136e00c0    t2.micro   1b   12 mins       0.15    0.07        5.5    0.0      0.9
93.2       0.0
 i-126e00c1    t2.micro   1b   12 mins       0.17    0.14        5.7    0.0      1.4
92.7       0.1
 i-8b2cf575    t2.micro   1c   1 hour        0.19    0.08        6.5    0.0      1.2
92.1       0.1
```

```
  instance-id   status     id   version              ago
                                  deployments
  i-d581497d    Deployed   1    Sample Application   12 mins
  i-d481497c    Deployed   1    Sample Application   12 mins
  i-136e00c0    Deployed   1    Sample Application   12 mins
  i-126e00c1    Deployed   1    Sample Application   12 mins
  i-8b2cf575    Deployed   1    Sample Application   1 hour
```

eb init

Description

Legt Standardwerte für Elastic-Beanstalk-Anwendungen fest, die mit der EB-CLI erstellt wurden, indem Sie eine Reihe von Fragen beantworten müssen.

> **Note**
>
> Die Werte, die Sie mit festlegen, eb init gelten für das aktuelle Verzeichnis und Repository auf dem aktuellen Computer.
> Der Befehl erstellt eine Elastic Beanstalk Beanstalk-Anwendung in Ihrem Konto. Um eine Elastic-Beanstalk-Umgebung zu erstellen, führen Sie eb create nach Ausführung von eb init aus.

Syntax

eb init

eb init *application-name*

Optionen

Wenn Sie eb init ohne Angeben der `--platform`-Option ausführen, werden Sie von der EB-CLI aufgefordert, einen Wert für jede Einstellung einzugeben oder auszuwählen.

> **Note**
>
> Damit mit eb init ein neues Schlüsselpaar erstellt werden kann, müssen Sie `ssh-keygen` auf Ihrem lokalen Computer installiert haben und über die Befehlszeile darauf zugreifen können.

Name	Beschreibung	

`-i`
 	
`--interactive`
	Zwingt EB-CLI, Sie aufzufordern, einen Wert für jede eb init-Befehlsoption anzugeben.
	
	Note
	Der `init`-Befehl fordert Sie auf, Werte für eb init-Befehlsoptionen anzugeben, die keinen (Standard-)Wert haben. Nach der ersten Ausführung des eb init-Befehls in einem Verzeichnis fordert EB-CLI Sie möglicherweise nicht hinsichtlich Befehlsoptionen auf. Aus diesem Grund verwenden Sie die `--interactive` -Option, wenn Sie eine Einstellung ändern möchten, die Sie zuvor festgelegt haben.	

`-k` *`keyname`* `--keyname` *`keyname`*	Der Name des EC2 Amazon-Schlüsselpaars, das mit dem Secure Shell (SSH) -Client verwendet werden soll, um sich sicher bei den EC2 Amazon-Instances anzumelden, auf denen Ihre Elastic Beanstalk Beanstalk-Anwendung ausgeführt wird.	
`--modules` *`folder-1 folder-2`*	Liste der untergeordneten Verzeichnisse zur Initialisierung. Nur zur Verwendung mit Compose-Umgebungen.	
`-p` *`platform-version`* `--platform` *`platform-version`*	Die zu verwendende Plattformversion . Sie können eine Plattform, Name und Version einer Plattform, einen Plattformzweig, einen Lösungs-Stack-Namen oder einen Lösungs-Stack-ARN angeben. Zum Beispiel: • `php`, `PHP`, `node.js` – Die aktuelle Plattformversion für die angegebene Plattform	

Name	Beschreibung	
	• `php-7.2`, `"PHP 7.2"` – Die empfohlene (typische rweise neueste) PHP 7.2-Plattformversion • `"PHP 7.2 running on 64bit Amazon Linux"` – Die empfohlene (typischerweise neueste) PHP-Plattformversion in diesem Plattformzweig • `"64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1"` – Die PHP-Plattformversion, die von diesem Lösungs-Stack-Namen angegeben wird • `"arn:aws:elasticbeanstalk:us-east-2::platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3"` – Die PHP-Platt formversion, die durch diesen Lösungs-Stack-ARN angegeben wird Verwenden Sie `eb platform list`, um eine Liste der verfügbaren Konfigurationen anzuzeigen. Geben Sie die Option `--platform` an, um die interaktive Konfiguration zu überspringen. **Note** Wenn Sie diese Option angeben, werden Sie von der EB-CLI nicht zur Eingabe von Werten für andere Optionen aufgefordert. Stattdessen werden die Standardwerte für die einzelnen Optionen verwendet. Sie können Optionen für alles angeben, für das Sie keine Standardw erte verwenden möchten.	
`--source codecommit/` *`repository-name`*`/`*`branch-name`*	CodeCommit Repository und Zweig.	

Name	Beschreibung	
`--tags key1=value1`	Markieren Sie Ihre Anwendung mit Tags. Tags werden als durch Komma getrennte Liste von `key=value` -Paaren angegeben. Weitere Details finden Sie unter Markieren von Anwendungen.	
Allgemeine Optionen		

CodeBuild Unterstützung

Wenn Sie eb init in einem Ordner mit einer buildspec.yml-Datei ausführen, analysiert Elastic Beanstalk die Datei auf einen eb_codebuild_settings-Eintrag mit für Elastic Beanstalk spezifischen Optionen. Informationen zur CodeBuild Unterstützung in Elastic Beanstalk finden Sie unter. Verwenden der EB-CLI mit AWS CodeBuild

Ausgabe

Wenn der Befehl erfolgreich war, werden Sie durch die Einrichtung einer neuen Elastic-Beanstalk-Anwendung mithilfe einer Reihe von Anweisungen geführt.

Beispiel

Die folgende Beispielanforderung initialisiert EB-CLI und fordert Sie auf, Informationen zu Ihrer Anwendung einzugeben. Ersetzen Sie *placeholder* Text durch Ihre eigenen Werte.

```
$ eb init -i
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
(default is 3): 3
```

```
Select an application to use
1) HelloWorldApp
2) NewApp
3) [ Create new Application ]
(default is 3): 3

Enter Application Name
(default is "tmp"):
Application tmp has been created.

It appears you are using PHP. Is this correct?
(y/n): y

Select a platform branch.
1) PHP 7.2 running on 64bit Amazon Linux
2) PHP 7.1 running on 64bit Amazon Linux (Deprecated)
3) PHP 7.0 running on 64bit Amazon Linux (Deprecated)
4) PHP 5.6 running on 64bit Amazon Linux (Deprecated)
5) PHP 5.5 running on 64bit Amazon Linux (Deprecated)
6) PHP 5.4 running on 64bit Amazon Linux (Deprecated)
(default is 1): 1
Do you want to set up SSH for your instances?
(y/n): y

Select a keypair.
1) aws-eb
2) [ Create new KeyPair ]
(default is 2): 1
```

eb labs

Description

Unterbefehle zur eb labs Unterstützung work-in-progress oder experimentellen Funktionalität. Diese Befehle werden möglicherweise in künftigen Versionen der EB-CLI entfernt oder überarbeitet und bleiben nicht garantiert kompatibel.

Für eine Liste der verfügbaren Unterbefehle und Beschreibungen führen Sie au eb labs --help.

eb list

Description

Listet alle Umgebungen in der aktuellen Anwendung oder alle Umgebungen in allen Anwendungen auf, wie in der `--all`-Option angegeben.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, listet dieser Befehl auch die Builder-Umgebung auf.

Syntax

eb list

Optionen

Name	Beschreibung

`-a`
oder
`--all`
	Listet alle Umgebungen von allen Anwendungen auf.

`-v`
oder
`--verbose`
	Bietet detaillierte Informationen über alle Umgebungen, einschließlich Instances.

Allgemeine Optionen	

Ausgabe

Wenn der Befehl erfolgreich war, gibt er eine Liste der Umgebungsnamen zurück, in denen Ihre aktuelle Umgebung mit einem Sternchen (*) gekennzeichnet ist.

Beispiel 1

Im folgenden Beispiel werden Ihre Umgebungen aufgelistet und angegeben, dass "tmp-dev" Ihre Standard-Umgebung ist.

```
$ eb list
* tmp-dev
```

Beispiel 2

Im folgenden Beispiel werden Ihre Umgebungen mit weiteren Details aufgelistet.

```
$ eb list --verbose
Region: us-west-2
Application: tmp
    Environments: 1
        * tmp-dev : ['i-c7ee492d']
```

eb local

Description

Verwenden Sie eb local run, um den Anwendungscontainer lokal in Docker auszuführen. Prüfen Sie den Status des Anwendungscontainers mit eb local status. Öffnen Sie die Anwendung mit in einem Web-Browser eb local open. Rufen Sie den Speicherort der Anwendungsprotokolle mit a eb local logs.

Mit eb local setenv und eb local printenv können Sie Umgebungsvariablen einrichten und anzeigen, welche für die lokal mit eb local run ausgeführten Docker-Container bereitgestellt werden.

Sie müssen alle eb local-Befehle im Projektverzeichnis einer Docker-Anwendung ausführen, die mit eb init als EB-CLI-Repository initialisiert wurde.

> **Note**
>
> Verwenden Sie eb local auf einem lokalen Computer mit Linux oder macOS. Der Befehl unterstützt Windows nicht.
> Bevor Sie den Befehl unter macOS verwenden, installieren Sie Docker für Mac und stellen Sie sicher, dass boot2docker nicht installiert ist (oder sich nicht im Ausführungspfad) befindet.

Der eb local-Befehl versucht boot2docker zu verwenden, sofern vorhanden, funktioniert damit jedoch unter macOS nicht gut.

Syntax

eb local run

eb local status

eb local open

eb local logs

eb local setenv

eb local printenv

Optionen

eb local run

Name	Beschreibung
`--envvars key1=value1,key2=value2`	Legt Umgebungsvariablen fest, die von der EB-CLI an die lokalen Docker-Container übergeben werden. In Multicontainer-Umgebungen werden alle Variablen an alle Container übertragen.
`--port hostport`	Ordnet einen Host-Port dem entsprechenden Container-Port zu. Wenn Sie diese Option nicht angeben, verwendet die EB-CLI denselben Port für Host und Container. Diese Option funktioniert nur mit Docker-Plattformanwendungen. Sie gilt nicht für die Multicontainer Docker-Plattform.
Allgemeine Optionen	

eb local status

eb local open

eb local logs

eb local setenv

eb local printenv

Name	Beschreibung
Allgemeine Optionen	

Ausgabe

eb local run

Statusmeldungen von Docker. Bleibt aktiv, solange die Anwendung ausgeführt wird. Drücken Sie Strg +C, um die Anwendung zu stoppen.

eb local status

Der Status der einzelnen Container, die von der Anwendung verwendet werden (ausgeführt oder nicht ausgeführt).

eb local open

Öffnet die Anwendung in einem Web-Browser und wird beendet.

eb local logs

Der Speicherort der Protokolle, die im Projektverzeichnis von lokal unter eb local run ausgeführten Anwendungen generiert werden.

eb local setenv

Keine

eb local printenv

Die Namen und Werte der mit festgelegten Umgebungsvariable eb local setenv.

Beispiele

eb local run

```
~/project$ eb local run
Creating elasticbeanstalk_phpapp_1...
Creating elasticbeanstalk_nginxproxy_1...
Attaching to elasticbeanstalk_phpapp_1, elasticbeanstalk_nginxproxy_1
phpapp_1     | [23-Apr-2015 23:24:25] NOTICE: fpm is running, pid 1
phpapp_1     | [23-Apr-2015 23:24:25] NOTICE: ready to handle connections
```

eb local status

Zeigen Sie den Status der lokalen Container an:

```
~/project$ eb local status
Platform: 64bit Amazon Linux 2014.09 v1.2.1 running Multi-container Docker 1.3.3
 (Generic)
Container name: elasticbeanstalk_nginxproxy_1
Container ip: 127.0.0.1
Container running: True
Exposed host port(s): 80
Full local URL(s): 127.0.0.1:80

Container name: elasticbeanstalk_phpapp_1
Container ip: 127.0.0.1
Container running: True
Exposed host port(s): None
Full local URL(s): None
```

eb local logs

Zeigen Sie den Protokollpfad für das aktuelle Projekt an:

```
~/project$ eb local logs
Elastic Beanstalk will write logs locally to /home/user/project/.elasticbeanstalk/logs/
local.
Logs were most recently created 3 minutes ago and written to /home/user/
project/.elasticbeanstalk/logs/local/150420_234011665784.
```

eb local setenv

Legen Sie Umgebungsvariablen zur Verwendung mit eb local run fest.

```
~/project$ eb local setenv PARAM1=value
```

Drucken Sie die mit eb local setenv festgelegten Umgebungsvariablen.

```
~/project$ eb local printenv
Environment Variables:
PARAM1=value
```

eb logs

Description

Der eb logs Befehl hat drei unterschiedliche Zwecke: das Aktivieren oder Deaktivieren des Log-Streamings in CloudWatch Logs, das Abrufen von Instanzprotokollen oder CloudWatch Log-Logs und das Anfordern einer KI-gestützten Analyse Ihrer Umgebung. Mit der Option `--cloudwatch-logs` (`-cw`) aktiviert oder deaktiviert der Befehl das Protokoll-Streaming. Mit der Option `--analyze` (`-ai`) fordert der Befehl eine KI-gestützte Analyse der Protokolle, Ereignisse und des Instanzzustands Ihrer Umgebung an. Ohne eine dieser Optionen werden Protokolle abgerufen.

Beim Abrufen der Protokolle geben Sie die Option `--all`, `--zip` oder `--stream` an, um vollständige Protokolle abzurufen. Wenn Sie keine dieser Optionen angeben, ruft Elastic Beanstalk Protokollfragmente ab.

Der Befehl verarbeitet Protokolle für die angegebene oder die Standardumgebung. Die entsprechenden Protokolle sind je nach Containertyp unterschiedlich. Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, verarbeitet dieser Befehl auch die Protokolle für die Builder-Umgebung.

Weitere Informationen finden Sie unter the section called “CloudWatch Logs”. Weitere Informationen zur KI-gestützten Analyse finden Sie unter. KI-gestützte Umgebungsanalyse

Syntax

So aktivieren oder deaktivieren Sie das Log-Streaming zu CloudWatch Logs:

```
eb logs --cloudwatch-logs [enable | disable] [--cloudwatch-log-source instance |
 environment-health | all] [environment-name]
```

So rufen Sie Instance-Protokolle ab:

```
eb logs [-all | --zip | --stream] [--cloudwatch-log-source instance] [--
instance instance-id] [--log-group log-group] [environment-name]
```

So rufen Sie Umgebungszustand-Protokolle ab:

```
eb logs [-all | --zip | --stream] --cloudwatch-log-source environment-health
 [environment-name]
```

Um eine KI-gestützte Analyse anzufordern:

```
eb logs --analyze [environment-name]
```

Optionen

Name	Beschreibung

`-cw [enable | disable]`
 	
oder
 	
`--cloudwatch-logs [enable | disable]`
	Aktiviert oder deaktiviert das Log-Streaming zu CloudWatch Logs. Falls kein Argument angegeben wird, ist das Streaming der Protokolle aktiviert. Wenn die Option `--cloudwatch-log-source` (`-cls`) nicht zusätzlich angegeben ist, ist das Instance-Protokoll-Streaming aktiviert oder deaktiviert.

`-cls instance | environment-health | all`
 	
oder
 	
`--cloudwatch-log-source instance | environment-health | all`
	Gibt die Quelle der Protokolle an, wenn mit CloudWatch Protokollen gearbeitet wird. In der Form „Aktivieren“ oder „Deaktivieren“ des Befehls sind dies die Protokolle, für die das Log-Streaming aktiviert oder deaktiviert CloudWatch werden soll. In der Abrufform des Befehls sind dies die Protokolle, die aus CloudWatch den Protokollen abgerufen werden sollen.
	
	Zulässige Werte:
	
	• Mit `--cloudwatch-logs` (Aktivieren oder Deaktivieren) – `instance | environment-health | all`
	• Ohne `--cloudwatch-logs` (Abrufen) – `instance | environment-health`
	
	Bedeutungen der Werte:

Name	Beschreibung
	• `instance` (Standard) – Instance-Protokolle • `environment-health` – Umgebungszustand-Protokolle (wird nur unterstützt, wenn erweiterte Zustandsberichte in der Umgebung aktiviert sind) • `all` – Beide Protokollquellen

`-a`
 	
oder
 	
`--all`
	Ruft vollständige Protokolle ab und speichert sie im Verzeichnis `.elasticbeanstalk/logs` .

`-z`
 	
oder
 	
`--zip`
	Ruft vollständige Protokolle ab, komprimiert sie in einer `.zip`-Datei und speichert diese im Verzeichnis `.elasticbeanstalk/logs` .

`--stream`	Streamt vollständige Protokolle (fortlaufende Ausgaben). Mit dieser Option wird der Befehl ausgeführt, bis Sie ihn abbrechen (drücken Sie hierzu **`Ctrl+C`**).
`-i` *`instance-id`* oder `--instance` *`instance-id`*	Ruft nur Protokolle für die angegebene Instance ab.

Name	Beschreibung

`-g log-group`
 	
oder
 	
`--log-group log-group`
	Gibt die Protokollgruppe CloudWatch Logs an, aus der Protokolle abgerufen werden sollen. Die Option ist nur gültig, wenn das Streaming von CloudWatch Instanzprotokollen in Logs aktiviert ist.
	
	Wenn das Instance-Protokoll-Streaming aktiviert ist und Sie nicht die `--log-group` -Option angeben, ist die Standard-Protokollgruppe eine der folgenden:
	
	• Amazon Linux 2 – `/aws/elasticbeanstalk/ environment-name /var/log/eb-engine.log`
	• Windows-Plattform – `/aws/elasticbeanstalk/ environment-name /EBDeploy-Log`
	• Amazon Linux-AMI (AL1) — `/aws/elasticbeanstalk/ environment-name /var/log/eb-activity.log`
	
	Note
	Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.
	
	Weitere Informationen über die Protokollgruppe für jede Protokolldatei finden Sie unter So richtet Elastic Beanstalk Logs ein CloudWatch.

Name	Beschreibung

`-ai`
 	
oder
 	
`--analyze`
	Fordert eine KI-gestützte Analyse der Logs, Ereignisse und des Instance-Zustands Ihrer Umgebung an. Die Analyse verwendet Amazon Bedrock, um die Grundursachen zu identifizieren und Lösungen für umweltbedingte Gesundheitsprobleme zu empfehlen. Der Befehl sendet die Anfrage, wartet, bis die Analyse abgeschlossen ist, und zeigt dann die Ergebnisse an.
	
	Diese Option ist nicht kompatibel mit`--instance` ,`--all`, `--zip``--log-group` , oder`--cloudwatch-logs` .
	
	Informationen zu den Voraussetzungen und erforderlichen Berechtigungen finden Sie unterKI-gestützte Umgebungs analyse.

Allgemeine Optionen	

Ausgabe

Standardmäßig werden die Protokolle direkt im Terminal angezeigt. Verwendet ein Paginierungs-Programm, um die Ausgabe anzuzeigen. Drücken Sie zum Beenden **`Q`** oder **`q`**.

Zeigt mit `--stream` vorhandene Protokolle im Terminal an und führt sie weiterhin aus. Drücken Sie **`Ctrl+C`** zum Beenden.

Speichert mit `--all` und `--zip` die Protokolle in lokalen Dateien und zeigt den Speicherort der Datei.

Mit`--analyze`, zeigt die KI-generierte Analyse direkt im Terminal an, nachdem die Analyse abgeschlossen ist.

Beispiele

Im folgenden Beispiel wird das Streaming von CloudWatch Instanzprotokollen zu Logs aktiviert.

```
$ eb logs -cw enable
Enabling instance log streaming to CloudWatch for your environment
After the environment is updated you can view your logs by following the link:
```

```
https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#logs:prefix=/aws/
elasticbeanstalk/environment-name/
Printing Status:
2018-07-11 21:05:20    INFO: Environment update is starting.
2018-07-11 21:05:27    INFO: Updating environment environment-name's configuration
 settings.
2018-07-11 21:06:45    INFO: Successfully deployed new configuration to environment.
```

Das folgende Beispiel ruft die Instance-Protokolle in eine `.zip` Datei ab.

```
$ eb logs --zip
Retrieving logs...
Logs were saved to /home/workspace/environment/.elasticbeanstalk/logs/150622_173444.zip
```

Im folgenden Beispiel wird eine KI-gestützte Analyse der Umgebung angefordert.

```
$ eb logs --analyze
Analyzing environment...
No critical issues detected. Environment appears healthy.

---
Note: This analysis was generated by AI.
```

eb migrate

Description

Migriert Internet Information Services (IIS) -Sites und -Anwendungen von einem Windows-Server zu Elastic Beanstalk. Der Befehl packt Ihre Anwendungen, behält ihre Konfigurationen bei und stellt sie in einer neuen Elastic Beanstalk Beanstalk-Umgebung bereit.

Weitere Informationen zur Migration Ihrer IIS-Sites und -Anwendungen finden Sie unter. IIS-Anwendungen migrieren

> **Note**
>
> Bevor Sie diesen Befehl verwenden, stellen Sie sicher, dass Ihr System die folgenden Anforderungen erfüllt:
>
> - Internetinformationsdienste (IIS) Version 7.0 oder höher

- Web Deploy 3.6 oder höher ist installiert
- Administratorrechte auf dem Windows-Server
- AWS Anmeldeinformationen, die mit den entsprechenden Berechtigungen konfiguriert wurden
- Ihr Quellserver hat ausgehenden Internetzugang zu AWS Diensten.

In den folgenden Schritten wird der Migrationsprozess zusammengefasst:

1. Entdecken Sie IIS-Websites und ihre Konfigurationen.
2. Inhalt und Konfiguration der Paketanwendung.
3. Erstellen Sie eine Elastic Beanstalk Beanstalk-Umgebung und -Anwendung.
4. Stellen Sie die Anwendung mit beibehaltenen Einstellungen bereit.

Der Befehl erstellt Migrationsartefakte in einem strukturierten Verzeichnis, wie in der folgenden Liste dargestellt:

```
C:\migration_workspace\
### .\migrations\latest\
    ### upload_target.zip
    ### upload_target\
        ### [SiteName].zip                     # One ZIP per default application of
 IIS site
        ### [SiteName-ApplicationName].zip     # One ZIP per additional application
        ### aws-windows-deployment-manifest.json
        ### ebmigrateScripts\
            ### site_installer.ps1             # Site installation scripts
            ### permission_handler.ps1         # Permission management
            ### other helper scripts
```

Wird verwendeteb migrate cleanup, um diese Artefakte zu verwalten.

Syntax

eb migrate ***[options]***

eb migrate explore ***[options]***

eb migrate cleanup ***[options]***

Wenn es ohne Argumente ausgeführt wird, wird eb migrate es im nicht interaktiven Modus ausgeführt. Um es im interaktiven Modus auszuführen, führen Sie den Befehl auseb migrate --interactive.

Der Befehl für den interaktiven Modus fordert Sie zur Eingabe der folgenden Informationen auf:

- Auswahl der zu migrierenden IIS-Sites
- Umgebungs- und Anwendungsnamen
- Auswahl der Plattformversion
- Instanztyp und andere Konfigurationsoptionen

Unterbefehle

entdecken

Der eb migrate explore Unterbefehl untersucht Ihren IIS-Server und listet die verfügbaren Sites auf.

Verwenden Sie diesen Befehl, um die folgenden Informationen anzuzeigen:

- Alle IIS-Sites auf dem Server anzeigen
- Untersuchen Sie mit `--verbose` die detaillierte Konfiguration, einschließlich:
  - Site-Bindungen und Ports
  - Anwendungspools
  - Virtuelle Verzeichnisse und ihre physischen Pfade
  - Einstellungen für die Authentifizierung

```
PS C:\migrations_workspace >  eb migrate explore
Default Web Site
Site2
site3
router
```

```
PS C:\migrations_workspace >  eb migrate explore --verbose
1: Default Web Site:
  - Bindings:
    - *:8083:
  - Application '/':
```

```
    - Application Pool: DefaultAppPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\inetpub\wwwroot
        - Logon Method: ClearText
  - Application '/dotnet-6-0':
    - Application Pool: DefaultAppPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\inetpub\AspNetCoreWebApps\CoreWebApp-6-0
        - Logon Method: ClearText
  - Application '/dotnet-8-0':
    - Application Pool: DefaultAppPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\inetpub\AspNetCoreWebApps\CoreWebApp-8-0
        - Logon Method: ClearText
2: Site2:
  - Bindings:
    - *:8081:
...
```

Bereinigen

Der eb migrate cleanup Unterbefehl verwaltet Migrationsartefakte mit den folgenden Aktionen:

- Beibehaltung der letzten erfolgreichen Migration in `./migrations/latest`
- Ältere Migrationsverzeichnisse werden entfernt
- Pflege kritischer Konfigurationsdateien

```
PS C:\migrations_workspace >   eb migrate cleanup
Are you sure you would like to cleanup older artifacts within `./migrations/`? (y/N):
```

Wird verwendet`--force`, um Bestätigungsaufforderungen während der Bereinigung zu überspringen.

```
PS C:\migrations_workspace >   eb migrate cleanup --force
```

Optionen

Keine dieser Optionen ist erforderlich. Wenn Sie eb migrate ohne Optionen ausführen, wird die EB-CLI im nicht interaktiven Modus ausgeführt. Mit eb migrate --interactive fordert Sie die EB CLI auf, einen Wert für die erforderlichen Einstellungen einzugeben oder auszuwählen.

Name	Beschreibung
`-a` *`application-name`* oder `--application-name` *`application-name`*	Name für die neue Elastic Beanstalk Beanstalk-Anwendung. Typ: Zeichenfolge Standard: App EBMigrated
`--archive` *`directory-or-zip`*	Das Verzeichnis oder die ZIP-Datei, die den Quellcode enthält, der zuvor von generiert wurdeeb migrate --archive-only. Verwenden Sie diese Option, um ein zuvor erstelltes Migrationspaket bereitzustellen. Beispiel: `--archive .\migrations\latest\upload_target` oder `--archive .\migrations\latest\upload_target.zip`

`-ao`
 	
oder
 	
`--archive-only`
	Erstellen Sie nur das Zielarchivverzeichnis ohne Bereitstellung.
	
	Das resultierende Verzeichnis kann eb migrate mit der `archive` Option oder manuell bereitgestellt werdeneb deploy.

`-c` *`subdomain-name`* oder `--cname` *`subdomain-name`*	Der Name der Subdomain, der dem CNAME-DNS-Eintrag für Ihre migrierte Anwendung vorangestellt werden soll. Typ: Zeichenfolge Standard: Der Umgebungsname

Name	Beschreibung

`-cf`
 	
oder
 	
`--copy-firewall-config`
	Kopieren Sie die Firewallkonfiguration des Quellservers für alle HTTP-Ports mit aktiven Bindungen auf das Ziel.
	
	Erstellt entsprechende Sicherheitsgruppenregeln in AWS.

`-es` *`snapshot-id`* [*`snapshot-id`* ...] oder `--ebs-snapshots` *`snapshot-id`* [*`snapshot-id`* ...]	Durch Kommas getrennte Liste der Amazon EBS-Snaps hots IDs , die der Umgebung zugeordnet werden sollen. Beispiel: `--ebs-snapshots snap-1234567890abc def0, snap-0987654321fedcba1`
`--encrypt-ebs-volumes`	Erzwingen Sie die Verschlüsselung für alle neuen Amazon EBS-Volumes. **Important** Dies ist eine kontoweite Einstellung, die sich auf alle future Amazon EBS-Volume-Erstellungen auswirkt.
`-e` *`environment-name`* oder `--environment-name` *`environment-name`*	Name für die neue Elastic Beanstalk Beanstalk-Umgebung. Typ: Zeichenfolge Standard: Env EBMigrated Einschränkungen: Muss zwischen 4 und 40 Zeichen lang sein. Kann nur Buchstaben, Zahlen und Bindestriche enthalten. Am Anfang und am Ende darf kein Bindestrich stehen.

Name	Beschreibung
`--force`	Überspringen Sie Bestätigungsaufforderungen während des Betriebs. Bei Verwendung mit einem cleanup Unterbefehl werden Migrationsartefakte ohne Bestätigung entfernt.
`-ip` *`profile-name`* oder `--instance-profile` *`profile-name`*	Instance-Profil, das den EC2 Amazon-Instances der Umgebung zugeordnet werden soll. Wenn nicht angegeben, wird ein Standard-Instanzprofil mit Berechtigungen für den Zugriff auf Elastic Beanstalk Beanstalk-Ressourcen erstellt. Weitere Informationen finden Sie unter the section called “Instance-Profil”.
`-i` *`instance-type`* oder `--instance-type` *`instance-type`*	Der EC2 Amazon-Instance-Typ für Ihre Elastic Beanstalk Beanstalk-Umgebung. Typ: Zeichenfolge Standard: c5.2xlarge Die verfügbaren Instance-Typen finden Sie unter EC2 Amazon-Instance-Typen im EC2 Amazon-Benutzerhandbuch.

`-in`
 	
oder
 	
`--interactive`
	Erzwingen Sie den interaktiven Modus für den Migration sprozess.
	
	Fordert zur Eingabe von Konfigurationswerten auf, auch wenn Standardwerte verfügbar sind.

Name	Beschreibung
`-k` *`key-name`* oder `--keyname` *`key-name`*	EC2 Amazon-Schlüsselpaar zur Aktivierung des RDP-Zugriffs auf Umgebungsinstanzen. Nützlich für die Untersuchung von Problemen auf Instanzebene, die in Protokollen nicht sichtbar sind. Gültige Werte: Ein vorhandener Schlüsselpaarname, der bei Amazon registriert ist EC2
`-p` *`platform-version`* oder `--platform` *`platform-version`*	Elastic Beanstalk-Plattformlaufzeit für die Umgebung. Wenn nicht angegeben, wird es automatisch von der Windows Server-Host-Version erkannt. Beispiel: `"64bit Windows Server 2016 v2.16.2 running IIS 10.0"` Eine Liste der verfügbaren Plattformversionen finden Sie unter web platform list.

Name	Beschreibung
`--remote`	Zeigt an, dass die Migration im Remote-Modus ausgeführt werden soll. Diese Option ermöglicht die Ausführung von einem Bastion-Host aus, der eine Verbindung zu dem Zielserver herstellt, der die Anwendung und die Konfigurationen enthält, die zu Elastic Beanstalk migriert werden sollen. Wird auf dem Bastion-Server ausgeführt, `eb migrate` erkennt Konfigurationen, stellt die Migrationslogik auf dem Bastion-Host bereit und stellt Ihre Anwendung anschließend in einer neuen Elastic Beanstalk Beanstalk-Umgebung bereit. Mit dieser Option müssen Sie EB CLI und Python nicht mehr auf dem Windows-Server installieren, den Sie migrieren müssen. Sie installieren Python und die EB-CLI stattdessen auf einem Bastion-Host, wo Sie den eb migrate Befehl mit der `--remote` Option ausführen. Verwenden Sie die `--target-ip` Option, um den Host mit den zu migrierenden IIS-Konfigurationen anzugeben. Muss mit `--target-ip --username` , und verwendet werden`--password` .
`--target-ip` *`ip-address`*	Öffentliche IP-Adresse des Windows-Remotecomputers, auf dem sich die zu migrierenden IIS-Server befinden. Erforderlich bei der Verwendung von`--remote`. Kann nur bei Verwendung angegeben werden`--remote`.
`--username` *`username`*	Benutzername des Benutzerprofils für den Zugriff auf den Windows-Remotecomputer, der die zu migrierenden IIS-Server enthält. Erforderlich bei der Verwendung`--remote`. Kann nur bei Verwendung angegeben werden`--remote`.

Name	Beschreibung
`--password` *password*	Passwort des Benutzerprofils für den Zugriff auf den Windows-Remotecomputer, der die zu migrierenden IIS-Server enthält. Erforderlich bei der Verwendung`--remote`. Kann nur bei Verwendung angegeben werden`--remote`.
`-sr` *role-name* oder `--service-role` *role-name*	IAM-Servicerolle für Elastic Beanstalk zur Verwaltung verwandter Dienste. AWS Wenn nicht angegeben, wird eine Standard-Servicerolle mit den erforderlichen Berechtigungen erstellt. Weitere Informationen finden Sie unter the section called “Service – -Rolle”. **Note** Geben Sie nur den Rollennamen an, nicht den vollständigen ARN. Elastic Beanstalk erstellt automatisch den kompletten ARN.
`-s` *site-names* oder `--sites` *site-names*	Durch Kommas getrennte Liste der zu migrierenden IIS-Sites. Wenn nicht angegeben, werden alle verfügbaren Sites auf dem Server migriert. Beispiel: `--sites "Default Web Site,Intranet,API"`

Name	Beschreibung
`--ssl-certificates` *certificate-arn* [,*certificate-arn* ...]	Durch Kommas getrennte Liste der ACM-SSL-Zertifikate, die dem Application Load Balancer zugeordnet ARNs werden sollen. Erforderlich für die Migration von Websites mit HTTPS-Bindungen. Beispiel: `--ssl-certificates arn:aws:acm:region:account:certificate/certificate-id`
`-t` *key1*=*value1*[,*key2*=*value2* ...] oder `--tags` *key1*=*value1*[,*key2*=*va*	Kommagetrennte Liste von `key=value` Paaren zur Kennzeichnung neuer Ressourcen in Ihrer Umgebung: Umgebung, Elastic Beanstalk Beanstalk-Anwendung, Anwendungsversion. Weitere Informationen finden Sie unter Tagging-Umgebungen.
`--verbose`	Zeigt detaillierte Informationen während des Migrationsprozesses an. Zeigt bei Verwendung mit einem explore Unterbefehl umfassende Informationen zur Site-Konfiguration an.

Name	Beschreibung

`-vpc` *config-file-or-string*
 	
oder
 	
`--vpc-config` *config-file-or-string*
	VPC-Konfiguration für die Umgebung, entweder als JSON-Dateipfad oder als JSON-Zeichenfolge angegeben.
	
	Die Konfiguration muss Folgendes beinhalten:
	
	`{`
	`"id": "vpc-1234567890abcdef0",`
	`"publicip": "true|false",`
	`"elbscheme": "public|private",`
	`"ec2subnets": ["subnet-a1b2c3d4",`
	`"subnet-e5f6g7h8"],`
	`"securitygroups": "sg-123456,sg-789012",`
	`"elbsubnets": ["subnet-a1b2c3d4",`
	`"subnet-e5f6g7h8"]`
	`}`
	
	• `id`: (Erforderlich) VPC-ID
	• `publicip`: Ob Instanzen öffentlich zugewiesen IPs werden sollen
	• `elbscheme` : Load Balancer-Schema (öffentlich oder privat)
	• `ec2subnets` : Liste der Subnetze IDs für Instanzen EC2
	• `securitygroups` : Kommagetrennte Sicherhei tsgruppe IDs
	• `elbsubnets` : Liste der Subnetze IDs für den Load Balancer
	
	Important
	Bei der Migration werden alle vorhandenen VPC-Einstellungen aus der Quellumgebung ignoriert, wenn Sie den *`--vpc-config`* Parameter angeben. Wenn Sie diesen Parameter verwenden, verwendet die Migration nur die

Name	Beschreibung
	VPC-Einstellungen, die in der Konfigurationsdatei angegeben sind, die Sie übergeben. Die Verwendung dieses Parameters setzt das Standardverhalten beim Erkennen der VPC-Konfiguration der Quell-Instance oder beim Verwenden der Standard-VPC außer Kraft.
Allgemeine Optionen	

Ausgabe

Der Befehl stellt während des gesamten Migrationsprozesses Statusaktualisierungen bereit:

1. VPC-Konfigurationserkennung (bei Ausführung auf einer EC2 Instance)
2. Fortschritt der Generierung des Quellpakets für jede Site
3. Status der Erstellung der Umgebung
4. Fortschritt bei der Bereitstellung

Bei erfolgreicher Ausführung werden die Details der neuen Umgebung angezeigt, darunter:

- Name und ID der Umgebung
- Anwendungsname
- Region
- Version der Plattform
- Umgebung CNAME

Verwenden Sie bei Problemen während der Migration die eb healthBefehle eb eventsund, um detaillierte Informationen zu erhalten.

Beispiele

Grundlegende Verwendung

Grundlegende Migration im interaktiven Modus:

```
PS C:\migrations_workspace >  eb migrate
Identifying VPC configuration of this EC2 instance (i-0123456789abcdef0):
  id: vpc-1234567890abcdef0
  publicip: true
  elbscheme: public
  ec2subnets: subnet-123,subnet-456,subnet-789
  securitygroups: sg-123,sg-456
  elbsubnets: subnet-123,subnet-456,subnet-789

Using .\migrations\latest to contain artifacts for this migration run.
Generating source bundle for sites, applications, and virtual directories...
  Default Web Site/ -> .\migrations\latest\upload_target\DefaultWebSite.zip

Creating application version
Creating environment

Environment details for: EBMigratedEnv
  Application name: EBMigratedApp
  Region: us-west-2
  Deployed Version: app-230320_153045
  Environment ID: e-abcdef1234
  Platform: 64bit Windows Server 2019 v2.7.0 running IIS 10.0
  Tier: WebServer-Standard-1.0
  CNAME: ebmigratedenv.us-west-2.elasticbeanstalk.com
  Updated: 2023-03-20 15:30:45
```

Migration bestimmter Websites mit benutzerdefinierter Konfiguration:

```
PS C:\migrations_workspace >  eb migrate `
    --sites "Default Web Site,InternalAPI" `
    --application-name "CorporateApp" `
    --environment-name "Production" `
    --instance-type "c5.xlarge" `
    --tags "Environment=Production,Team=WebOps" `
    --copy-firewall-config
```

Migrationsarchiv ohne Bereitstellung erstellen:

```
PS C:\migrations_workspace >  eb migrate --archive-only
Using .\migrations\latest to contain artifacts for this migration run.
Generating source bundle for sites, applications, and virtual directories...
  Default Web Site/ -> .\migrations\latest\upload_target\DefaultWebSite.zip

Generated destination archive directory at .\migrations\latest\upload_target
You can execute `eb init` and `eb create` from this directory to deploy to EB.
```

Beispiele für erweiterte Konfigurationen

Migration mit benutzerdefinierter VPC-Konfiguration mithilfe einer JSON-Datei:

```
PS C:\migrations_workspace >  cat vpc-config.json
{
    "id": "vpc-1234567890abcdef0",
    "publicip": "false",
    "elbscheme": "internal",
    "ec2subnets": [
        "subnet-private1",
        "subnet-private2"
    ],
    "securitygroups": [
        "sg-app",
        "sg-database",
        "sg-monitoring"
    ],
    "elbsubnets": [
        "subnet-private1",
        "subnet-private2"
    ]
}

PS C:\migrations_workspace eb migrate `
    --sites "InternalAPI" `
    --vpc-config vpc-config.json `
    --instance-type "r5.xlarge" `
    --tags "Environment=Internal,Security=High"
```

Migration von Websites mit SSL-Zertifikaten und Host-Headern:

```
PS C:\migrations_workspace >  eb migrate `
```

```
    --sites "SecurePortal" `
    --ssl-certificates "arn:aws:acm:region:account:certificate/
cert1,arn:aws:acm:region:account:certificate/cert2" `
    --verbose
INFO: Detected HTTPS bindings:
  - www.example.com:443
  - api.example.com:443

INFO: Configuring Application Load Balancer with SSL certificates
INFO: Creating host-based routing rules:
  - www.example.com -> target group 1
  - api.example.com -> target group 2
```

Migration mit EBS-Snapshot-Konfiguration:fo

```
PS C:\migrations_workspace >  eb migrate `
    --sites "Default Web Site" `
    --ebs-snapshots "snap-1234567890abcdef0" "snap-0987654321fedcba1" `
    --encrypt-ebs-volumes
Using .\migrations\latest to contain artifacts for this migration run.
INFO: Enabling EBS encryption for all new volumes in us-west-2
INFO: Configuring environment with specified EBS snapshots
```

Beispiele für Sicherheitskonfigurationen

Umgang mit Websites mit komplexen Firewallregeln:

```
PS C:\migrations_workspace >  eb migrate `
    --sites "Default Web Site,ReportingService" `
    --copy-firewall-config `
    --verbose
INFO: Detected the following Windows Firewall rules:
  - Allow Web Traffic (TCP 80, 443)
  - Allow Reporting Traffic (TCP 8081)
INFO: Creating corresponding security group rules
```

Migration mit benutzerdefinierten IAM-Rollen:

```
PS C:\migrations_workspace >  eb migrate `
    --sites "SecureApp" `
    --instance-profile "CustomInstanceProfile" `
    --service-role "CustomServiceRole"
```

Beispiele für die Ausführung per Fernzugriff

Migration von IIS-Anwendungen von einem Windows-Remoteserver:

```
PS C:\migrations_workspace >  eb migrate `
    --remote `
    --target-ip "192.0.2.10" `
    --username "administrator" `
    --password "YourPassword123" `
    --application-name "RemoteApp" `
    --environment-name "RemoteEnv"
INFO: Establishing SSH connection to remote host 192.0.2.10...
INFO: Connection established
INFO: Discovering IIS sites on remote host...
INFO: Found 2 sites: Default Web Site, API
INFO: Extracting site configurations...
INFO: Generating source bundle for sites, applications, and virtual directories...
  Default Web Site/ -> .\migrations\latest\upload_target\DefaultWebSite.zip
  API/ -> .\migrations\latest\upload_target\API.zip

Creating application version
Creating environment

Environment details for: RemoteEnv
  Application name: RemoteAppstage mi
  Region: us-west-2
  Deployed Version: app-230320_153045
  Environment ID: e-abcdef1234
  Platform: 64bit Windows Server 2019 v2.7.0 running IIS 10.0
  Tier: WebServer-Standard-1.0
  CNAME: remoteenv.us-west-2.elasticbeanstalk.com
  Updated: 2023-03-20 15:30:45
```

Remote-Migration mit spezifischer Standortwahl:

```
PS C:\migrations_workspace >  eb migrate `
    --remote `
    --target-ip "192.0.2.10" `
    --username "administrator" `
    --password "YourPassword123" `
    --sites "API" `
    --instance-type "c5.large"
```

eb open

Description

Öffnet die öffentliche URL der Website im Standardbrowser.

Syntax

eb open

eb open ***environment-name***

Optionen

Name	Beschreibung
Allgemeine Optionen	

Ausgabe

Der Befehl eb open erzeugt keine Ausgabe. Stattdessen wird die Anwendung in einem Browserfenster geöffnet.

eb platform

Description

Dieser Befehl unterstützt zwei unterschiedliche Workspaces:

Plattform

Verwenden Sie diesen Workspace zum Verwalten von benutzerdefinierten Plattformen.

Umgebung

Verwenden Sie diesen Workspace, um eine Standardplattform auszuwählen oder Informationen zur aktuellen Plattform anzuzeigen.

Elastic Beanstalk bietet den Schnellzugang ebp für eb platform.

> **Note**
>
> Windows PowerShell verwendet ebp als Befehlsalias. Wenn Sie die EB-CLI in Windows ausführen PowerShell, verwenden Sie die Langform dieses Befehls —eb platform.

Verwenden der eb-Plattform für benutzerdefinierte Plattformen

Listet die Versionen der aktuellen Plattform auf und ermöglicht Ihnen die Verwaltung von benutzerdefinierten Plattformen.

Syntax

eb platform create [***version***] [***options***]

eb platform delete [***version***] [***options***]

eb platform events [***version***] [***options***]

eb platform init [***platform***] [***options***]

eb platform list [***options***]

eb platform logs [***version***] [***options***]

eb platform status [***version***] [***options***]

eb platform use [***platform***] [***options***]

Optionen

Name	Beschreibung
`create [version] [options]`	Erstellen Sie eine neue Version der Plattform. Weitere Informationen.
`delete version [options]`	Löschen Sie eine Plattformversion. Weitere Informationen.
`events [version] [options]`	Zeigen Sie die Ereignisse einer Plattformversion an. Weitere Informationen.

Name	Beschreibung
`init [platform] [options]`	Initialisieren Sie ein Plattform-Repository. Weitere Informationen.
`list [options]`	Listen Sie die Versionen der aktuellen Plattform auf. Weitere Informationen.
`logs [version] [options]`	Zeigen Sie Protokolle aus der Builder-Umgebung für eine Plattformversion an. Weitere Informationen.
`status [version] [options]`	Zeigen Sie den Status einer Plattformversion an. Weitere Informationen.
`use [platform] [options]`	Wählen Sie eine andere Plattform, aus der neue Versionen erstellt werden. Weitere Informationen.
Allgemeine Optionen	

Allgemeine Optionen

Alle eb platform-Befehle umfassen die folgenden allgemeinen Optionen.

Name	Beschreibung

`-h`
ODER
`--help`
	Zeigt eine Hilfenachricht an und wird beendet.

`--debug`	Zeigt zusätzliche Debugging-Ausgabe.
`--quiet`	Unterdrückt alle Ausgabedateien.

`-v`
ODER
`--verbose`
	Zeigt zusätzliche Ausgabedateien an.

Name	Beschreibung
`--profile` *PROFILE*	Verwendet die in Ihren Anmeldedaten angegebenen *PROFILE* Daten.
`-r` *REGION* ODER `--region` *REGION*	Verwenden Sie die Region*REGION*.
`--no-verify-ssl`	Überprüfen Sie keine AWS SSL-Zertifikate.

Eb platform create

Erstellt eine neue Version der Plattform und gibt den ARN für die neue Version zurück. Wenn keine Builder-Umgebung in der aktuellen Region ausgeführt wird, startet dieser Befehl eine. Die Optionen *version* und Inkrement (`-M`-`m`, und`-p`) schließen sich gegenseitig aus.

Optionen

Name	Description
version	Wenn *version* nicht angegeben, wird eine neue Version auf der Grundlage der neuesten Plattform erstellt, wobei die Patch-Version (N in N.n.N) inkrementiert wird.

`-M`
 	
ODER
 	
`--major-increment`
	Erhöht die höhere Versionsnummer (die N in N.n.n).

`-m`
 	
ODER
 	
`--minor-increment`
	Erhöht die kleinere Versionsnummer (die N in n.N.n).

`-p`	Erhöht die Patch-Versionsnummer (die N in n.n.N).

Name	Description
ODER `--patch-increment`	
`-i` *`INSTANCE_TYPE`* ODER --instance-type *`INSTANCE_TYPE`*	Verwenden Sie *`INSTANCE_TYPE`* als Instanztyp, z. B. **`t1.micro`**
`-ip` *`INSTANCE_PROFILE`* ODER `--instance-profile` *`INSTANCE_PROFILE`*	Verwenden Sie *`INSTANCE_PROFILE`* es als Instanzprofil, wenn Sie AMIs Inhalte für eine benutzerdefinierte Plattform erstellen. Wenn die `-ip`-Option nicht angegeben ist, wird das Instance-Profil `aws-elasticbeanstalk-custom-platforme-ec2-role` erstellt und für die benutzerdefinierte Plattform verwendet.
`--tags` *`key1`*`=`*`value1`*`[,`*`ke`	Tags werden als durch Komma getrennte Liste von `key=value` -Paaren angegeben.
`--timeout` *`minutes`*	Legen Sie Anzahl der Minuten fest, bevor ein Timeout für den Befehl auftritt.
`--vpc.id` *`VPC_ID`*	Die ID der VPC, in der Packer erstellt.
`--vpc.subnets` *`VPC_SUBNETS`*	Die VPC-Subnetze, in der Packer erstellt.
`--vpc.publicip`	Ordnet den gestarteten EC2 Instances IPs die Öffentlichkeit zu.

Eb platform delete

Löschen Sie eine Plattformversion. Die Version wird nicht gelöscht, wenn eine Umgebung diese Version verwendet.

Optionen

Name	Beschreibung
version	Die zu löschende Version. Dieser Wert ist erforderlich.
--cleanup	Entfernen Sie alle Plattformversionen mit dem Failed-Status.
--all-platforms	Wenn --cleanup angegeben ist, entfernen Sie alle Plattform versionen mit dem Failed-Status für alle Plattformen.
--force	Keine Bestätigung beim Löschen einer Version erforderlich.

Eb platform events

Zeigen Sie die Ereignisse einer Plattformversion an. Wenn *version* angegeben, werden die Ereignisse aus dieser Version angezeigt, andernfalls werden die Ereignisse aus der aktuellen Version angezeigt.

Optionen

Name	Description
version	Die Version, für die die Ereignisse angezeigt werden. Dieser Wert ist erforderlich.
-f ODER --follow	Ereignisse weiterhin anzeigen, wenn sie auftreten.

Eb platform init

Initialisieren Sie ein Plattform-Repository.

Optionen

Name	Beschreibung
platform	Der Name der zu initialisierenden Plattform. Dieser Wert ist erforderlich, es sei denn, `-i` (interaktiver Modus) ist aktiviert.

`-i`
ODER
`--interactive`
	Verwenden Sie den interaktiven Modus.

`-k` *KEYNAME* ODER `--keyname` *KEYNAME*	Der EC2 Standardschlüsselname.

Sie können diesen Befehl in einem Verzeichnis ausführen, das zuvor initialisiert wurde. Sie können den Workspace-Typ jedoch nicht mehr ändern, wenn er in einem Verzeichnis ausgeführt wird, das zuvor initialisiert wurde.

Um mit anderen Optionen neu zu initialisieren, verwenden Sie die `-i`-Option.

Eb platform list

Listen Sie die Versionen der Plattform auf, die mit dem Workspace (Verzeichnis) oder einer Region verknüpft sind.

Der Befehl gibt unterschiedliche Ergebnisse zurück, abhängig von der Workspace-Art, die Sie ausführen:

- In einem Plattform-Workspace (ein Verzeichnis, das durch `eb platform init` initialisiert wurde) gibt der Befehl eine Liste aller Plattformversionen der in diesem Workspace definierten benutzerspezifischen Plattformen zurück. Fügen Sie die Option `--all-platforms` oder `--verbose` hinzu, um eine Liste aller Plattformversionen aller benutzerdefinierten Plattformen anzuzeigen, mit denen Ihr Konto in der Region mit dem Workspace verknüpft ist.
- In einem Anwendungs-Workspace (ein Verzeichnis, dass durch `eb init` initialisiert wurde) gibt der Befehl eine Liste aller Plattformversionen aus. Diese enthält die durch Elastic Beanstalk

verwalteten Plattformen und die benutzerdefinierten Plattformen Ihres Kontos. Die Liste enthält die Kurznamen der Plattformversionen und einige Plattformversionsvarianten können zusammengefasst aufgelistet sein. Fügen Sie die Option `--verbose` hinzu, um eine detaillierte Liste mit den vollständigen Namen anzuzeigen, in der alle Versionsvarianten separat aufgelistet sind.

- In einem nicht initialisierten Verzeichnis funktioniert nur der Befehl mit der `--region`-Option. Sie gibt eine Liste aller von Elastic Beanstalk verwalteten Plattformversionen zurück, die in der Region unterstützt werden. Die Liste enthält die Kurznamen der Plattformversionen und einige Plattformversionsvarianten können zusammengefasst aufgelistet sein. Fügen Sie die Option `--verbose` hinzu, um eine detaillierte Liste mit den vollständigen Namen anzuzeigen, in der alle Versionsvarianten separat aufgelistet sind.

Optionen

Name	Beschreibung

`-a`
 	
ODER
 	
`--all-platforms`
	Gilt nur in einem initialisierten Workspace (einem Verzeichn is, dass durch `eb platform init` oder `eb init` initialisiert wurde). Listet die Versionen aller benutzerdefinierten Plattform en auf, die Ihrem Konto zugeordnet sind.

`-s` *`STATUS`* ODER `--status` *`STATUS`*	Nur die passenden Plattformen auflisten*`STATUS`*: • Bereit • Fehlgeschlagen • Deleting (Wird gelöscht) • Creating

Eb platform logs

Zeigen Sie Protokolle aus der Builder-Umgebung für eine Plattformversion an.

Optionen

Name	Beschreibung
version	Die Version der Plattform, für die Protokolle angezeigt werden. Wenn nicht angegeben, werden Protokolle aus der aktuellen Version angezeigt.
--stream	Stream-Bereitstellungsprotokolle, mit denen eingerichtet wurde CloudWatch.

Eb platform status

Zeigen Sie den Status einer Plattformversion an.

Optionen

Name	Beschreibung
version	Die Version der Plattform, für die der Status abgerufen wird. Wenn nicht angegeben, wird der Status der aktuellen Version angezeigt.

Eb platform use

Wählen Sie eine andere Plattform, aus der neue Versionen erstellt werden.

Optionen

Name	Description
platform	Gibt *platform* als aktive Version für diesen Workspace an. Dieser Wert ist erforderlich.

Verwenden der eb-Plattform für Umgebungen

Listet die unterstützten Plattformen auf und ermöglicht Ihnen, die Standardplattform und die zu verwendende Plattformversion festzulegen, wenn Sie eine Umgebung starten. Verwenden Sie eb platform list, um eine Liste aller unterstützten Plattformen anzuzeigen. Verwenden Sie eb platform select zum Ändern der Plattform für Ihr Projekt. Verwenden Sie eb platform show zum Anzeigen der ausgewählten Plattform Ihres Projekts.

Syntax

eb platform list

eb platform select

eb platform show

Optionen

Name	Beschreibung
`list`	Listen Sie die Version der aktuellen Plattform auf.
`select`	Wählen Sie die Standardplattform.
`show`	Zeigen Sie Informationen zur aktuellen Plattform.

Beispiel 1

Im folgenden Beispiel werden die Namen aller Konfigurationen für alle Plattformen aufgelistet, die Elastic Beanstalk unterstützt.

```
$ eb platform list
docker-1.5.0
glassfish-4.0-java-7-(preconfigured-docker)
glassfish-4.1-java-8-(preconfigured-docker)
go-1.3-(preconfigured-docker)
go-1.4-(preconfigured-docker)
iis-7.5
iis-8
iis-8.5
multi-container-docker-1.3.3-(generic)
```

```
node.js
php-5.3
php-5.4
php-5.5
python
python-2.7
python-3.4
python-3.4-(preconfigured-docker)
ruby-1.9.3
ruby-2.0-(passenger-standalone)
ruby-2.0-(puma)
ruby-2.1-(passenger-standalone)
ruby-2.1-(puma)
ruby-2.2-(passenger-standalone)
ruby-2.2-(puma)
tomcat-6
tomcat-7
tomcat-7-java-6
tomcat-7-java-7
tomcat-8-java-8
```

Beispiel 2

Im folgenden Beispiel werden Sie aufgefordert, aus einer Liste von Plattformen auszuwählen und die Version zu nehmen, die Sie für die angegebene Plattform bereitstellen möchten.

```
$ eb platform select
Select a platform.
1) PHP
2) Node.js
3) IIS
4) Tomcat
5) Python
6) Ruby
7) Docker
8) Multi-container Docker
9) GlassFish
10) Go
(default is 1): 5

Select a platform version.
1) Python 2.7
2) Python
```

```
3) Python 3.4 (Preconfigured - Docker)
```

Beispiel 3

Im folgenden Beispiel werden Informationen über die aktuelle Standardplattform gezeigt.

```
$ eb platform show
Current default platform: Python 2.7
New environments will be running:  64bit Amazon Linux 2014.09 v1.2.0 running Python 2.7

Platform info for environment "tmp-dev":
Current: 64bit Amazon Linux 2014.09 v1.2.0 running Python
Latest:  64bit Amazon Linux 2014.09 v1.2.0 running Python
```

eb printenv

Description

Druckt alle Umgebungseigenschaften im Befehlsfenster.

Syntax

eb printenv

eb printenv ***environment-name***

Optionen

Name	Beschreibung
Allgemeine Optionen	

Ausgabe

Bei Erfolg gibt der Befehl den Status des `printenv`-Vorgangs aus.

Beispiel

Das folgende Beispiel druckt Umgebungseigenschaften für die angegebene Umgebung.

```
$ eb printenv
Environment Variables:
     PARAM1 = Value1
```

eb restore

Description

Erstellt eine beendete Umgebung erneut, erstellt eine neue Umgebung mit demselben Namen, derselben ID und Konfiguration. Die Umgebungsname, Domänennamen und die Anwendungsversion müssen verwendet werden können, damit die Neuerstellung erfolgreich abgeschlossen werden kann.

Syntax

eb restore

eb restore ***environment_id***

Optionen

Name	Beschreibung
Allgemeine Optionen	

Ausgabe

Die EB-CLI zeigt eine Liste der beendeten Umgebungen, die für Wiederherstellungen verfügbar sind.

Beispiel

```
$ eb restore
Select a terminated environment to restore

  #   Name          ID             Application Version      Date Terminated        Ago

  3   gamma         e-s7mimej8e9   app-77e3-161213_211138   2016/12/14 20:32 PST   13
 mins
```

```
  2   beta          e-sj28uu2wia   app-77e3-161213_211125   2016/12/14 20:32 PST   13
 mins
  1   alpha         e-gia8mphu6q   app-77e3-161213_211109   2016/12/14 16:21 PST   4
 hours

 (Commands: Quit, Restore, # #)

Selected environment alpha
Application:    scorekeep
Description:    Environment created from the EB CLI using "eb create"
CNAME:          alpha.h23tbtbm92.us-east-2.elasticbeanstalk.com
Version:        app-77e3-161213_211109
Platform:       64bit Amazon Linux 2016.03 v2.1.6 running Java 8
Terminated:     2016/12/14 16:21 PST
Restore this environment? [y/n]: y

2018-07-11 21:04:20    INFO: restoreEnvironment is starting.
2018-07-11 21:04:39    INFO: Created security group named: sg-e2443f72
...
```

eb scale

Description

Skaliert die Umgebung so, dass sie immer auf einer bestimmten Anzahl von Instances ausgeführt wird, wobei die minimale und maximale Anzahl an Instances auf die angegebene Zahl festgelegt wird.

Syntax

eb scale ***number-of-instances***

eb scale ***number-of-instances environment-name***

Optionen

Name	Beschreibung
--timeout	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.
Allgemeine Optionen	

Ausgabe

Wenn der Befehl erfolgreich war, wird die Anzahl der minimal und maximal auszuführenden Instances auf die angegebene Zahl aktualisiert.

Beispiel

Im folgenden Beispiel wird die Anzahl der Instances auf 2 festgelegt.

```
$ eb scale 2
2018-07-11 21:05:22    INFO: Environment update is starting.
2018-07-11 21:05:27    INFO: Updating environment tmp-dev's configuration settings.
2018-07-11 21:08:53    INFO: Added EC2 instance 'i-5fce3d53' to Auto Scaling Group
 'awseb-e-2cpfjbra9a-stack-AWSEBAutoScalingGroup-7AXY7U13ZQ6E'.
2018-07-11 21:08:58    INFO: Successfully deployed new configuration to environment.
2018-07-11 21:08:59    INFO: Environment update completed successfully.
```

eb setenv

Description

Damit werden Umgebungseigenschaften für die Standardumgebung festgelegt.

Syntax

eb setenv ***key*=*value***

Sie können beliebig viele Eigenschaften einbeziehen, die Gesamtgröße aller Eigenschaften darf jedoch nicht mehr als 4 096 Bytes betragen. Sie können eine Variable löschen, indem Sie den Wert leer lassen. Informationen zu den Beschränkungen finden Sie unter Konfiguration von Umgebungseigenschaften (Umgebungsvariablen).

> **Note**
>
> Falls `value` ein Sonderzeichen enthält, müssen Sie ein Escape-Zeichen in Form von \ voranstellen.

Optionen

Name	Beschreibung
--timeout	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.
Allgemeine Optionen	

Ausgabe

Bei Erfolg wird vom Befehl angezeigt, dass die Umgebungsaktualisierung erfolgreich ausgeführt werden konnte.

Beispiel

Im folgenden Beispiel wird die Umgebungsvariable ExampleVar festgelegt.

```
$ eb setenv ExampleVar=ExampleValue
2018-07-11 21:05:25    INFO: Environment update is starting.
2018-07-11 21:05:29    INFO: Updating environment tmp-dev's configuration settings.
2018-07-11 21:06:50    INFO: Successfully deployed new configuration to environment.
2018-07-11 21:06:51    INFO: Environment update completed successfully.
```

Mit dem folgenden Befehl werden mehrere Umgebungseigenschaften definiert. Er fügt die Umgebungseigenschaft `foo` hinzu, legt deren Wert auf `bar` fest, ändert den Wert der Eigenschaft `JDBC_CONNECTION_STRING` und löscht die Eigenschaften `PARAM4` und `PARAM5`.

```
$ eb setenv foo=bar JDBC_CONNECTION_STRING=hello PARAM4= PARAM5=
```

eb ssh

Description

> **Note**
>
> Dieser Befehl kann nicht auf Umgebungen mit Windows-Server-Instances ausgeführt werden.

Stellen Sie mithilfe von Secure Shell (SSH) eine Connect zu einer EC2 Linux-Amazon-Instance in Ihrer Umgebung her. Wenn eine Umgebung mehrere Instances ausführt, werden Sie von der EB-CLI aufgefordert anzugeben, mit welcher Instance Sie eine Verbindung herstellen möchten. Um diesen Befehl verwenden zu können, muss SSH auf Ihrem lokalen Computer installiert und über die Befehlszeile verfügbar sein. Private Schlüsseldateien müssen sich in einem Ordner befinden, der `.ssh` unter Ihrem Benutzerverzeichnis benannt ist, und die EC2 Instances in Ihrer Umgebung müssen öffentliche IP-Adressen haben.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, stellt dieser Befehl auch eine Verbindung zu Instances in der benutzerdefinierten Umgebung her.

> **SSH-Schlüssel**
>
> Wenn Sie SSH nicht zuvor konfiguriert haben, können Sie mit der EB-CLI einen Schlüssel erstellen, wenn ausgeführt wird eb init. Wenn Sie den Befehl eb init bereits ausgeführt haben, führen Sie ihn erneut mit der `--interactive`-Option aus und wählen Sie Yes (Ja) und Create New Keypair (Neues Schlüsselpaar anlegen) aus, wenn Sie zum Einrichten von SSH aufgefordert werden. Schlüssel, die während dieses Prozesses erstellt wurden, werden von der EB-CLI im richtigen Ordner gespeichert.

Mit diesem Befehl wird Port 22 vorübergehend in der Sicherheitsgruppe Ihrer Umgebung für eingehenden Datenverkehr von 0.0.0.0/0 (alle IP-Adressen) geöffnet, wenn keine Regeln für Port 22 vorhanden sind. Wenn Sie die Sicherheitsgruppe Ihrer Umgebung für mehr Sicherheit so konfiguriert haben, dass Port 22 für einen eingeschränkten CIDR-Bereich geöffnet wird, berücksichtigt die EB-CLI diese Einstellung und verzichtet auf Änderungen an der Sicherheitsgruppe. Wenn Sie dieses Verhalten überschreiben und die EB-CLI zwingen möchten, Port 22 für allen eingehenden Datenverkehr zu öffnen, verwenden Sie die `--force`-Option.

Weitere Informationen zum Konfigurieren der Sicherheitsgruppe Ihrer Umgebung finden Sie unter EC2 Sicherheitsgruppen.

Syntax

eb ssh

eb ssh ***environment-name***

Optionen

Name	Beschreibung

`-i`
 	
oder
 	
`--instance`
	Gibt die Instance-ID der Instance an, zu der Sie eine Verbindung herstellen. Wir empfehlen die Verwendung dieser Option.

`-n`
 	
oder
 	
`--number`
	Geben Sie durch eine Zahl die Instance an, zu der eine Verbindung hergestellt werden soll.

`-o`
 	
oder
 	
`--keep_open`
	Lassen Sie Port 22 auf der Sicherheitsgruppe offen, nachdem die SSH-Sitzung beendet wurde.

`--command`	Führen Sie einen Shell-Befehl auf der angegebenen Instance aus, anstatt eine SSH-Sitzung zu starten.
`--custom`	Geben Sie einen SSH-Befehl an, der anstatt "ssh -i keyfile" verwendet werden soll. Verwenden Sie den Remote-Benutzer- und Hostnamen nicht.
`--setup`	Ändern Sie das Schlüsselpaar, das den Instances der Umgebung zugewiesen ist (erfordert das Ersetzen von Instances).
`--force`	Öffnen Sie Port 22 für eingehenden Datenverkehr von 0.0.0.0/0 in der Sicherheitsgruppe der Umgebung, selbst wenn die Sicherheitsgruppe bereits für SSH konfiguriert ist. Verwenden Sie diese Option, wenn die Sicherheitsgruppe Ihrer Umgebung so konfiguriert ist, dass Port 22 für einen eingeschr

Name	Beschreibung
	änkten CIDR-Bereich geöffnet wird, der die IP-Adresse, von der Sie die Verbindung herstellen möchten, nicht enthält.
`--timeout` *minutes*	Legen Sie Anzahl der Minuten fest, bevor ein Timeout für den Befehl auftritt. Kann nur mit dem Argument `--setup` verwendet werden.
Allgemeine Optionen	

Ausgabe

Wenn der Befehl erfolgreich war, wird eine SSH-Verbindung mit der Instance geöffnet.

Beispiel

Im folgenden Beispiel werden Sie mit der angegebenen Umgebung verbunden.

```
$ eb ssh
Select an instance to ssh into
1) i-96133799
2) i-5931e053
(default is 1): 1
INFO: Attempting to open port 22.
INFO: SSH port 22 open.
The authenticity of host '54.191.45.125 (54.191.45.125)' can't be established.
RSA key fingerprint is ee:69:62:df:90:f7:63:af:52:7c:80:60:1b:3b:51:a9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.191.45.125' (RSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2014.09-release-notes/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-8-185 ~]$ ls
[ec2-user@ip-172-31-8-185 ~]$ exit
```

```
logout
Connection to 54.191.45.125 closed.
INFO: Closed port 22 on ec2 instance security group
```

eb status

Description

Bietet Informationen über den Status der Umgebung.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, bietet dieser Befehl auch Informationen über die Builder-Umgebung.

Syntax

eb status

eb status ***environment-name***

Optionen

Name	Beschreibung

`-v`
oder
`--verbose`
	Hier finden Sie weitere Informationen zu den einzelnen Instances, wie zum Beispiel ihren Status mit dem Elastic-Load-Balancing-Lastenverteilung.

Allgemeine Optionen	

Ausgabe

Bei Erfolg gibt der Befehl folgende Informationen zur Umgebung zurück:

- Environment name
- Anwendungsname
- Bereitgestellte Anwendungsversion
- Umgebungs-ID

- Plattform
- Umgebungsschicht
- CNAME
- Zeitpunkt der letzten Aktualisierung der Umgebung
- Status
- Integrität

Wenn Sie den ausführlichen Modus verwenden, stellt Ihnen EB CLI auch die Anzahl der laufenden EC2 Amazon-Instances zur Verfügung.

Beispiel

Im folgenden Beispiel wird der Status der Umgebung "tmp-dev" angezeigt.

```
$ eb status
Environment details for: tmp-dev
  Application name: tmp
  Region: us-west-2
  Deployed Version: None
  Environment ID: e-2cpfjbra9a
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev.elasticbeanstalk.com
  Updated: 2014-10-29 21:37:19.050000+00:00
  Status: Launching
  Health: Grey
```

eb swap

Description

Damit wird der CNAME einer Umgebung mit dem CNAME einer anderen Umgebung ausgetauscht (z. B. um Ausfallzeiten beim Aktualisieren der Anwendungsversion zu verhindern).

> **Note**
>
> Bei mehr als zwei Umgebungen werden Sie aufgefordert, aus einer Liste den Namen der Umgebung auszuwählen, die derzeit den gewünschten CNAME hat. Um dies zu unterbinden,

geben Sie den Namen der entsprechenden Umgebung an. Dazu binden Sie die Option `-n` ein, wenn Sie den Befehl ausführen.

Syntax

eb swap

eb swap ***environment-name***

> **Note**
>
> Das *environment-name* ist die Umgebung, für die Sie einen anderen CNAME wünschen. Wenn Sie bei der Ausführung keinen Befehlszeilenparameter angeben *environment-name*eb swap, aktualisiert EB CLI den CNAME der Standardumgebung.

Optionen

Name	Beschreibung

`-n`
oder
`--destination_name`
	Gibt den Namen der Umgebung an, mit der Sie wechseln CNAMEs möchten. Falls Sie eb swap ohne diese Option ausführen, werden Sie von der EB-CLI aufgefordert, aus einer Liste eine Umgebung auszuwählen.

Allgemeine Optionen	

Ausgabe

Bei Erfolg gibt der Befehl den Status des `swap`-Vorgangs aus.

Beispiele

Im folgenden Beispiel wird die Umgebung "tmp-dev" mit "live-env" ausgetauscht.

```
$ eb swap
```

```
Select an environment to swap with.
1) staging-dev
2) live-env
(default is 1): 2
2018-07-11 21:05:25    INFO: swapEnvironmentCNAMEs is starting.
2018-07-11 21:05:26    INFO: Swapping CNAMEs for environments 'tmp-dev' and 'live-env'.
2018-07-11 21:05:30    INFO: 'tmp-dev.elasticbeanstalk.com' now points to 'awseb-e-j-
AWSEBLoa-M7U21VXNLWHN-487871449.us-west-2.elb.amazonaws.com'.
2018-07-11 21:05:30    INFO: Completed swapping CNAMEs for environments 'tmp-dev' and
 'live-env'.
```

Im folgenden Beispiel wird die Umgebung "tmp-dev" mit der Umgebung "live-env" ausgetauscht, aber Sie werden nicht aufgefordert, einen Wert für eine Einstellung einzugeben oder auszuwählen.

```
$ eb swap tmp-dev --destination_name live-env
2018-07-11 21:18:12    INFO: swapEnvironmentCNAMEs is starting.
2018-07-11 21:18:13    INFO: Swapping CNAMEs for environments 'tmp-dev' and 'live-env'.
2018-07-11 21:18:17    INFO: 'tmp-dev.elasticbeanstalk.com' now points to 'awseb-e-j-
AWSEBLoa-M7U21VXNLWHN-487871449.us-west-2.elb.amazonaws.com'.
2018-07-11 21:18:17    INFO: Completed swapping CNAMEs for environments 'tmp-dev' and
 'live-env'.
```

eb tags

Description

Sie können Tags einer Elastic-Beanstalk-Ressource hinzufügen, löschen, aktualisieren und auflisten.

Weitere Details zur Ressourcen-Markierung mit Tags in Elastic Beanstalk finden Sie unter Markieren von Elastic Beanstalk-Anwendungsressourcen.

Syntax

eb tags [***environment-name***] [--resource ***ARN***] -l | --list

eb tags [***environment-name***] [--resource ***ARN***] -a | --add ***key1***=***value1***[,***key2***=***value2*** ...]

eb tags [***environment-name***] [--resource ***ARN***] -u | --update ***key1***=***value1***[,***key2***=***value2*** ...]

eb tags [***environment-name***] [--resource ***ARN***] -d | --delete ***key1***[,***key2*** ...]

Sie können die Unterbefehlsoptionen `--add`, `--update` und `--delete` in einem einzelnen Befehl kombinieren. Es ist mindestens eine der Optionen erforderlich. Diese drei Unterbefehlsoptionen können nicht mit `--list` kombiniert werden.

Ohne zusätzliche Argumente werden mit diesen Befehlen Tags der Standardumgebung in der Anwendung des aktuellen Verzeichnisses aufgelistet oder geändert. Mit einem *`environment-name`* Argument listen die Befehle die Tags dieser Umgebung auf oder ändern sie. Mit der `--resource`-Option führen die Befehle Tags einer beliebigen Elastic-Beanstalk-Ressource – eine Anwendung, eine Umgebung, eine Anwendungsversion, eine gespeicherte Konfiguration oder eine benutzerdefinierte Plattformversion – auf oder ändern sie. Geben Sie die -Ressource anhand ihres Amazon-Amazon-Ressourcenname (ARN) an.

Optionen

Keine dieser Optionen ist erforderlich. Wenn Sie eb create ohne Optionen ausführen, werden Sie aufgefordert, einen Wert für jede Einstellung einzugeben oder auszuwählen.

Name	Beschreibung

`-l`
oder
`--list`
	Listen Sie alle Tags auf, die derzeit auf die Umgebung angewendet werden.

`-a` *`key1=value1[,key2=value2`* oder `--add` *`key1=value1[,key2=val`*	Wenden Sie neue Tags auf die Ressource an. Legen Sie Tags als durch Komma getrennte Liste von `key=value` -Paaren fest. Sie können keine Schlüssel für vorhandene Tags festlegen. Folgende Werte sind zulässig: Siehe Taggen von Ressourcen.
`-u` *`key1=value1[,key2=value2`* oder `--updat e` *`key1=value1[,key2=value2`* .	Aktualisieren Sie die Werte von vorhandenen Ressource n-Tags. Legen Sie Tags als durch Komma getrennte Liste von `key=value` -Paaren fest. Sie müssen Schlüssel für vorhandene Tags festlegen.

Name	Beschreibung
	Folgende Werte sind zulässig: Siehe Taggen von Ressourcen.

`-d key1[,key2 ...]`
 	
oder
 	
`--delete key1[,key2 ...]`
	Löschen Sie vorhandene Ressourcen-Tags. Legen Sie Tags als durch Komma getrennte Liste von Schlüssel n fest. Sie müssen Schlüssel für vorhandene Tags festlegen.
	
	Folgende Werte sind zulässig: Siehe Taggen von Ressourcen.

`-r region`
 	
oder
 	
`--region region`
	Die, AWS-Region in der Ihre Ressource existiert.
	
	Standard: die konfigurierte Standardregion
	
	Eine Liste der Werte, die Sie für diese Option angeben können, finden Sie unter AWS Elastic Beanstalk - Endpunkte und -Kontingente im Allgemeine AWS-Refer enz.

`--resource ARN`	Der ARN der Ressource, die vom Befehl geändert wird oder für die Tags aufgelistet werden. Wenn Sie diesen Parameter nicht angeben, bezieht sich der Befehl auf die (Standard- oder angegebene) Umgebung im aktuellen Verzeichnis. Gültige Werte: Weitere Informationen finden Sie in einem der Unterthemen von Taggen von Ressourcen, das sich auf die entsprechende Ressource bezieht. In diesen Themen wird gezeigt, wie der ARN der Ressource aufgebaut ist, und es wird erklärt, wie Sie eine Liste der Ressourcen abrufen können ARNs , die für Ihre Anwendung oder Ihr Konto vorhanden sind.

Ausgabe

Mit der `--list`-Unterbefehlsoption können Sie eine Liste der Tags der Ressource anzeigen. Die Ausgabe enthält sowohl die Tags, die von Elastic Beanstalk standardmäßig angewendet werden, als auch benutzerdefinierte Tags.

```
$ eb tags --list
Showing tags for environment 'MyApp-env':

Key                                 Value

Name                                MyApp-env
elasticbeanstalk:environment-id     e-63cmxwjaut
elasticbeanstalk:environment-name   MyApp-env
mytag                               tagvalue
tag2                                2nd value
```

Für die Unterbefehlsoptionen `--add`, `--update` und `--delete` gibt es bei erfolgreicher Ausführung keine Ausgabe. Fügen Sie die Option `--verbose` hinzu, um eine detaillierte Ausgabe der Befehlsaktivitäten anzuzeigen.

```
$ eb tags --verbose --update "mytag=tag value"
Updated Tags:

Key                                 Value

mytag                               tag value
```

Beispiele

Der folgende Befehl fügt ein Tag mit dem Schlüssel `tag1` und dem Wert `value1` der Standardumgebung der Anwendung hinzu und löscht gleichzeitig das Tag `tag2`.

```
$ eb tags --add tag1=value1 --delete tag2
```

Mit dem folgenden Befehl wird einer gespeicherten Konfiguration innerhalb einer Anwendung ein Tag hinzugefügt.

```
$ eb tags --add tag1=value1 \
```

```
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:configurationtemplate/my-app/my-template"
```

Der folgende Befehl schlägt fehl, da versucht wird, ein nicht vorhandenes Tag zu aktualisieren.

```
$ eb tags --update tag3=newval
ERROR: Tags with the following keys can't be updated because they don't exist:

  tag3
```

Der folgende Befehl schlägt fehl, da versucht wird, denselben Schlüssel zu aktualisieren und zu löschen.

```
$ eb tags --update mytag=newval --delete mytag
ERROR: A tag with the key 'mytag' is specified for both '--delete' and '--update'. Each
 tag can be either deleted or updated in a single operation.
```

eb terminate

Description

Beendet die laufende Umgebung, sodass Ihnen keine Gebühren für ungenutzte AWS Ressourcen entstehen.

Die Verwendung der `--all`-Option löscht die Anwendung, für die das aktuelle Verzeichnis mit eb init initialisiert wurde. Der Befehl beendet alle Umgebungen in der Anwendung. Es beendet auch die Anwendungsversionen und gespeicherten Konfigurationen für die Anwendung und löscht dann die Anwendung.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, beendet dieser Befehl die ausgeführte benutzerdefinierte Umgebung.

> **Note**
>
> Sie können zu einem späteren Zeitpunkt problemlos eine neue Umgebung mit derselben Version starten.

Wenn Sie über Daten aus einer Umgebung verfügen, die Sie beibehalten möchten, setzen Sie die Richtlinie zum Löschen von Datenbanken auf `Retain`, bevor Sie die Umgebung beenden. Dies hält

die Datenbank außerhalb von Elastic Beanstalk betriebsbereit. Danach müssen sich alle Elastic-Beanstalk-Umgebungen als externe Datenbank damit verbinden. Wenn Sie die Daten sichern möchten, ohne die Datenbank betriebsbereit zu halten, legen Sie die Löschrichtlinie so fest, dass eine Snapshot der Datenbank erstellt wird, bevor die Umgebung beendet wird. Weitere Informationen finden Sie unter Datenbank-Lebenszyklus im Kapitel Konfigurieren von Umgebungen in diesem Leitfaden.

> **Important**
>
> Wenn Sie eine Umgebung beenden, müssen Sie auch alle CNAME-Zuordnungen löschen, die Sie erstellt haben, damit andere Kunden einen verfügbaren Hostnamen wiederverwenden können. Achten Sie darauf, DNS-Einträge zu löschen, die auf Ihre beendete Umgebung verweisen, um zu verhindern, dass ein DNS-Eintrag hängenbleibt. Ein hängender DNS-Eintrag kann den Internetdatenverkehr, der für Ihre Domain bestimmt ist, Sicherheitslücken aussetzen. Er kann auch andere Risiken bergen.
> Weitere Informationen finden Sie unter Schutz vor hängenden Delegierungsdatensätzen in Route 53 im Entwicklerhandbuch von Amazon Route 53. Weitere Informationen zu fehlernden DNS-Einträgen finden Sie auch im AWS Sicherheits-Blog unter Enhanced Domain Protections for Amazon CloudFront Requests.

Syntax

eb terminate

eb terminate ***environment-name***

Optionen

Name	Beschreibung
`--all`	Beendet alle Umgebungen in der Anwendung, die Anwendungsversionen der Anwendung sowie ihre gespeicherten Konfigurationen und löscht dann die Anwendung.
`--force`	Die Umgebung wird ohne Bestätigungsaufforderung beendet.
`--ignore-links`	Die Umgebung wird beendet, auch wenn abhängige Umgebungen mit Links zu dieser vorhanden sind. Weitere

Name	Beschreibung
	Informationen finden Sie im Abschnitt zu den Compose-Umgebungen.
`--timeout`	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.

Ausgabe

Bei Erfolg gibt der Befehl den Status des `terminate`-Vorgangs aus.

Beispiel

In der folgenden Beispielanforderung wird die Umgebung "tmp-dev" beendet.

```
$ eb terminate
The environment "tmp-dev" and all associated instances will be terminated.
To confirm, type the environment name: tmp-dev
2018-07-11 21:05:25    INFO: terminateEnvironment is starting.
2018-07-11 21:05:40    INFO: Deleted CloudWatch alarm named: awseb-e-2cpfjbra9a-stack-
AWSEBCloudwatchAlarmHigh-16V08YOF2KQ7U
2018-07-11 21:05:41    INFO: Deleted CloudWatch alarm named: awseb-e-2cpfjbra9a-stack-
AWSEBCloudwatchAlarmLow-6ZAWH9F20P7C
2018-07-11 21:06:42    INFO: Deleted Auto Scaling group policy named:
 arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:5d7d3e6b-
d59b-47c5-b102-3e11fe3047be:autoScalingGroupName/awseb-e-2cpfjbra9a-stack-
AWSEBAutoScalingGroup-7AXY7U13ZQ6E:policyName/awseb-e-2cpfjbra9a-stack-AWSEBAutoSca
lingScaleUpPolicy-1876U27JEC34J
2018-07-11 21:06:43    INFO: Deleted Auto Scaling group policy named:
 arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:29c6e7c7-7ac8-46fc-91f5-
cfabb65b985b:autoScalingGroupName/awseb-e-2cpfjbra9a-stack-
AWSEBAutoScalingGroup-7AXY7U13ZQ6E:policyName/awseb-e-2cpfjbra9a-stack-AWSEBAutoSca
lingScaleDownPolicy-SL4LHODMOMU
2018-07-11 21:06:48    INFO: Waiting for EC2 instances to terminate. This may take a
 few minutes.
2018-07-11 21:08:55    INFO: Deleted Auto Scaling group named: awseb-e-2cpfjbra9a-
stack-AWSEBAutoScalingGroup-7AXY7U13ZQ6E
2018-07-11 21:09:10    INFO: Deleted security group named: awseb-e-2cpfjbra9a-stack-
AWSEBSecurityGroup-XT4YYGFL7I99
2018-07-11 21:09:40    INFO: Deleted load balancer named: awseb-e-2-AWSEBLoa-
AK6RRYFQVV3S
```

```
2018-07-11 21:09:42    INFO: Deleting SNS topic for environment tmp-dev.
2018-07-11 21:09:52    INFO: terminateEnvironment completed successfully.
```

eb upgrade

Description

Aktualisiert die Plattform Ihrer Umgebung auf die neueste Version der Plattform, auf der Sie derzeit ausgeführt wird.

Wenn das Stammverzeichnis eine `platform.yaml`-Datei enthält, die eine benutzerdefinierte Plattform angibt, aktualisiert dieser Befehl die Umgebung auf die neueste Version der benutzerdefinierten Plattform, auf der sie derzeit ausgeführt wird.

Syntax

eb upgrade

eb upgrade ***environment-name***

Optionen

Name	Beschreibung
`--force`	Nimmt eine Aktualisierung vor, ohne dass Sie den Umgebungs namen bestätigen müssen, bevor Sie den Aktualisierungsvor gang starten.
`--noroll`	Aktualisiert alle Instances ohne fortlaufende Aktualisierungen, damit einige Instances während der Aktualisierung in Betrieb bleiben.
Allgemeine Optionen	

Ausgabe

Der Befehl zeigt eine Übersicht über die Änderung und fordert Sie auf, die Aktualisierung zu bestätigen, indem Sie den Umgebungsnamen eingeben. Bei Erfolg wird Ihre Umgebung aktualisiert und dann mit der neuesten Version der Plattform gestartet.

Beispiel

Im folgenden Beispiel wird die aktuelle Plattformversion der angegebenen Umgebung auf die aktuell verfügbare Plattformversion aktualisiert.

```
$ eb upgrade
Current platform: 64bit Amazon Linux 2014.09 v1.0.9 running Python 2.7
Latest platform:  64bit Amazon Linux 2014.09 v1.2.0 running Python 2.7

WARNING: This operation replaces your instances with minimal or zero downtime. You may
 cancel the upgrade after it has started by typing "eb abort".
You can also change your platform version by typing "eb clone" and then "eb swap".

To continue, type the environment name:
```

eb use

Description

Legt die angegebene Umgebung als Standardumgebung fest.

Bei der Verwendung von Git legt eb use die Standardumgebung für die aktuelle Verzweigung fest. Führen Sie diesen Befehl einmal in jeder Verzweigung aus, die Sie für Elastic Beanstalk bereitstellen möchten.

Syntax

eb use ***environment-name***

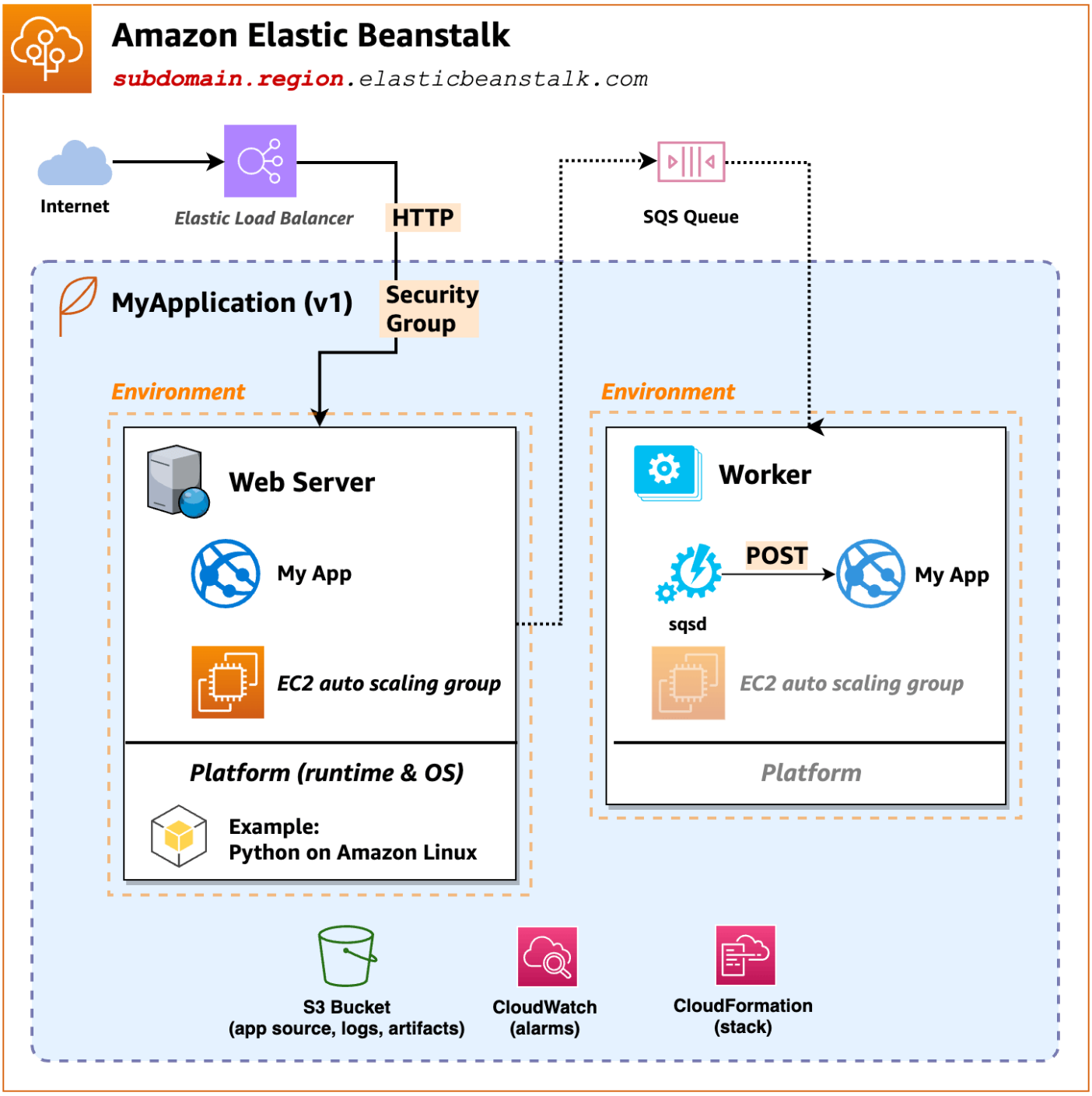
Optionen

Name	Description
`--source codecommit/` *repository-name*`/`*branch-name*	CodeCommit Endlager und Filiale.
`-r` *region* `--region` *region*	Ändern Sie die Region, in der Sie Umgebungen erstellen.

Name	Description
Allgemeine Optionen	

Konzepte in Elastic Beanstalk verstehen

Wenn Sie sich mit den Konzepten und Begriffen vertraut machen, erhalten Sie das nötige Verständnis für die Bereitstellung Ihrer Anwendungen mit Elastic Beanstalk.



Anwendung

Eine Elastic Beanstalk Beanstalk-Anwendung ist ein Container für Elastic Beanstalk Beanstalk-Komponenten, einschließlich Umgebungen, Versionen und Umgebungskonfigurationen. In einer Elastic Beanstalk Beanstalk-Anwendung verwalten Sie alle Ressourcen, die für die Ausführung Ihres Codes relevant sind.

Anwendungsversion

In Elastic Beanstalk bezieht sich eine Anwendungsversion auf eine spezifische, benannte Wiederholung des Bereitstellungscodes für eine Webanwendung. Eine Anwendungsversion verweist auf ein Amazon Simple Storage Service (Amazon S3)-Objekt mit dem bereitstellbaren Code wie eine Java WAR-Datei.

Eine Anwendungsversion ist Teil einer Anwendung. Anwendungen können verschiedene Versionen haben und jede Anwendungsversion ist einzigartig. In einer laufenden Umgebung können Sie jede Anwendungsversion bereitstellen, die Sie bereits in die Anwendung hochgeladen haben, oder eine neue Anwendungsversion hochladen und sofort bereitstellen. Sie könnten beispielsweise mehrere Anwendungsversionen hochladen, um die Unterschiede zwischen ihnen zu testen.

Umgebung

Eine Umgebung ist eine Sammlung von AWS Ressourcen, auf denen eine Anwendungsversion ausgeführt wird. In jeder Umgebung wird jeweils nur eine einzige Anwendungsversion ausgeführt. Sie können jedoch dieselbe Anwendungsversion oder verschiedene Anwendungsversionen in vielen Umgebungen gleichzeitig ausführen. Wenn Sie eine Umgebung erstellen, stellt Elastic Beanstalk die Ressourcen bereit, die in Ihrem AWS Konto benötigt werden, um die von Ihnen angegebene Anwendungsversion auszuführen.

Umgebungsebene

Wenn Sie eine Elastic Beanstalk-Umgebung starten, wählen Sie zuerst eine Umgebungsschicht aus. Die Umgebungsebene bestimmt die Art der Anwendung, die in der Umgebung ausgeführt wird, und bestimmt, welche Ressourcen Elastic Beanstalk zu ihrer Unterstützung bereitstellt. Anwendungen, die HTTP-Anforderungen senden, werden in einer Webserver-Umgebungsschicht ausgeführt. Eine Backend-Umgebung, die Aufgaben aus einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange abruft, wird in einer Worker-Umgebungsebene ausgeführt.

Konfiguration der Umgebung

Eine Umgebungskonfiguration definiert die Parameter und Einstellungen, die das Verhalten einer Umgebung und der zugeordneten Ressourcen definieren. Wenn Sie die Konfigurationseinstellungen einer Umgebung aktualisieren, wendet Elastic Beanstalk automatisch die Änderungen auf bestehende Ressourcen an oder löscht sie und stellt neue Ressourcen bereit (je nach Art der Änderung).

Gespeicherte Konfiguration

Eine gespeicherte Konfiguration ist eine Vorlage, die Sie als Ausgangspunkt zur Erstellung von eindeutigen Umgebungskonfigurationen verwenden können. Mithilfe der Elastic Beanstalk Beanstalk-Konsole, der EB-CLI oder der API können Sie gespeicherte Konfigurationen erstellen und ändern und sie auf Umgebungen anwenden. AWS CLI Die API und die AWS CLI bezeichnen gespeicherte Konfigurationen als Konfigurationsvorlagen.

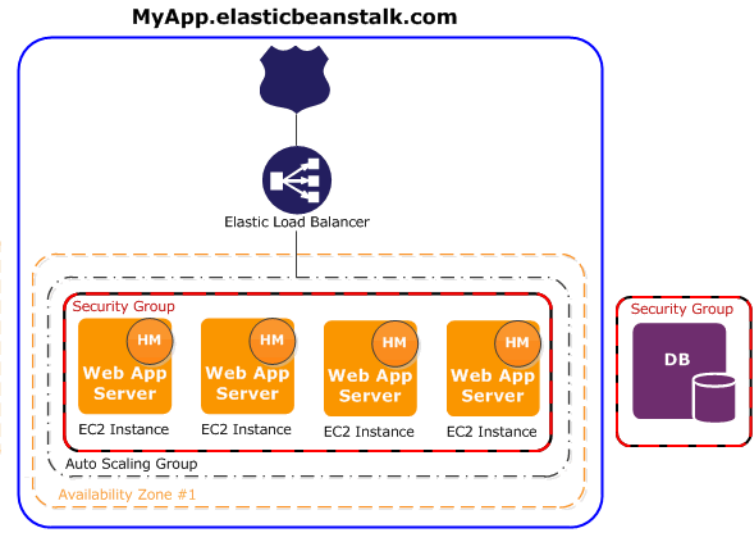
Plattform

Eine Plattform ist eine Kombination von Betriebssystem-, Programmiersprachen-Laufzeit-, Webserver-, Anwendungsserver- und Elastic Beanstalk-Komponenten. Sie entwerfen eine Webanwendung und richten diese auf eine Plattform aus. Elastic Beanstalk bietet eine Vielzahl von Plattformen, auf denen Sie Ihre Anwendungen aufbauen können.

Details hierzu finden Sie unter Elastic Beanstalk-Plattformen.

Elastic Beanstalk Beanstalk-Webserverumgebungen

Das folgende Diagramm zeigt ein Beispiel für eine Elastic Beanstalk-Architektur für eine Webserver-Umgebungsschicht und veranschaulicht, wie die Komponenten in dieser Art von Umgebung zusammen funktionieren.



Die Umgebung ist der Kern der Anwendung. Im Diagramm wird die Umgebung innerhalb der obersten durchgezogenen Linie dargestellt. Wenn Sie eine Umgebung erstellen, stellt Elastic Beanstalk die Ressourcen bereit, die für die Ausführung Ihrer Anwendung erforderlich sind. AWS Zu den für eine Umgebung erstellten Ressourcen gehören ein Elastic Load Balancer (ELB im Diagramm), eine Auto Scaling Scaling-Gruppe und eine oder mehrere Amazon Elastic Compute Cloud (Amazon EC2) -Instances.

Jede Umgebung hat einen CNAME (URL), der auf einen Load Balancer verweist. Die Umgebung verfügt über eine URL, z. B. `myapp.us-west-2.elasticbeanstalk.com`. Diese URL ist in Amazon Route 53 als Alias für eine Elastic Load Balancing-URL festgelegt – etwa `abcdef-123456.us-west-2.elb.amazonaws.com` – indem ein CNAME-Datensatz verwendet wird. Amazon Route 53 ist ein hochverfügbarer und skalierbarer Domain Name System (DNS)-Web-Service. Er bietet sicheres und zuverlässiges Routing an Ihre Infrastruktur. Ihr Domänenname, den Sie beim DNS-Anbieter registriert haben, leitet Anfragen an den CNAME weiter.

Der Load Balancer befindet sich vor den EC2 Amazon-Instances, die Teil einer Auto Scaling Scaling-Gruppe sind. Amazon EC2 Auto Scaling startet automatisch zusätzliche EC2 Amazon-Instances, um der zunehmenden Belastung Ihrer Anwendung Rechnung zu tragen. Wenn die Auslastung Ihrer Anwendung abnimmt, stoppt Amazon EC2 Auto Scaling die Instances, lässt aber immer mindestens eine Instance laufen.

Der Software-Stack, der auf den EC2 Amazon-Instances ausgeführt wird, hängt vom Containertyp ab. Ein Containertyp definiert die Infrastrukturtopologie und den Softwarestack, die für ein bestimmte Umgebung verwendet werden sollen. Eine Elastic Beanstalk-Umgebung mit einem Apache Tomcat-Container verwendet beispielsweise das Amazon Linux-Betriebssystem, Apache-Webserver und

Apache Tomcat-Software. Eine Liste mit unterstützten Containertypen finden Sie unter Von Elastic Beanstalk unterstützte Plattformen. Jede EC2 Amazon-Instance, die Ihre Anwendung ausführt, verwendet einen dieser Containertypen. Darüber hinaus läuft auf jeder EC2 Amazon-Instance eine Softwarekomponente namens Host Manager (HM). Der Host-Manager ist für Folgendes verantwortlich:

- Bereitstellen der Anwendung
- Aggregieren von Ereignissen und Metriken für den Abruf über die Konsole, die API oder die Befehlszeile
- Generieren von Ereignissen auf Instance-Ebene
- Überwachen der Anwendungs-Protokolldateien auf kritische Fehler
- Überwachen der Anwendungsserver
- Patchen von Instance-Komponenten
- Rotieren der Protokolldateien der Anwendung und Veröffentlichen in Amazon S3

Der Host-Manager meldet Metriken, Fehler und Ereignisse sowie den Status der Serverinstanz, die über die Elastic Beanstalk Beanstalk-Konsole verfügbar sind APIs, und. CLIs

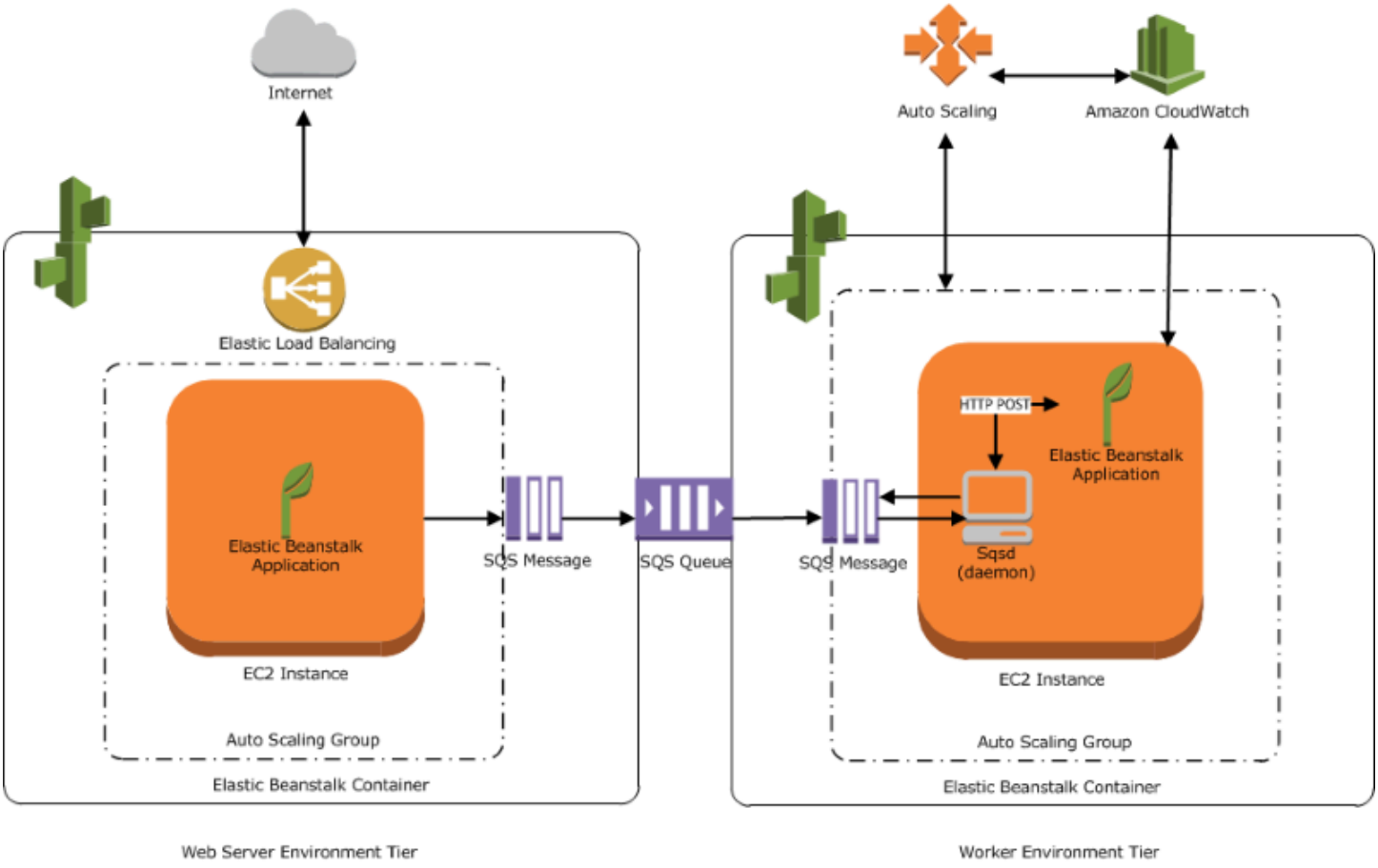
Die im Diagramm gezeigten EC2 Amazon-Instances sind Teil einer Sicherheitsgruppe. Eine Sicherheitsgruppe definiert Firewall-Regeln für Ihre Instances. Standardmäßig definiert Elastic Beanstalk eine Sicherheitsgruppe, die allen Benutzern die Verbindung über Port 80 (HTTP) erlaubt. Sie können mehr als eine Sicherheitsgruppe definieren. Sie können beispielsweise eine Sicherheitsgruppe für Ihren Datenbankserver definieren. Weitere Informationen zu EC2 Amazon-Sicherheitsgruppen und deren Konfiguration für Ihre Elastic Beanstalk Beanstalk-Anwendung finden Sie unter. EC2 Sicherheitsgruppen

Elastic-Beanstalk-Worker-Umgebungen

AWS Zu den Ressourcen, die für eine Worker-Umgebungsebene erstellt wurden, gehören eine Auto Scaling Scaling-Gruppe, eine oder mehrere EC2 Amazon-Instances und eine IAM-Rolle. Für die Worker-Umgebungs-Schicht erstellt Elastic Beanstalk außerdem eine Amazon SQS-Warteschlange und stellt diese bereit, falls Sie noch keine haben. Wenn Sie eine Worker-Umgebung starten, installiert Elastic Beanstalk die erforderlichen Unterstützungsdateien für die Programmiersprache Ihrer Wahl und einen Daemon auf jeder EC2 Instance in der Auto Scaling Scaling-Gruppe. Der Deamon liest Nachrichten aus einer Amazon SQS-Warteschlange. Der Daemon sendet Daten aus jeder gelesenen Nachricht zur Verarbeitung an die Webanwendung, die in der Worker-Umgebung

ausgeführt wird. Wenn Sie über mehrere Instances in Ihrer Worker-Umgebung verfügen, hat jede Instance einen eigenen Daemon, aber sie alle werden aus derselben Amazon SQS-Warteschlange gelesen.

Das folgende Diagramm zeigt die verschiedenen Komponenten und ihre Interaktionen zwischen Umgebungen und Diensten. AWS



Amazon CloudWatch wird für Alarme und Gesundheitsüberwachung verwendet. Weitere Informationen finden Sie unter Grundlegende Zustandsberichte.

Weitere Informationen darüber, wie die Worker-Umgebungsschicht funktioniert, finden Sie unter Elastic-Beanstalk-Worker-Umgebungen.

Überlegungen zum Design Ihrer Elastic Beanstalk Beanstalk-Anwendungen

Da Anwendungen, die mithilfe von AWS Cloud Ressourcen bereitgestellt werden, nur mit Ressourcen AWS Elastic Beanstalk ausgeführt werden, sollten Sie mehrere Konfigurationsfaktoren

berücksichtigen, um Ihre Anwendungen zu optimieren: Skalierbarkeit, Sicherheit, persistenter Speicher, Fehlertoleranz, Bereitstellung von Inhalten, Softwareupdates und Patches sowie Konnektivität. Jeder dieser Aspekte wird in diesem Thema separat behandelt. Eine umfassende Liste technischer AWS Whitepapers zu Themen wie Architektur sowie Sicherheit und Wirtschaft finden Sie unter AWS Cloud-Computing-Whitepapers.

Skalierbarkeit

Wenn Sie in einer physischen Hardware-Umgebung arbeiten, können Sie im Gegensatz zu einer Cloud-Umgebung die Skalierbarkeit auf zwei Arten angehen. Entweder können Sie durch vertikale Skalierung hochskalieren oder durch horizontale Skalierung aufskalieren. Der Hochskalierungsansatz erfordert, dass Sie in leistungsstarke Hardware investieren, die die steigenden Anforderungen Ihres Unternehmens unterstützen kann. Der Aufskalierungsansatz setzt voraus, dass Sie einem verteilten Anlagemodell folgen. Daher können Ihre Hardware- und Anwendungsakquisitionen gezielter sein, Ihre Datensätze werden verbunden und Ihr Design ist serviceorientiert. Das Hochskalierungsansatz kann sehr teuer sein und es besteht weiterhin das Risiko, dass Ihr Bedarf Ihre Kapazität übersteigt. In dieser Hinsicht ist der Aufskalierungsansatz normalerweise effektiver. Wenn Sie ihn verwenden, müssen Sie jedoch in der Lage sein, den Bedarf in regelmäßigen Abständen vorherzusagen und Infrastruktur in Teilen bereitzustellen, um diesen Bedarf zu decken. Dieser Ansatz führt oftmals zu nicht genutzten Kapazitäten und erfordert eine sorgfältige Überwachung.

Durch die Migration in die Cloud können Sie Ihre Infrastruktur durch Ausnutzen der Elastizität der Cloud anpassen. Elastizität hilft, die Erfassung und Freigabe von Ressourcen zu rationalisieren. Mit ihr kann Ihre Infrastruktur rasch mit schwankendem Bedarf aufskaliert werden. Sie können mit Ihren Auto-Scaling-Einstellungen nach oben oder unten skalieren, basierend auf Metriken der Ressourcen in Ihrer Umgebung. Sie können beispielsweise Metriken (wie Serverauslastung oder NetzwerkI/O. You can use Auto Scaling for compute capacity to be added automatically whenever usage rises and for it to be removed whenever usage drops. You can publish system metrics (for example, CPU, memory, disk I/O, and network I/O) auf Amazon festlegen CloudWatch. Anschließend können Sie Alarme konfigurieren, CloudWatch um Auto Scaling Scaling-Aktionen auszulösen oder Benachrichtigungen auf der Grundlage dieser Metriken zu senden. Weitere Informationen zur Konfiguration von Auto Scaling finden Sie unter Auto Scaling Ihrer Elastic Beanstalk Beanstalk-Umgebungsinstanzen.

Elastic-Beanstalk-Anwendungen sollten darüber hinaus möglichst zustandslos sein, dank lose gekoppelter, fehlertoleranter Komponenten, die bei Bedarf aufskaliert werden können. Weitere Informationen zum Entwerfen skalierbarer Anwendungsarchitekturen für AWS finden Sie unter AWS Well-Architected Framework.

Sicherheit

Sicherheit AWS ist eine gemeinsame Verantwortung. Amazon Web Services schützt die physischen Ressourcen in Ihrer Umgebung und stellt sicher, dass die Cloud ein sicherer Ort zum Ausführen von Anwendungen ist. Sie sind für die Sicherheit von Daten in und aus Ihrer Elastic-Beanstalk-Umgebung und die Sicherheit Ihrer Anwendung verantwortlich.

Konfigurieren Sie SSL, um Informationen zu schützen, die zwischen Ihrer Anwendung und den Clients fließen. Dazu benötigen Sie ein kostenloses Zertifikat von AWS Certificate Manager (ACM). Wenn Sie bereits ein Zertifikat einer externen Zertifizierungsstelle (CA) besitzen, können Sie ACM verwenden, um dieses Zertifikat zu importieren. Andernfalls können Sie es mit dem importieren AWS CLI.

Wenn ACM in Ihrem nicht verfügbar ist AWS-Region, können Sie ein Zertifikat von einer externen Zertifizierungsstelle erwerben, z. VeriSign B. von Entrust. Verwenden Sie dann AWS Command Line Interface (AWS CLI), um ein Drittanbieter- oder selbstsigniertes Zertifikat und einen privaten Schlüssel in AWS Identity and Access Management (IAM) hochzuladen. Der öffentliche Schlüssel des Zertifikats authentifiziert Ihren Server gegenüber dem Browser. Außerdem dient er als Grundlage für die Erstellung des gemeinsamen Sitzungsschlüssels, der die Daten in beide Richtungen verschlüsselt. Weitere Anweisungen zum Erstellen, Hochladen und Zuweisen eines SSL-Zertifikats zu Ihrer Umgebung finden Sie unter Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung.

Wenn Sie ein SSL-Zertifikat für Ihre Umgebung konfigurieren, werden Ihre Daten zwischen dem Client und der Elastic-Load-Balancing-Lastenverteilung verschlüsselt. Standardmäßig wird die Verschlüsselung am Load Balancer beendet und der Verkehr zwischen dem Load Balancer und EC2 Amazon-Instances ist unverschlüsselt.

Persistenter Speicher

Elastic Beanstalk Beanstalk-Anwendungen werden auf EC2 Amazon-Instances ausgeführt, die keinen persistenten lokalen Speicher haben. Wenn die EC2 Amazon-Instances beendet werden, wird das lokale Dateisystem nicht gespeichert. Neue EC2 Amazon-Instances beginnen mit einem Standarddateisystem. Wir empfehlen Ihnen, Ihre Anwendung zum Speichern von Daten in einer persistenten Datenquelle zu konfigurieren. AWS bietet eine Reihe von persistenten Speicherservices, von denen Sie Gebrauch machen können. In der folgende Tabelle sind sie aufgelistet.

Storage Service	Servicedokumentation	Elastic Beanstalk-Integration
Amazon S3	Amazon Simple Storage Service – Dokumentation	Verwenden von Elastic Beanstalk mit Amazon S3
Amazon Elastic File System	Amazon Elastic File System – Dokumentation	Elastic Beanstalk mit Amazon Elastic File System verwenden
Amazon Elastic Block Store	Amazon Elastic Block Store Funktionsleitfaden: Elastic Block Store	
Amazon-DynamoDB	Amazon DynamoDB – Dokumentation	Elastic Beanstalk mit Amazon DynamoDB verwenden
Amazon Relational Database Service (RDS)	Amazon Relational Database Service – Dokumentation	Verwenden von Elastic Beanstalk mit Amazon RDS

Note

Elastic Beanstalk erstellt einen Webapp-Benutzer, den Sie als Eigentümer von Anwendungsverzeichnissen auf Instances einrichten können. EC2 Für Amazon-Linux-2-Plattformversionen, die am 3. Februar 2022 oder danach veröffentlicht werden, weist Elastic Beanstalk dem webapp-Benutzer eine UID (Benutzer-ID) und einen gid-Wert (Gruppen-ID-Wert) von 900 für neue Umgebungen zu. Dasselbe geschieht für vorhandene Umgebungen nach einem Plattformversionsupdate. Dieser Ansatz behält eine konsistente Zugriffsberechtigung für den webapp-Benutzer auf den permanenten Dateisystemspeicher bei.
In dem unwahrscheinlichen Fall, dass ein anderer Benutzer oder Prozess bereits 900 verwendet, setzt das Betriebssystem die uid und gid des webapp-Benutzers auf einen anderen Wert zurück. Führen Sie den Linux-Befehl id webapp auf Ihren EC2 Instances aus, um die UID- und GID-Werte zu überprüfen, die dem Webapp-Benutzer zugewiesen sind.

Fehlertoleranz

In der Regel sollten Sie pessimistisch bei der Entwicklung einer Architektur für die Cloud sein. Nutzen Sie die Elastizität, die dadurch ermöglicht wird. Berücksichtigen Sie beim Design, der Implementierung und Bereitstellung immer eine automatische Wiederherstellung nach einem Ausfall. Verwenden Sie mehrere Availability Zones für Ihre EC2 Amazon-Instances und für Amazon RDS. Availability Zonen basieren auf dem gleichen Konzept wie logische Rechenzentren. Nutzen Sie Amazon CloudWatch , um sich einen besseren Überblick über den Zustand Ihrer Elastic Beanstalk Beanstalk-Anwendung zu verschaffen und bei Hardwarefehlern oder Leistungseinbußen geeignete Maßnahmen zu ergreifen. Konfigurieren Sie Ihre Auto Scaling Scaling-Einstellungen so, dass Ihre Flotte von EC2 Amazon-Instances auf einer festen Größe bleibt, sodass fehlerhafte EC2 Amazon-Instances durch neue ersetzt werden. Wenn Sie Amazon RDS verwenden, richten Sie den Aufbewahrungszeitraum für Backups ein, damit Amazon RDS automatisch Backups erstellen kann.

Bereitstellung von Inhalten

Wenn Benutzer eine Verbindung mit Ihrer Website herstellen, werden ihre Anforderungen über eine Reihe einzelner Netzwerke geleitet. Daher kann es sein, dass Anwender eine schlechtere Leistung durch hohe Latenz bemerken. Amazon CloudFront kann Ihnen helfen, Latenzprobleme zu beheben, indem es Ihre Webinhalte wie Bilder und Videos über ein Netzwerk von Edge-Standorten auf der ganzen Welt verteilt. Benutzeranfragen werden an den nächstgelegenen Edge-Standort weitergeleitet, sodass Inhalte mit der bestmöglichen Leistung bereitgestellt werden. CloudFront funktioniert nahtlos mit Amazon S3, das die ursprünglichen, definitiven Versionen Ihrer Dateien dauerhaft speichert. Weitere Informationen zu Amazon CloudFront finden Sie im Amazon CloudFront Developer Guide.

Software-Aktualisierungen und Patches

AWS Elastic Beanstalk veröffentlicht regelmäßig Plattform-Updates, um Problembehebungen, Softwareupdates und neue Funktionen bereitzustellen. Elastic Beanstalk bietet mehrere Optionen zur Handhabung von Plattformupdates. Mit verwalteten Plattformupdates wird Ihre Umgebung während eines geplanten Wartungsfensters automatisch auf die neueste Version einer Plattform aktualisiert, während Ihre Anwendung in Betrieb bleibt. In Umgebungen, die am 25. November 2019 oder später mit der Elastic-Beanstalk-Konsole erstellt wurden, werden verwaltete Updates standardmäßig aktiviert (sofern möglich). Sie können Updates auch manuell mit der Elastic Beanstalk-Konsole oder der EB CLI starten.

Konnektivität

Elastic Beanstalk muss eine Verbindung zu den Instances in Ihrer Umgebung herstellen können, um Bereitstellungen abzuschließen. Wenn Sie eine Elastic Beanstalk-Anwendung in einer Amazon VPC bereitstellen, hängt die Konfiguration für die Konnektivität, die Sie erstellen, vom Typ der Amazon VPC-Umgebung ab, die Sie erstellen:

- Für Einzel-Instance-Umgebungen ist keine zusätzliche Konfiguration erforderlich. Das liegt daran, dass Elastic Beanstalk in diesen Umgebungen jeder EC2 Amazon-Instance eine öffentliche Elastic IP-Adresse zuweist, die es der Instance ermöglicht, direkt mit dem Internet zu kommunizieren.
- Für skalierbare Umgebungen mit Lastausgleich in einer Amazon VPC mit öffentlichen und privaten Subnetzen müssen Sie die folgenden Schritte ausführen:
  - Erstellen Sie einen Load Balancer im öffentlichen Subnetz, um eingehenden Datenverkehr aus dem Internet an die Amazon-Instances weiterzuleiten. EC2
  - Erstellen Sie ein NAT-Gerät (Network Address Translation), um ausgehenden Datenverkehr von EC2 Amazon-Instances in privaten Subnetzen ins Internet weiterzuleiten.
  - Erstellen Sie Regeln für eingehendes und ausgehendes Routing für die EC2 Amazon-Instances innerhalb des privaten Subnetzes.
  - Wenn Sie eine NAT-Instance verwenden, konfigurieren Sie die Sicherheitsgruppen für die NAT-Instance und EC2 Amazon-Instances, um die Internetkommunikation zu ermöglichen.
- Für eine skalierbare Umgebung mit Lastenverteilung in einer Amazon VPC mit einem öffentlichen Subnetz ist keine zusätzliche Konfiguration erforderlich. Dies liegt daran, dass Ihre EC2 Amazon-Instances in dieser Umgebung mit einer öffentlichen IP-Adresse konfiguriert sind, die es den Instances ermöglicht, mit dem Internet zu kommunizieren.

Weitere Informationen zum Verwenden von Elastic Beanstalk mit Amazon VPC finden Sie unter Verwenden von Elastic Beanstalk mit Amazon VPC.

Elastic Beanstalk Beanstalk-Anwendungen verwalten

In diesem Kapitel wird beschrieben, wie Sie Ihre Elastic Beanstalk Beanstalk-Anwendungen verwalten und konfigurieren. Der erste Schritt bei der Verwendung AWS Elastic Beanstalk besteht darin, eine Anwendung zu erstellen, die Ihre Webanwendung in darstellt. AWS In Elastic Beanstalk dient eine Anwendung als Container für die Umgebungen, in denen die Web-App ausgeführt wird, sowie für Quellcodeversionen der Web-App, gespeicherte Konfigurationen, Protokolle und andere Artefakte, die Sie in Elastic Beanstalk erstellen.

So erstellen Sie eine Anwendung

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Applications (Anwendungen) und anschließend Create a new application (Neue Anwendung erstellen).
3. Verwenden Sie das Formular auf dem Bildschirm, um einen Anwendungsnamen anzugeben.
4. (Optional) Geben Sie eine Beschreibung ein und fügen Sie Tag-Schlüssel und -Werte hinzu.
5. Wählen Sie Erstellen aus.

Nachdem die Anwendung generiert ist, werden Sie von der Konsole aufgefordert, eine Umgebung dafür zu erstellen. Detaillierte Informationen zu allen verfügbaren Optionen finden Sie unter Erstellen einer Elastic Beanstalk-Umgebung.

Wenn Sie eine Anwendung nicht mehr benötigen, können Sie sie löschen.

> **Warning**
>
> Wenn Sie eine Anwendung löschen, werden alle zugeordneten Umgebungen und sämtliche Anwendungsversionen sowie gespeicherte Konfigurationen für die Anwendung gelöscht.

So löschen Sie eine -Anwendung

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Applications (Anwendungen) und wählen Sie dann Ihre Anwendung in der Liste aus.

3. Wählen Sie Actions (Aktionen) und anschließend Delete Application (Anwendung löschen).

Themen

- Elastic Beanstalk-Anwendungsmanagement-Konsole
- Verwalten von Anwendungsversionen
- Ein Elastic Beanstalk Beanstalk-Anwendungsquellpaket erstellen
- Verwenden der EB-CLI mit AWS CodeBuild
- Markieren von Anwendungen
- Markieren von Elastic Beanstalk-Anwendungsressourcen

Elastic Beanstalk-Anwendungsmanagement-Konsole

In diesem Thema wird erklärt, wie Sie mit der AWS Elastic Beanstalk Konsole Anwendungen, Anwendungsversionen und gespeicherte Konfigurationen verwalten können.

So greifen Sie auf die Konsole zur Anwendungsverwaltung zu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

   Die Anwendungsübersichtsseite zeigt eine Liste mit einer Übersicht über alle mit der Anwendung verknüpften Umgebungen.
3. Sie haben verschiedene Möglichkeiten, um fortzufahren:

   a. Im Drop-down-Menü „Aktionen“ können Sie eine der Anwendungsverwaltungsaktionen auswählen: Umgebung erstellen, Anwendung löschen, Anwendungsversionen anzeigen, Gespeicherte Konfigurationen anzeigen, Beendete Umgebung wiederherstellen.

      Um eine Umgebung in dieser Anwendung zu starten, können Sie direkt Umgebung erstellen wählen. Details hierzu finden Sie unter the section called “Erstellen von Umgebungen”.

   b. Auf der Seite werden die Umgebungsnamen neben den Anwendungen aufgeführt, die in einer Umgebung bereitgestellt werden. Wählen Sie einen Umgebungsnamen, um zur Konsole zur Umgebungsverwaltung für diese Umgebung zu wechseln, in der Sie die Umgebung konfigurieren, überwachen oder verwalten können.

c. Wenn Sie eine Anwendung aus der Liste auswählen, wird die Anwendung im linken Navigationsbereich aufgeführt.

   - Wählen Sie im Navigationsbereich Application versions (Anwendungsversionen) hinter dem Anwendungsnamen, um die Anwendungsversionen für Ihre Anwendung anzuzeigen und zu verwalten.

     Eine Anwendungsversion ist eine hochgeladene Version Ihres Anwendungscodes. Sie können neue Versionen hochladen, eine vorhandene Version auf einer beliebigen Umgebung der Anwendung bereitstellen oder alte Versionen löschen. Weitere Informationen finden Sie unter Verwalten von Anwendungsversionen.

   - Wählen Sie Saved configurations (Gespeicherte Konfigurationen) hinter dem Anwendungsnamen im Navigationsbereich, um Konfigurationen anzuzeigen und zu verwalten, die in ausgeführten Umgebungen gespeichert wurden.

     Eine gespeicherte Konfiguration ist eine Sammlung von Einstellungen, die Sie verwenden können, um die Einstellungen einer Umgebung in einem vorherigen Status wiederherzustellen oder eine Umgebung mit den gleichen Einstellungen zu erstellen. Weitere Informationen finden Sie unter Verwenden gespeicherter Elastic Beanstalk-Konfigurationen.

Verwalten von Anwendungsversionen

In diesem Thema werden Anwendungsversionen und deren Erstellung und Verwaltung erläutert.

Elastic Beanstalk erstellt eine Anwendungsversion, wenn Sie Quellcode hochladen. Dies erfolgt in der Regel beim Erstellen einer Umgebung oder dem Hochladen und Bereitstellen von Code mit der Environment Management Console oder EB CLI. Elastic Beanstalk löscht diese Anwendungsversionen gemäß Lebenszyklus-Richtlinie der Anwendung, und wenn Sie die Anwendung löschen. Details zu den Lebenszyklusrichtlinien von Anwendungen finden Sie unter Konfigurieren von Einstellungen für den Lebenszyklus der Anwendungsversion.

Sie können auch einen Quell-Bundle ohne Bereitstellung von der Konsole zur Anwendungsverwaltung oder mit dem EB CLI-Befehl eb appversion hochladen. Elastic Beanstalk speichert Quell-Bundles in Amazon Simple Storage Service (Amazon S3) und löscht sie nicht automatisch.

Sie können eine Anwendungsversion mit Tags markieren, wenn Sie sie erstellen, und Tags von vorhandenen Anwendungsversionen bearbeiten. Details hierzu finden Sie unter Markieren von Anwendungsversionen.

Anwendungsversionen erstellen

Sie können auch eine neue Anwendungsversion mit der EB CLI erstellen. Weitere Informationen finden Sie unter eb appversion im Kapitel EB CLI-Befehle.

> **Note**
>
> Im Lauf der Zeit kann Ihre Anwendung viele Anwendungsversionen ansammeln. Um Speicherplatz zu sparen und das Kontingent für die Anwendungsversion nicht auszureizen, sollten Sie nicht mehr benötigte Anwendungsversionen löschen.

Die Datei, die Sie im folgenden Verfahren angeben, ist mit Ihrer Anwendung verknüpft. Sie können die Anwendungsversion in einer neuen oder vorhandenen Umgebung bereitstellen.

So erstellen Sie eine neue Anwendungsversion

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen)aus.
4. Klicken Sie auf Upload. Verwenden Sie das Formular auf dem Bildschirm, um das Quellpaket Ihrer Anwendung hochzuladen.

   > **Note**
   >
   > Die maximale Dateigröße des Quell-Bundles beträgt 500 MB.

5. Geben Sie optional eine kurze Beschreibung ein und fügen Sie Tag-Schlüssel und -Werte hinzu.
6. Klicken Sie auf Upload.

Anwendungsversionen löschen

Sie können auch eine Anwendungsversion mit der EB CLI löschen. Weitere Informationen finden Sie unter eb appversion im Kapitel EB CLI-Befehle.

> **Note**
>
> Das Löschen einer Anwendungsversion hat keine Auswirkungen auf aktuell ausgeführte Umgebungen dieser Version.

Sie können Elastic Beanstalk auch so konfigurieren, dass alte Versionen automatisch gelöscht werden, indem Sie die Lebenszykluseinstellungen der Anwendungsversion konfigurieren. Wenn Sie diese Lebenszykluseinstellungen konfigurieren, werden sie angewendet, wenn Sie neue Anwendungsversionen erstellen. Wenn Sie maximal 25 Anwendungsversionen konfigurieren, löscht Elastic Beanstalk die älteste Version, wenn Sie eine 26. Version hochladen. Wenn Sie ein Höchstalter von 90 Tagen festlegen, werden alle Versionen, die älter sind als 90 Tage alt, gelöscht, sobald Sie eine neue Version hochladen. Details hierzu finden Sie unter the section called “Lebenszyklus der Version”.

So löschen Sie eine Anwendungsversion

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen)aus.
4. Wählen Sie eine oder mehrere Anwendungsversionen aus, die Sie löschen möchten.
5. Wählen Sie Actions (Aktionen) und dann Delete (Löschen).
6. (Optional) Um das Programm-Quellbündel für diese Anwendungsversionen in Ihrem Amazon Simple Storage Service (Amazon S3)-Bucket zu belassen, deaktivieren Sie die option Delete versions from Amazon S3 (Versionen von Amazon S3 löschen}.
7. Wählen Sie Löschen aus.

Wenn Sie sich nicht dafür entscheiden, das Quell-Bundle von Amazon S3 zu löschen, löscht Elastic Beanstalk trotzdem die Version aus seinen Datensätzen. Das Quell-Bundle verbleibt jedoch in Ihrem Elastic Beanstalk Storage-Bucket. Das Kontingent für die Anwendungsversion gilt nur für die Elastic Beanstalk-Versionen. Daher können Sie Versionen löschen, um das Kontingent nicht zu überschreiten und um alle Quell-Bundles in Amazon S3 zu erhalten.

> **Note**
>
> Das Anwendungsversionslimit gilt nicht für Quell-Bundles, aber es können Ihnen dennoch Amazon S3-Gebühren entstehen, und sie behalten möglicherweise personenbezogene Daten länger als notwendig. Elastic Beanstalk löscht Quell-Bundles niemals automatisch. Sie sollten Quell-Bundles löschen, wenn Sie sie nicht mehr benötigen.

Konfigurieren von Einstellungen für den Lebenszyklus der Anwendungsversion

In diesem Thema werden die Richtlinien und Kontingente erklärt, die Elastic Beanstalk auf die Versionen Ihrer Anwendung in einer bestimmten Umgebung anwendet, einschließlich der Dauer, für die eine Anwendungsversion in einer Umgebung verbleibt.

Jedes Mal, wenn Sie eine neue Version Ihrer Anwendung mit der Elastic Beanstalk-Konsole oder der EB CLI hochladen, erstellt Elastic Beanstalk eine Anwendungsversion. Falls Sie nicht länger benötigte Versionen nicht löschen, erreichen Sie irgendwann das Anwendungsversionskontingent und können keine neuen Versionen dieser Anwendung mehr erstellen.

Das Erreichen dieses Kontingente können Sie vermeiden, indem Sie den Anwendungen eine Lebenszyklusrichtlinie für die Anwendungsversion hinzufügen. Mit einer Lebenszyklus-Richtlinie wird Elastic Beanstalk angewiesen, Anwendungsversionen mit einem bestimmten Alter oder bei Überschreitung eines vorgegebenen Maximalwerts an Versionen zu löschen.

Elastic Beanstalk wendet die Lebenszyklus-Richtlinie der Anwendung bei jeder neu erstellten Anwendungsversion an und löscht dabei jedes Mal bis zu 100 Versionen. Elastic Beanstalk löscht alte Versionen nach dem Erstellen der neuen Version. Die neue Version wird nicht hinsichtlich der maximalen Anzahl an Versionen berücksichtigt, die Sie in der Richtlinie festgelegt haben.

Elastic Beanstalk löscht weder Anwendungsversionen, die derzeit in einer Umgebung ausgeführt werden, noch solche, die in Umgebungen bereitgestellt wurden, die weniger als zehn Wochen vor dem Inkrafttreten der Richtlinie beendet wurden.

Das Anwendungsversionskontingent gilt für alle Anwendungen in einer Region. Bei mehreren Anwendungen sollten Sie für jede eine angemessene Lebenszyklus-Richtlinie konfigurieren, damit das Kontingent nicht erreicht wird. Wenn Sie z. B. 10 Anwendungen in einer Region haben und das Kontingent 1.000 Anwendungsversionen beträgt, sollten Sie in Erwägung ziehen, eine Lebenszyklusrichtlinie mit einem Kontingent von 99 Anwendungsversionen für alle Anwendungen festzulegen oder andere Werte in jeder Anwendung festzulegen, solange die Summe weniger als 1.000 Anwendungsversionen beträgt. Elastic Beanstalk wendet die Richtlinie nur an, sofern die Anwendungsversion erfolgreich erstellt wurde. Falls das Limit also bereits erreicht ist, müssen Sie einige Versionen manuell löschen, bevor Sie eine neue Version erstellen können.

Standardmäßig belässt Elastic Beanstalk das Quell-Bundle der Anwendungsversion in Amazon S3, um Datenverlust zu verhindern. Sie können das Quell-Bundle löschen, um Speicherplatz freizusetzen.

Sie können die Lebenszykluseinstellungen über die Elastic Beanstalk-CLI und festlegen. APIs Weitere Informationen finden Sie eb appversion unter CreateApplication(unter Verwendung des `ResourceLifecycleConfig` Parameters). UpdateApplicationResourceLifecycle

Festlegen der Einstellungen für den Anwendungslebenszyklus in der Konsole

Sie können die Lebenszyklus-Einstellungen in der Elastic Beanstalk-Konsole festlegen.

So legen Sie die Einstellungen für den Anwendungslebenszyklus fest

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen)aus.
4. Wählen Sie Einstellungen aus.
5. Verwenden Sie das Formular auf dem Bildschirm, um Einstellungen für den Anwendungslebenszyklus zu konfigurieren.
6. Wählen Sie Speichern.

Auf der Einstellungsseite haben Sie folgende Möglichkeiten.

- Konfigurieren Sie Lebenszykluseinstellungen basierend auf der Gesamtzahl der Anwendungsversionen oder dem Alter der Anwendungsversionen.
- Legen Sie fest, ob das Quell-Bundle in S3 beim Löschen einer Anwendungsversion ebenfalls gelöscht werden soll.
- Wählen Sie die Servicerolle aus, unter der die Anwendungsversion gelöscht wird. Um alle erforderlichen Berechtigungen zum Löschen der Version aufzunehmen, wählen Sie die standardmäßige Elastic Beanstalk-Servicerolle, `aws-elasticbeanstalk-service-role`, oder eine andere Servicerolle unter Verwendung der von Elastic Beanstalk verwalteten Service-Richtlinien. Weitere Informationen finden Sie unter Rollen von Elastic Beanstalk Service verwalten.

Markieren von Anwendungsversionen

In diesem Thema werden die Vorteile des Taggens Ihrer Elastic Beanstalk Beanstalk-Anwendungsversionen und die Verwaltung der Tags erläutert.

Sie können Tags auf Ihre AWS Elastic Beanstalk Anwendungsversionen anwenden. Tags sind Schlüssel-Wert-Paare, die Ressourcen zugeordnet AWS sind. Weitere Informationen zum Elastic Beanstalk-Ressourcen-Tagging, zu Anwendungsfälle, Einschränkungen für Tag-Schlüssel und -Werte sowie zu unterstützten Ressourcentypen finden Sie unter Markieren von Elastic Beanstalk-Anwendungsressourcen.

Sie können Tags angeben, wenn Sie eine Anwendung erstellen. In einer vorhandenen Anwendungsversion können Sie Tags hinzufügen oder entfernen sowie die Werte von vorhandenen Tags aktualisieren. Sie können jeder Anwendungsversion bis zu 50 Tags hinzufügen.

Hinzufügen von Tags während der Anwendungsversionserstellung

Wenn Sie über die Elastic Beanstalk-Konsole eine Umgebung erstellen und eine Version des Anwendungscodes hochladen, können Sie Tag-Schlüssel und -Werte angeben, um sie der neuen Anwendungsversion zuzuordnen.

Die Elastic Beanstalk-Konsole können Sie auch verwenden, um eine Anwendungsversion hochzuladen, ohne sie sofort in einer Umgebung zu verwenden. Sie können Tag-Schlüssel und -Werte angeben, wenn Sie eine Anwendungsversion hochladen.

Fügen Sie mit dem AWS CLI oder anderen API-basierten Clients Tags hinzu, indem Sie den `--tags` Parameter im Befehl verwenden. create-application-version

```
$ aws elasticbeanstalk create-application-version \
```

```
    --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
    --application-name my-app --version-label v1
```

Wenn Sie die EB CLI zum Erstellen oder Aktualisieren einer Umgebung verwenden, wird eine Anwendungsversion aus dem Code erstellt, den Sie bereitstellen. Es gibt keine direkte Methode, um eine Anwendungsversion bei der Erstellung über die EB CLI mit Tags zu markieren. Weitere Informationen zum Hinzufügen von Tags zu einer vorhandenen Anwendungsversion finden Sie im folgenden Abschnitt.

Verwalten von Tags einer vorhandenen Anwendungsversion

Sie können Tags in einer vorhandenen Elastic Beanstalk-Anwendungsversion hinzufügen, aktualisieren und löschen.

So verwalten Sie die Tags einer Anwendungsversion mit der Elastic Beanstalk-Konsole:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen)aus.
4. Wählen Sie die Anwendungsversion aus, die Sie verwalten möchten.
5. Klicken Sie auf Actions (Aktionen) und dann auf Manage Tags (Tags verwalten).
6. Verwenden Sie das Formular auf dem Bildschirm, um Tags hinzuzufügen, zu aktualisieren oder zu löschen.
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Wenn Sie Ihre Anwendungsversion mit der EB CLI aktualisieren, verwenden Sie den Befehl eb tags zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags.

Beispiel: Der folgende Befehl listet die Tags in einer Anwendungsversion auf.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

Mit dem folgenden Befehl wird das Tag `mytag1` aktualisiert und das Tag `mytag2` gelöscht.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:applicationversion/my-app/my-version"
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter `eb tags`.

Verwenden Sie bei den AWS CLI oder anderen API-basierten Clients den list-tags-for-resource Befehl, um die Tags einer Anwendungsversion aufzulisten.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-
version"
```

Verwenden Sie den Befehl update-tags-for-resource zum Hinzufügen, Aktualisieren oder Löschen von Tags in einer Anwendungsversion.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:applicationversion/my-app/my-version"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter `--tags-to-add` des Befehls update-tags-for-resource an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

> **Note**
>
> Um einige der EB-CLI und AWS CLI Befehle mit einer Elastic Beanstalk Beanstalk-Anwendungsversion zu verwenden, benötigen Sie den ARN der Anwendungsversion. Verwenden Sie den folgenden Befehl, um den ARN abzurufen.
>
> ```
> $ aws elasticbeanstalk describe-application-versions --application-name my-app
>  --version-label my-version
> ```

Ein Elastic Beanstalk Beanstalk-Anwendungsquellpaket erstellen

In diesem Thema wird erklärt, wie Sie Ihre Anwendungsquelldateien in einem Quellpaket auf Elastic Beanstalk hochladen. Es werden die Anforderungen an ein Quellpaket, die Struktur und die Methoden zur Erstellung eines solchen Pakets erläutert.

Wenn Sie die AWS Elastic Beanstalk Konsole verwenden, um eine neue Anwendung oder eine Anwendungsversion bereitzustellen, müssen Sie die Dateien für die Anwendung in einem Quellpaket hochladen. Das Quell-Bundle muss folgende Anforderungen erfüllen:

- Es muss aus einer einzigen `ZIP`- oder `WAR`-Datei bestehen (Sie können mehrere `WAR`-Dateien in eine `ZIP`-Datei einbinden).
- Darf nicht größer sein als 500 MB
- Es darf keinen übergeordneten Ordner bzw. kein Top-Level-Verzeichnis enthalten (Unterverzeichnisse sind in Ordnung).

Soll eine Worker-Anwendung zur Verarbeitung regelmäßiger Hintergrundaufgaben bereitgestellt werden, muss das Quell-Bundle der Anwendung zudem eine `cron.yaml`-Datei enthalten. Weitere Informationen finden Sie unter Regelmäßige Aufgaben.

Wenn Sie Ihre Anwendung mit dem Elastic Beanstalk Command Line Interface (EB CLI), dem AWS Toolkit for Eclipse oder dem AWS Toolkit for Visual Studio bereitstellen, wird die ZIP- oder WAR-Datei automatisch korrekt strukturiert. Weitere Informationen finden Sie unter Einrichtung der EB-Befehlszeilenschnittstelle (EB CLI) zur Verwaltung von Elastic Beanstalk, Bereitstellung von Java-Anwendungen mit Elastic Beanstalk und Die AWS Toolkit for Visual Studio.

Sections

- Erstellen eines Quell-Bundles mit der Befehlszeile
- Erstellen eines Quell-Bundles mit Git
- Komprimieren von Dateien im Mac OS X Finder oder Windows-Explorer
- Erstellen eines Quell-Bundles für eine .NET-Anwendung
- Testen des Quell-Bundles

Erstellen eines Quell-Bundles mit der Befehlszeile

Erstellen Sie ein Quell-Bundle mit dem Befehl `zip`. Um ausgeblendete Dateien und Ordner einzuschließen, verwenden Sie ein Muster wie das Folgende.

```
~/myapp$ zip ../myapp.zip -r * .[^.]*
  adding: app.js (deflated 63%)
  adding: index.js (deflated 44%)
  adding: manual.js (deflated 64%)
  adding: package.json (deflated 40%)
  adding: restify.js (deflated 85%)
  adding: .ebextensions/ (stored 0%)
  adding: .ebextensions/xray.config (stored 0%)
```

So wird sichergestellt, dass Elastic Beanstalk-Konfigurationsdateien und andere Dateien und Ordner, die mit einem Punkt beginnen, im Archiv eingeschlossen werden.

Verwenden Sie für Tomcat-Webanwendungen `jar`, um ein Web-Archiv zu erstellen.

```
~/myapp$ jar -cvf myapp.war .
```

Die obigen Befehle schließen ausgeblendete Dateien ein, wodurch Ihr Quell-Bundle unnötig vergrößert werden kann. Verwenden Sie für eine bessere Kontrolle ein detaillierteres Dateimuster oder erstellen Sie Ihr Quell-Bundle mit Git.

Erstellen eines Quell-Bundles mit Git

Wenn Sie den Anwendungsquellcode mit Git verwalten, erstellen Sie das Quell-Bundle über den Befehl `git archive`.

```
$ git archive -v -o myapp.zip --format=zip HEAD
```

`git archive` schließt nur Dateien ein, die in git gespeichert sind, und schließt ignorierte Dateien und git Dateien aus. Dies trägt dazu bei, dass Ihr Quell-Bundle so klein wie möglich ist. Weitere Informationen finden Sie unter git-archive manual page.

Komprimieren von Dateien im Mac OS X Finder oder Windows-Explorer

Achten Sie beim Erstellen einer `ZIP`-Datei im Mac OS X Finder oder Windows-Explorer darauf, die Dateien und Unterordner selbst zu komprimieren (und nicht den übergeordneten Ordner).

> **Note**
>
> In der grafischen Benutzeroberfläche (GUI) von Mac OS X- und Linux-basierten Betriebssystemen werden keine Dateien und Ordner angezeigt, deren Namen mit einem Punkt (.) beginnen. Soll die `ZIP`-Datei einen versteckten Ordner wie z. B. `.ebextensions` enthalten, komprimieren Sie die Anwendung mit der Befehlszeile (anstelle der GUI). Informationen zu Befehlszeilenverfahren zum Erstellen einer `ZIP`-Datei auf einem Mac OS X- oder Linux-basierten Betriebssystem finden Sie unter Erstellen eines Quell-Bundles mit der Befehlszeile.

Example

Angenommen, Sie haben einen Python-Projektordner mit dem Namen `myapp` und folgenden Dateien und Unterordnern:

```
myapplication.py
README.md
static/
static/css
static/css/styles.css
static/img
static/img/favicon.ico
static/img/logo.png
templates/
templates/base.html
templates/index.html
```

Wie bereits oben in der Anforderungsliste dargelegt, muss das Quell-Bundle ohne übergeordneten Ordner komprimiert werden, sodass in der extrahierten Struktur kein zusätzliches Top-Level-Verzeichnis vorhanden ist. In diesem Beispiel soll bei der Dateiextraktion kein `myapp`-Ordner erstellt werden (bzw. in der Befehlszeile soll kein `myapp`-Segment zu den Dateipfaden hinzugefügt werden).

Dieses Dateistrukturbeispiel wird im ganzen Thema verwendet und veranschaulicht das Komprimieren von Dateien.

Erstellen eines Quell-Bundles für eine .NET-Anwendung

Wenn Sie Visual Studio verwenden, können Sie das in der enthaltene Bereitstellungstool verwenden, AWS Toolkit for Visual Studio um Ihre .NET-Anwendung auf Elastic Beanstalk bereitzustellen.

Weitere Informationen finden Sie unter Bereitstellung von Elastic Beanstalk Beanstalk-Anwendungen in.NET mithilfe von AWS Bereitstellungstools.

Falls das Quell-Bundle für die .NET-Anwendung manuell erstellt werden muss, können Sie nicht einfach eine `ZIP`-Datei mit enthaltenem Projektverzeichnis generieren. Dazu müssen Sie ein Webbereitstellungspaket für das Projekt erstellen, das für die Bereitstellung in Elastic Beanstalk geeignet ist. Es gibt mehrere Möglichkeiten, um ein Bereitstellungspaket zu erstellen:

- Erstellen Sie das Bereitstellungspaket mithilfe des Visual Studio-Assistenten Web veröffentlichen. Weitere Informationen finden Sie unter Gewusst wie: Erstellen eines Webbereitstellungspakets in Visual Studio.

  > **Important**
  >
  > Beim Erstellen eines Webbereitstellungspakets muss der Site name (Site-Name) mit `Default Web Site` beginnen.

- Falls Sie ein .NET-Projekt haben, können Sie das Bereitstellungspaket wie in folgendem Beispiel dargestellt mit dem Befehl msbuild erstellen.

  > **Important**
  >
  > Der Parameter `DeployIisAppPath` muss mit `Default Web Site` beginnen.

  ```
  C:/> msbuild <web_app>.csproj /t:Package /p:DeployIisAppPath="Default Web Site"
  ```

- Bei einem Website-Projekt können Sie das IIS Web Deploy-Tool zum Erstellen des Bereitstellungspakets nutzen. Weitere Informationen finden Sie unter Packaging and Restoring a Web site.

  > **Important**
  >
  > Der Parameter `apphostconfig` muss mit `Default Web Site` beginnen.

Sofern Sie mehrere Anwendungen oder eine ASP.NET Core-Anwendung bereitstellen, platzieren Sie den Ordner `.ebextensions` im Stamm des Quell-Bundles, parallel zu den Anwendungs-Bundles und der Manifestdatei:

```
~/workspace/source-bundle/
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- AspNetCore101HelloWorld.zip
|-- AspNetCoreHelloWorld.zip
|-- aws-windows-deployment-manifest.json
`-- VS2015AspNetWebApiApp.zip
```

Testen des Quell-Bundles

Möglicherweise möchten Sie das Quell-Bundle lokal testen, bevor Sie es in Elastic Beanstalk hochladen. Da die Dateien in Elastic Beanstalk im Grunde über die Befehlszeile extrahiert werden, ist es am besten, wenn Sie die Tests mit der Befehlszeile (anstatt mit einem GUI-Tool) ausführen.

Stellen Sie sicher, dass die Dateien in denselben Ordner wie das Archiv selbst extrahiert werden und nicht in einen neuen Top-Level-Ordner bzw. -Verzeichnis.

Verwenden der EB-CLI mit AWS CodeBuild

AWS CodeBuild kompiliert den Quellcode, führt Komponententests aus und erzeugt Artefakte, die bereitgestellt werden können. Sie können CodeBuild zusammen mit der EB CLI verwenden, um die Erstellung Ihrer Anwendung aus ihrem Quellcode zu automatisieren. Die Erstellung der Umgebung und alle anschließenden Bereitstellungen beginnen mit einem Erstellungsschritt, nach dem die so erstellte Anwendung bereitgestellt wird.

> **Note**
>
> Einige Regionen bieten dies nicht an CodeBuild. Die Integration zwischen Elastic Beanstalk und CodeBuild funktioniert in diesen Regionen nicht.
> Informationen zu den in den einzelnen Regionen angebotenen AWS Diensten finden Sie in der Regionentabelle.

Erstellen einer Anwendung

Um eine Elastic Beanstalk Beanstalk-Anwendung zu erstellen, die CodeBuild

1. Fügen CodeBuild Sie eine Build-Spezifikationsdatei, `buildspec.yml`, in Ihren Anwendungsordner ein.
2. Fügen Sie der Datei einen `eb_codebuild_settings`-Eintrag mit spezifischen Optionen für Elastic Beanstalk hinzu.
3. Führen Sie eb init im Ordner aus.

> **Note**
>
> Verwenden Sie keine Punkte (`.`) oder Leerzeichen ( ) im Anwendungsnamen, wenn Sie die EB-CLI mit verwenden CodeBuild.

Elastic Beanstalk erweitert das CodeBuild Build-Spezifikationsdateiformat um die folgenden zusätzlichen Einstellungen:

```
eb_codebuild_settings:
  CodeBuildServiceRole: role-name
  ComputeType: size
  Image: image
  Timeout: minutes
```

`CodeBuildServiceRole`

Der ARN oder Name der AWS Identity and Access Management (IAM) -Servicerolle, die verwendet CodeBuild werden kann, um in Ihrem Namen mit abhängigen AWS Diensten zu interagieren. Dieser Wert ist erforderlich. Wenn Sie diesen auslassen, schlagen alle folgenden eb create- oder eb deploy-Befehle fehl.

Weitere Informationen zum Erstellen einer Servicerolle für CodeBuild finden Sie unter Erstellen einer CodeBuild Servicerolle im AWS CodeBuild Benutzerhandbuch.

> **Note**
>
> Sie benötigen außerdem Berechtigungen, um eigenständig Aktionen ausführen zu können. CodeBuild Die Richtlinie für verwaltete Benutzer von Elastic

AWSElasticBeanstalk AdministratorAccess — Beanstalk umfasst alle erforderlichen Aktionsberechtigungen. CodeBuild Wenn Sie die verwaltete Richtlinie nicht verwenden, müssen Sie in Ihrer Benutzerrichtlinie die folgenden Berechtigungen erteilen:

```
"codebuild:CreateProject",
"codebuild:DeleteProject",
"codebuild:BatchGetBuilds",
"codebuild:StartBuild"
```

Details hierzu finden Sie unter Elastic Beanstalk-Benutzerrichtlinien verwalten.

`ComputeType`

Die Menge der Ressourcen, die vom Docker-Container in der Build-Umgebung verwendet werden. CodeBuild Gültige Werte sind BUILD_ GENERAL1 _SMALL, BUILD_ _MEDIUM und BUILD_ GENERAL1 _LARGE. GENERAL1

`Image`

Der Name des Docker Hub- oder Amazon ECR-Images, das für die Build-Umgebung CodeBuild verwendet wird. Dieses Docker-Image sollte alle Tools und Laufzeitbibliotheken enthalten, die zum Erstellen Ihres Codes erforderlich sind, und sollte der Zielplattform Ihrer Anwendung entsprechen. CodeBuild verwaltet und verwaltet eine Reihe von Images, die speziell für die Verwendung mit Elastic Beanstalk vorgesehen sind. Es wird empfohlen, eines dieser Images zu verwenden. Einzelheiten finden Sie CodeBuild im Benutzerhandbuch unter Docker-Images bereitgestellt von.AWS CodeBuild

Der `Image`-Wert ist optional. Wenn Sie diesen nicht angeben, wählt der Befehl eb init das am besten mit Ihrer Zielplattform übereinstimmende Image aus. Wenn Sie eb init im interaktiven Modus ausführen und der Befehl kein Image auswählen kann, werden Sie aufgefordert, ein Image auszuwählen. Am Ende einer erfolgreichen Initialisierung schreibt eb init das ausgewählte Image in die Datei `buildspec.yml`.

`Timeout`

Die Dauer in Minuten, während der der CodeBuild Build ausgeführt wird, bevor das Timeout überschritten wird. Dieser Wert ist optional. Einzelheiten zu gültigen Werten und Standardwerten finden Sie unter Erstellen eines Build-Projekts in CodeBuild.

> **Note**
>
> Dieses Timeout steuert die maximale Dauer eines CodeBuild Laufs, und die EB-CLI berücksichtigt es auch als Teil des ersten Schritts, eine Anwendungsversion zu erstellen. Es handelt sich nicht um den Wert, den Sie mittels der Option `--timeout` der Befehle eb create oder eb deploy angeben können. Dieser Wert steuert die Zeit, die die EB CLI auf die Erstellung oder Aktualisierung einer Umgebung maximal warten soll.

Erstellen und Bereitstellen Ihres Anwendungscodes

Wann immer Ihr Anwendungscode bereitgestellt werden muss, führt die EB-CLI einen Build aus und stellt dann die resultierenden Build-Artefakte in Ihrer Umgebung bereit. CodeBuild Dies ist bei Erstellung einer Elastic Beanstalk-Umgebung für Ihre Anwendung über den Befehl eb create und bei jeder späteren Bereitstellung von Codeänderungen in der Umgebung über den Befehl eb deploy der Fall.

Wenn der CodeBuild Schritt fehlschlägt, wird die Erstellung oder Bereitstellung der Umgebung nicht gestartet.

Markieren von Anwendungen

In diesem Thema werden die Vorteile des Taggens Ihrer Elastic Beanstalk Beanstalk-Anwendungen erläutert. Es enthält auch Anweisungen zum Erstellen und Verwalten von Anwendungs-Tags. Tags sind Schlüssel-Wert-Paare, die Ressourcen zugeordnet AWS sind. Weitere Informationen zum Elastic Beanstalk-Ressourcen-Tagging, zu Anwendungsfälle, Einschränkungen für Tag-Schlüssel und -Werte sowie zu unterstützten Ressourcentypen finden Sie unter Markieren von Elastic Beanstalk-Anwendungsressourcen.

Sie können Tags angeben, wenn Sie eine Anwendung erstellen. In einer vorhandenen Anwendung können Sie Tags hinzufügen oder entfernen sowie die Werte von vorhandenen Tags aktualisieren. Sie können jeder Anwendung bis zu 50 Tags hinzufügen.

Hinzufügen von Tags während der Anwendungserstellung

Wenn Sie die Elastic Beanstalk-Konsole zum Erstellen einer Anwendung nutzen, können Sie Tag-Schlüssel und -Werte im Dialogfeld Create New Application (Neue Anwendung erstellen) angeben.

Wenn Sie mit der EB CLI eine Anwendung erstellen, verwenden Sie die `--tags`-Option mit eb init, um Tags hinzuzufügen.

```
~/workspace/my-app$ eb init --tags mytag1=value1,mytag2=value2
```

Fügen Sie mit dem AWS CLI oder anderen API-basierten Clients Tags hinzu, indem Sie den `--tags` Parameter im Befehl verwenden. create-application

```
$ aws elasticbeanstalk create-application \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
      --application-name my-app --version-label v1
```

Verwalten der Tags einer vorhandenen Anwendung

Sie können Tags in einer vorhandenen Elastic Beanstalk-Anwendung hinzufügen, aktualisieren und löschen.

So verwalten Sie die Tags einer Anwendung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Manage Tags (Tags verwalten).
4. Verwenden Sie das Formular auf dem Bildschirm, um Tags hinzuzufügen, zu aktualisieren oder zu löschen.
5. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Wenn Sie Umgebungen mit der EB CLI aktualisieren, verwenden Sie zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags den Befehl eb tags.

Beispiel: Der folgende Befehl listet die Tags in einer Anwendung auf.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Mit dem folgenden Befehl wird das Tag `mytag1` aktualisiert und das Tag `mytag2` gelöscht.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter `eb tags`.

Verwenden Sie bei den AWS CLI oder anderen API-basierten Clients den list-tags-for-resource Befehl, um die Tags einer Anwendung aufzulisten.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Verwenden Sie den Befehl update-tags-for-resource zum Hinzufügen, Aktualisieren und Löschen von Tags in einer Anwendung.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-
app"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter `--tags-to-add` des Befehls update-tags-for-resource an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

> **Note**
>
> Um einige der EB-CLI und AWS CLI Befehle mit einer Elastic Beanstalk Beanstalk-Anwendung verwenden zu können, benötigen Sie den ARN der Anwendung. Verwenden Sie den folgenden Befehl, um den ARN abzurufen.
>
> ```
> $ aws elasticbeanstalk describe-applications --application-names my-app
> ```

Markieren von Elastic Beanstalk-Anwendungsressourcen

In diesem Thema werden die Vorteile der Verwendung von Tags mit Ihren Elastic Beanstalk Beanstalk-Anwendungsressourcen sowie die damit verbundenen Einschränkungen erläutert. Außerdem wird erklärt, wie Sie Tags für Anwendungsressourcen erstellen und verwalten.

Sie können Tags auf Ressourcen Ihrer AWS Elastic Beanstalk Anwendungen anwenden. Tags sind Schlüssel-Wert-Paare, die Ressourcen zugeordnet AWS sind. Mit Tags können Sie Ressourcen kategorisieren. Sie sind besonders nützlich, wenn Sie viele Ressourcen als Teil mehrerer AWS Anwendungen verwalten.

Hier finden Sie einige Möglichkeiten zur Nutzung von Tags mit Elastic Beanstalk-Ressourcen:

- Bereitstellungsstufen – Identifizieren Sie Ressourcen in verschiedenen Stufen Ihrer Anwendung, wie z. B. Entwicklung, Beta und Produktion.
- Kostenzuweisung — Verwenden Sie Kostenzuordnungsberichte, um Ihre Nutzung von AWS Ressourcen im Zusammenhang mit verschiedenen Ausgabenkonten nachzuverfolgen. Die Berichte enthalten außerdem markierte und nicht markierte Ressourcen, aggregieren die Kosten jedoch gemäß den Tags. Weitere Informationen darüber, wie Kostenzuordnungsberichte Tags verwenden, finden Sie unter Verwenden von Kostenzuordnungs-Tags für benutzerdefinierte Fakturierungsberichte im AWS -Benutzerhandbuch für Fakturierungs- und Kostenverwaltung.
- Zugriffskontrolle – Verwenden Sie Tags zur Verwaltung der Berechtigungen für Anfragen und Ressourcen. Beispiel: Ein Benutzer, der nur Beta-Umgebungen erstellen und verwalten kann, sollte nur Zugriff auf Beta-Ressourcen haben. Details hierzu finden Sie unter Tags zur Steuerung des Zugriffs auf Elastic Beanstalk-Ressourcen verwenden.

Sie können bis zu 50 Tags für jede -Ressource hinzufügen. Umgebungen unterscheiden sich geringfügig: Elastic Beanstalk fügt Umgebungen drei Standard-System-Tags hinzu. Diese Tags können Sie weder bearbeiten noch löschen. Neben den Standard-Tags können Sie jeder Umgebung bis zu 47 zusätzliche Tags hinzufügen.

Die folgenden Einschränkungen gelten für Tag-Schlüssel und -Werte:

- Schlüssel und Werte dürfen Buchstaben, Zahlen, Leerzeichen sowie die folgenden Symbole enthalten: `_ . : / = + - @`
- Schlüssel können bis zu 127 Zeichen enthalten. Werte können bis zu 255 Zeichen enthalten.

> **Note**
>
> Diese Längenbeschränkungen gelten für Unicode-Zeichen in UTF-8. Für andere Multibyte-Kodierungen können niedrigere Limits gelten.

- Bei Schlüsseln muss die Groß- und Kleinschreibung beachtet werden.

- Schlüssel dürfen nicht mit `aws:` oder `elasticbeanstalk:` beginnen.

Ressourcen, die markiert werden können

Im Folgenden werden die Arten von Elastic Beanstalk-Ressourcen, die Sie markieren können, sowie Links zu bestimmten Themen zum Verwalten von Tags für die einzelnen Ressourcen aufgeführt:

- Anwendungen
- Umgebungen
- Anwendungsversionen
- Gespeicherte Konfigurationen

Tag-Propagierung zu Startvorlagen

Elastic Beanstalk bietet eine Option, um die Propagierung von Umgebungs-Tags zu Startvorlagen zu ermöglichen. Diese Option bietet eine kontinuierliche Unterstützung für die Tag-basierte Zugriffskontrolle mit Startvorlagen.

> **Note**
>
> Startkonfigurationen werden nach und nach eingestellt und durch Startvorlagen ersetzt. Weitere Informationen finden Sie unter Startkonfigurationen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Um Ausfallzeiten laufender EC2 Instances zu verhindern, werden Tags CloudFormation nicht an bestehende Startvorlagen weitergegeben. Bei Anwendungsfällen, die Tags für die Ressourcen Ihrer Umgebung erfordern, können Sie Elastic Beanstalk das Erstellen von Startvorlagen mit Tags für diese Ressourcen ermöglichen. Stellen Sie dazu die Option `LaunchTemplateTagPropagationEnabled` im Namespace aws:autoscaling:launchconfiguration auf `true` ein. Der Standardwert ist `false`.

Die folgende beispielhafte Konfigurationsdatei ermöglicht die Propagierung von Tags zu Startvorlagen.

```
option_settings:
  aws:autoscaling:launchconfiguration:
```

```
    LaunchTemplateTagPropagationEnabled: true
```

Elastic Beanstalk kann Tags nur für die folgenden Ressourcen an Startvorlagen propagieren:

- EBS-Datenträger
- EC2 Instanzen
- EC2 Netzwerkschnittstellen
- CloudFormation Vorlagen starten, die eine Ressource definieren

Diese Einschränkung besteht darin, dass Tags bei der Vorlagenerstellung CloudFormation nur für bestimmte Ressourcen zulässig sind. Weitere Informationen finden Sie TagSpecificationim AWS CloudFormation Benutzerhandbuch.

> **Important**
>
> - Wenn Sie diesen Optionswert für eine bestehende Umgebung von `false` in `true` ändern, kann dies eine gravierende Änderung für zuvor vorhandene Tags herbeiführen.
> - Wenn diese Funktion aktiviert ist, muss die Weitergabe von Tags EC2 ersetzt werden, was zu Ausfallzeiten führen kann. Sie können fortlaufende Updates aktivieren, um Konfigurationsänderungen stapelweise anzuwenden und Ausfallzeiten während der Aktualisierung zu vermeiden. Weitere Informationen finden Sie unter Konfigurationsänderungen.

Weitere Informationen über Startvorlagen finden Sie in den folgenden Abschnitten:

- Vorlagen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch starten
- Arbeiten mit Vorlagen im Benutzerhandbuch von AWS CloudFormation
- Elastic-Beanstalk-Vorlagenausschnitte im Benutzerhandbuch von AWS CloudFormation

Elastic Beanstalk Beanstalk-Umgebungen verwalten

In diesem Kapitel wird beschrieben, wie Sie Ihre Elastic Beanstalk Beanstalk-Umgebungen erstellen und verwalten. Diese Einführungsseite bietet einen Überblick über Updates, Wartungen und Konfigurationen, die Sie im Laufe der Zeit anwenden werden, wenn sich Ihre Anwendung und Umgebung weiterentwickeln.

Funktionen der Umgebung

Sie können eigene Umgebungen für Entwicklung, Tests und Produktion erstellen und verwalten und jede Version der Anwendung in jeder Umgebung bereitstellen. Umgebungen können lange oder temporär eingesetzt werden. Wenn Sie eine Umgebung beenden, können Sie deren Konfiguration für eine spätere erneute Erstellung speichern.

Bereitstellungen von Anwendungen

Im Entwicklungsverlauf der Anwendung wird diese häufig – und zu verschiedenen Zwecken in mehreren unterschiedlichen Umgebungen – bereitgestellt. Mit Elastic Beanstalk können Sie konfigurieren, wie Bereitstellungen durchgeführt werden. Sie können die Bereitstellung für alle Instances der Umgebung gleichzeitig ausführen oder sie in Stapel für eine fortlaufende Bereitstellung unterteilen.

Konfigurationsänderungen

Konfigurationsänderungen werden getrennt von Bereitstellungen verarbeitet und haben einen eigenen Umfang. Wenn Sie beispielsweise den Typ der EC2-Instances ändern, auf denen die Anwendung ausgeführt wird, müssen alle Instances ersetzt werden. Ändern Sie hingegen die Konfiguration vom Load Balancer der Umgebung, kann diese Änderung direkt ohne Serviceunterbrechung oder Kapazitätseinbußen erfolgen. Sie können auch Konfigurationsänderungen in fortlaufenden Konfigurationsaktualisierungen ausführen, bei denen die Instances der Umgebung in Stapeln geändert werden.

> **Note**
>
> Ändern Sie die Ressourcen der Umgebung nur mit Elastic Beanstalk. Wenn Sie Ressourcen mithilfe der Konsole, der CLI-Befehle oder eines anderen Dienstes ändern SDKs, kann Elastic Beanstalk den Status dieser Ressourcen nicht genau überwachen, und Sie können die Konfiguration nicht speichern oder die Umgebung nicht zuverlässig neu erstellen. Out-

of-Band-Änderungen können zudem zu Problemen beim Aktualisieren oder Beenden einer Umgebung führen.

Plattformaktualisierungen

Beim Erstellen einer Umgebung wählen Sie eine Plattformversion aus. Wir aktualisieren die Plattformen regelmäßig mit neuen Plattformversionen und stellen damit Leistungsverbesserungen und neue Funktionen bereit. Sie können Ihre Umgebung jederzeit auf die neueste Plattformversion aktualisieren. Im AWS Elastic Beanstalk Plattformhandbuch finden Sie eine Liste der unterstützten Plattformen und einen Versionsverlauf der Plattform, der die Datumsbereiche enthält, in denen sie aktuell waren.

Architektur-Optionen

Mit zunehmender Komplexität der Anwendung können Sie diese in mehrere Komponenten aufteilen, von denen jede in einer eigenen Umgebung ausgeführt wird. Für lange laufende Workloads können Sie Worker-Umgebungen starten, die Aufträge von einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange verarbeiten.

Themen

- Elastic Beanstalk-Umgebungsverwaltungskonsole verwenden
- Erstellen einer Elastic Beanstalk-Umgebung
- Verwalten mehrerer Elastic Beanstalk-Umgebungen als Gruppe mit der EB CLI
- Anwendungen in Elastic Beanstalk-Umgebungen bereitstellen
- Konfigurationsänderungen
- Aktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung
- Abbrechen von Umgebungskonfigurations-Updates und Anwendungsbereitstellungen
- Elastic Beanstalk-Umgebungen wiederherstellen
- Typen von Umgebungen
- Elastic-Beanstalk-Worker-Umgebungen
- Verknüpfungen zwischen Elastic Beanstalk-Umgebungen erstellen
- Wiederherstellung Ihrer Elastic Beanstalk Beanstalk-Umgebung aus einem ungültigen Zustand

Elastic Beanstalk-Umgebungsverwaltungskonsole verwenden

In diesem Abschnitt wird beschrieben, wie Sie Ihre Elastic Beanstalk Beanstalk-Umgebung mithilfe der Environment Management Console verwalten können. Die Konsole bietet die Möglichkeit, die Konfiguration Ihrer Umgebung zu verwalten und allgemeine Aktionen auszuführen. Zu diesen Aktionen gehören der Neustart der in Ihrer Umgebung laufenden Webserver, das Klonen Ihrer Umgebung oder der Neuaufbau Ihrer Umgebung von Grund auf.

Themen

- Zugriff auf die Umgebungsmanagementkonsole
- Bereich „Übersicht über die Umgebung“
- Einzelheiten zur Umgebung
- Umgebungsaktionen

Zugriff auf die Umgebungsmanagementkonsole

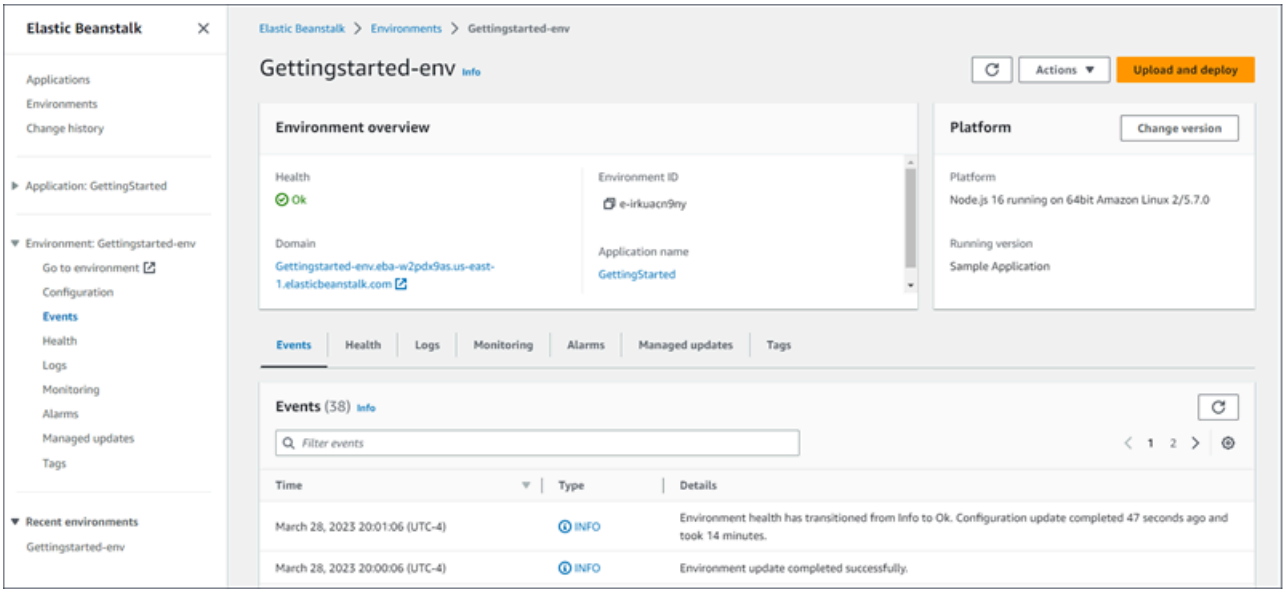
Das folgende Verfahren enthält Schritte zum Starten der Environment Management Console.

Wenn Sie bereits bei der Elastic Beanstalk Beanstalk-Konsole angemeldet sind, können Sie die Umgebungsverwaltungsseite auch über den aufrufen. Konsole zur Anwendungsverwaltung Wählen Sie eine Umgebung aus der Liste aus, um die Details der Managementkonsole für die ausgewählte Umgebung anzuzeigen.

Verwenden der Environment Management Console

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

Die folgende Abbildung zeigt die Environment Management Console.



Der obere Bereich ist die Seite mit der Umgebungsübersicht. Es enthält Informationen auf oberster Ebene über Ihre Umgebung.

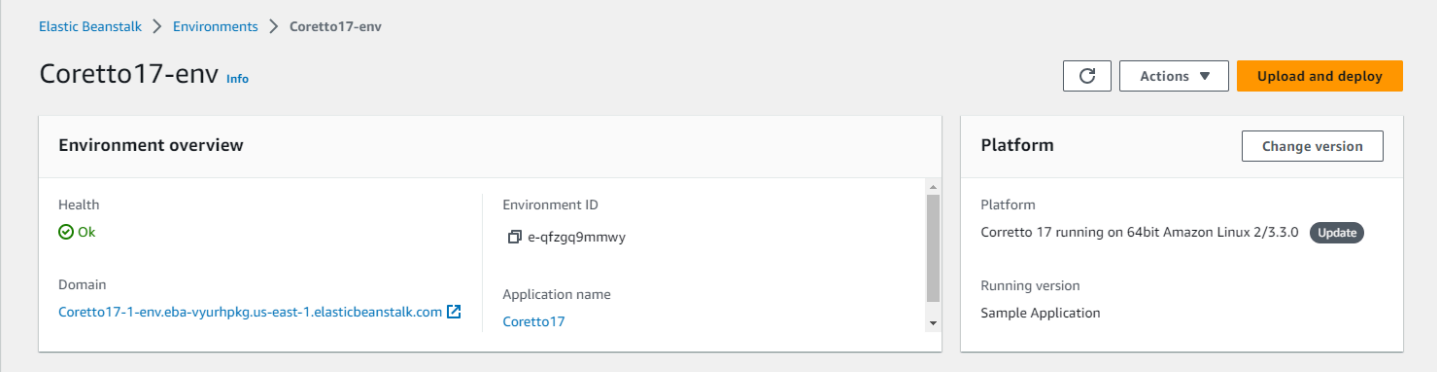
In der unteren Hälfte der Seite werden Registerkarten mit detaillierteren Informationen angezeigt. Die Registerkarte Ereignisse wird standardmäßig angezeigt. Die Seiten, die mit den Registerkarten verknüpft sind, sind auch im linken Navigationsbereich unter der Umgebung aufgeführt.

Im Navigationsbereich der Konsole wird der Name der Anwendung angezeigt, die in der Umgebung bereitgestellt wurde, sowie die zugehörigen Anwendungsverwaltungsseiten. Der Umgebungsname wird auch auf der Navigationsseite angezeigt, gefolgt von den Seiten zur Umgebungsverwaltung. Die unter dem Umgebungsnamen aufgeführten Links beinhalten zusätzlich zu den bereits erwähnten Registerseiten auch „Gehe zu Umgebung“ und „Konfiguration“.

Bereich „Übersicht über die Umgebung“

In diesem Thema werden die Informationen beschrieben, die im Bereich Umgebungsübersicht bereitgestellt werden. Er enthält Informationen auf oberster Ebene über Ihre Umgebung und befindet sich in der oberen Hälfte der Umgebungsmanagementkonsole.

Die folgende Abbildung zeigt den Bereich mit der Umgebungsübersicht.



Gesundheit

Der Gesamtzustand der Umgebung. Wenn sich der Zustand Ihrer Umgebung verschlechtert, wird der Link Ursachen anzeigen neben dem Zustand der Umgebung angezeigt. Wählen Sie diesen Link, um die Registerkarte Zustand mit weiteren Details anzuzeigen.

Domain

Die Domäne oder URL der Umgebung befindet sich im oberen Teil der Umgebungsübersichtsseite, unter Health der Umgebung. Dies ist die URL der Webanwendung, die die Umgebung ausführt. Sie können die Anwendung starten, indem Sie die URL auswählen.

Umgebungs-ID

Die Umgebungs-ID. Dies ist eine interne ID, die bei der Erstellung der Umgebung generiert wird.

Anwendungsname

Der Name der in Ihrer Umgebung bereitgestellten und ausgeführten Anwendung.

Ausgeführte Version

Der Name der in Ihrer Umgebung bereitgestellten und ausgeführten Anwendungsversion. Wählen Sie Upload and Deploy (Hochladen und bereitstellen) aus, um ein Quell-Bundle hochzuladen und es in Ihrer Umgebung bereitzustellen. Diese Option erstellt eine neue Anwendungsversion.

Plattform

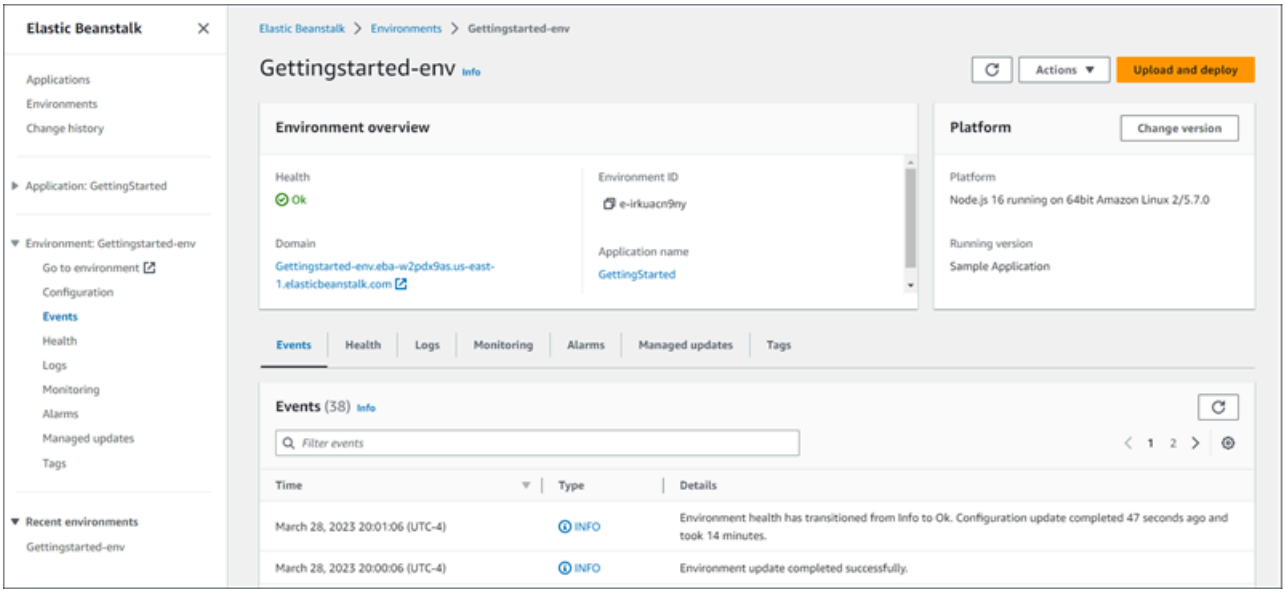
Der Name der in Ihrer Umgebung ausgeführten Plattformversion. In der Regel umfasst dies die Architektur, das Betriebssystem (OS), die Sprache und den Anwendungsserver (zusammen als Plattformzweig bekannt) mit einer bestimmten Plattformversionsnummer.

Wenn es sich bei Ihrer Plattformversion nicht um die neueste verfügbare Version handelt, wird neben der Plattformversion im Abschnitt Plattform ein Statuslabel angezeigt. Das Label Update weist darauf hin, dass die Plattformversion zwar unterstützt wird, aber eine neuere Version verfügbar ist. Die Plattformversion kann auch als Veraltet oder außer Betrieb genommen bezeichnet werden. Wählen Sie Version ändern, um Ihren Plattformzweig auf eine neuere Version zu aktualisieren. Weitere Informationen zu den Zuständen einer Plattformversion finden Sie im Abschnitt Plattformzweig in der Elastic Beanstalk-Plattformen – Glossar. Das vorherige Bild auf dieser Seite zeigt die Bezeichnung Aktualisierungsstatus für die angegebene Plattform.

Einzelheiten zur Umgebung

In diesem Thema werden die zusätzlichen Informationen beschrieben, die die Environment Management Console im linken Navigationsbereich und auf den Seiten mit Registerkarten bereitstellt.

Die folgende Abbildung zeigt die Environment Management Console.



In der unteren Hälfte der Environment Management Console sind Registerkarten mit detaillierteren und vielfältigeren Informationen über die Umgebung aufgeführt. Sie können entweder die Registerkarte oder die Seitenbezeichnung im linken Navigationsbereich auswählen.

Im linken Navigationsbereich der Konsole gibt es unter dem Umgebungsnamen zwei Optionen, die nicht auf den Seiten mit Registern angezeigt werden. Diese lauten „Gehe zur Umgebung“ und „Konfiguration“.

> **Note**
>
> Wählen Sie Gehe zur Umgebung, um Ihre Anwendung zu starten.

Konfiguration

Verwenden Sie die Konfigurationsseite im linken Navigationsbereich, um die aktuellen Konfigurationseinstellungen für Ihre Umgebung und deren Ressourcen anzuzeigen und zu aktualisieren. Dazu gehören Netzwerkkonfiguration, Datenbankkonfiguration, Lastenausgleich, Benachrichtigungen, Einstellungen für die Systemüberwachung, Konfiguration für verwaltete Plattformupdates, Bereitstellungskonfiguration, Instanzprotokoll-Streaming AWS X-Ray, CloudWatch Integration, Proxyserver-Einstellungen, Umgebungseigenschaften und plattformspezifische Optionen. Verwenden Sie die Einstellungen auf dieser Seite, um das Verhalten Ihrer Umgebung während Bereitstellungen anzupassen, zusätzliche Funktionen zu aktivieren und den Instance-Typ und andere Einstellungen zu ändern, die Sie während der Umgebungserstellung wählen.

Weitere Informationen finden Sie unter Konfigurieren von Elastic-Beanstalk-Umgebungen.

Ereignisse

Die Seite Events (Ereignisse) zeigt den Event-Stream für Ihre Umgebung. Elastic Beanstalk gibt Ereignismeldungen aus, wenn Sie mit der Umgebung interagieren und wenn dadurch Ressourcen Ihrer Umgebung erstellt oder geändert werden.

Weitere Informationen finden Sie unter Ereignis-Stream einer Elastic Beanstalk-Umgebung anzeigen.

Gesundheit

Wenn die erweiterte Zustandsüberwachung aktiviert ist, werden auf dieser Seite die EC2 Instances in Ihrer Umgebung sowie die Live-Gesundheitsinformationen für jede Instance aufgeführt.

Auf der Seite „Gesamtzustand“ werden die Gesundheitsdaten als Durchschnittswerte für alle Instanzen Ihrer Umgebung zusammen angezeigt.

Im Bereich „Erweiterter Instanzstatus“ werden Live-Integritätsinformationen für jede einzelne EC2 Instanz in Ihrer Umgebung angezeigt. Die erweiterte Zustandsüberwachung ermöglicht Elastic Beanstalk, die Ressourcen in Ihrer Umgebung genau zu überwachen, sodass der Zustand Ihrer Anwendung genauer bewertet werden kann.

Wenn die erweiterte Zustandsüberwachung aktiviert ist, zeigt diese Seite Informationen über die Anfragen, die von den Instances in Ihrer Umgebung verarbeitet werden, und Metriken aus dem Betriebssystem, wie z. B. Latenz, Last und CPU-Auslastung.

Weitere Informationen finden Sie unter Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk.

Protokolle

Auf der Seite „Protokolle“ können Sie Protokolle von den EC2 Instances in Ihrer Umgebung abrufen. Wenn Sie Protokolle anfordern, sendet Elastic Beanstalk einen Befehl an die Instances, die dann Protokolle in Ihren Elastic Beanstalk Storage Bucket in Amazon S3 hochladen. Wenn Sie Protokolle auf dieser Seite anfordern, löscht Elastic Beanstalk diese nach 15 Minuten automatisch aus Amazon S3.

Sie können die Instances Ihrer Umgebung auch so konfigurieren, dass Protokolle zur dauerhaften Speicherung in Amazon S3 hochgeladen werden, nachdem sie lokal rotiert sind.

Weitere Informationen finden Sie unter Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen.

Überwachen

Die Seite Monitoring (Überwachung) zeigt eine Übersicht über die Zustandsdaten für Ihre Umgebung an. Dazu gehören die von Elastic Load Balancing und Amazon bereitgestellten Standardmetriken sowie Grafiken EC2, die zeigen, wie sich der Zustand der Umgebung im Laufe der Zeit verändert hat.

Weitere Informationen finden Sie unter Überwachung des Zustands der Umgebung in der AWS Managementkonsole.

Alarme

Auf der Seite Existing Alarms (Vorhandene Alarme) werden Informationen zu allen Alarmen angezeigt, die Sie für Ihre Umgebung konfiguriert haben. Sie können die Optionen auf dieser Seite verwenden, um Alarme zu erstellen oder zu löschen.

Weitere Informationen finden Sie unter Alarme verwalten.

Verwaltete Aktualisierungen

Auf der Seite Verwaltete Updates werden Informationen zu bevorstehenden und abgeschlossenen Updates der verwalteten Plattform und zum Austausch von Instances angezeigt.

Mit der Funktion für verwaltete Updates können Sie Ihre Umgebung so konfigurieren, dass sie während eines von Ihnen gewählten wöchentlichen Wartungsfensters automatisch auf die neueste Plattformversion aktualisiert wird. Zwischen den Plattformversionen können Sie festlegen, dass Ihre Umgebung während des Wartungsfensters alle EC2 Amazon-Instances ersetzt. Dadurch können Probleme vermieden werden, die auftreten, wenn Ihre Anwendung für längere Zeiträume ausgeführt wird.

Weitere Informationen finden Sie unter Verwaltete Plattformaktualisierungen.

Tags (Markierungen)

Auf der Seite Tags (Tags) werden die Tags angezeigt, die Elastic Beanstalk auf die Umgebung angewendet hat, als sie erstellt wurde, sowie alle von Ihnen hinzugefügten Tags. Sie können benutzerdefinierte Tags hinzufügen, bearbeiten und löschen. Sie können die Tags, die Elastic Beanstalk angewendet hat, nicht bearbeiten oder löschen.

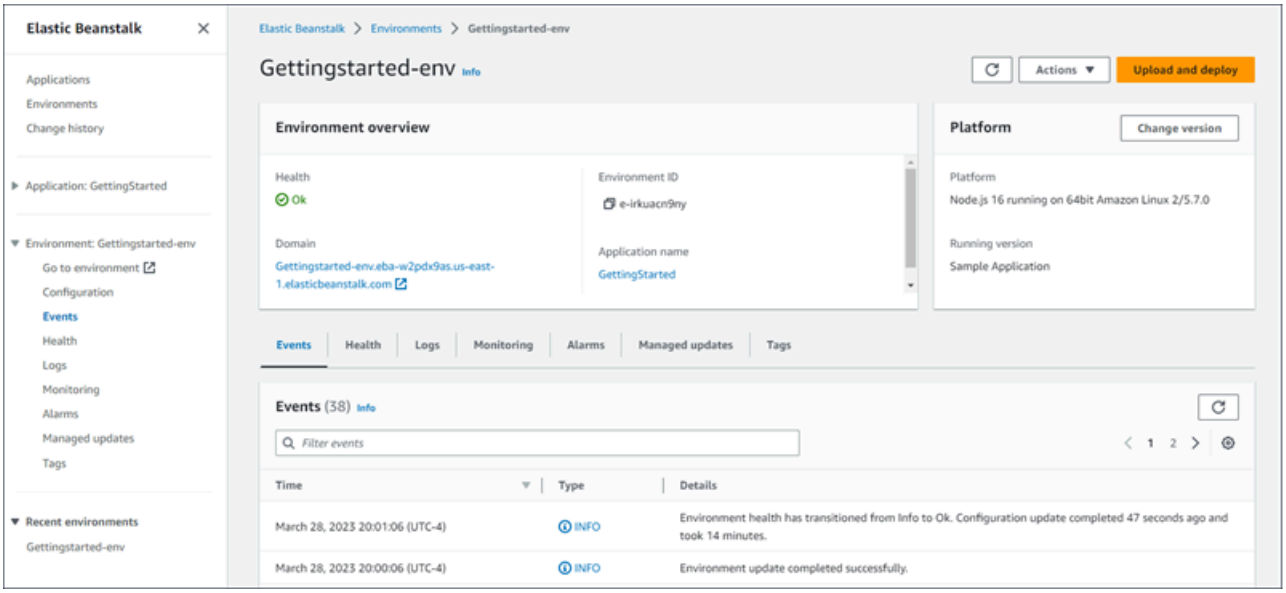
Umgebungs-Tags werden auf alle Ressourcen angewendet, die Elastic Beanstalk zur Unterstützung Ihrer Anwendung erstellt.

Weitere Informationen finden Sie unter Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren.

Umgebungsaktionen

In diesem Thema werden die allgemeinen Operationen beschrieben, die Sie im Drop-down-Menü Aktionen in der Umgebungsmanagement-Konsole auswählen können, um sie in Ihrer Umgebung auszuführen.

Die folgende Abbildung zeigt die Environment Management Console. Das Dropdownmenü Aktionen befindet sich auf der rechten Seite der Kopfzeile, in der der Umgebungsname angezeigt wird, neben der Schaltfläche Aktualisieren.



> **Note**
>
> Einige Aktionen sind nur unter bestimmten Bedingungen verfügbar und bleiben deaktiviert, bis die richtigen Bedingungen erfüllt sind.

Laden der Konfiguration

Laden Sie eine zuvor gespeicherte Konfiguration. Konfigurationen werden in Ihrer Anwendung gespeichert und können von jeder zugehörigen Umgebung geladen werden. Wenn Sie Änderungen an der Konfiguration Ihrer Umgebung vorgenommen haben, können Sie eine gespeicherte Konfiguration laden, um diese Änderungen rückgängig zu machen. Sie können auch eine gespeicherte Konfiguration laden, die Sie aus einer anderen Umgebung mit derselben Anwendung gespeichert haben, um Konfigurationsänderungen zwischen ihnen zu propagieren.

Konfiguration speichern

Speichern Sie die aktuelle Konfiguration Ihrer Umgebung in Ihrer Anwendung. Bevor Sie Änderungen an der Konfiguration Ihrer Umgebung vornehmen, speichern Sie die aktuelle Konfiguration, damit Sie sie später bei Bedarf wiederherstellen können. Sie können auch eine gespeicherte Konfiguration anwenden, wenn Sie eine neue Umgebung starten.

Tauschen Sie die Umgebungsdomänen aus (URLs)

Tauschen Sie den CNAME der aktuellen Umgebung gegen eine neue Umgebung aus. Nach einem CNAME-Austausch wird der gesamte Datenverkehr zur Anwendung, die die Umgebungs-URL verwendet, an die neue Umgebung geleitet. Wenn Sie bereit sind, eine neue Version Ihrer Anwendung bereitzustellen, können Sie eine separate Umgebung unter der neuen Version starten. Wenn die neue Umgebung bereit ist, Anfragen anzunehmen, führen Sie einen CNAME-Austausch durch, um den Datenverkehr an die neue Umgebung weiterleiten zu können. Dadurch werden Ihre Services nicht unterbrochen. Weitere Informationen finden Sie unter Blau/Grün-Bereitstellungen mit Elastic Beanstalk.

Umgebung klonen

Starten Sie eine neue Umgebung mit der gleichen Konfiguration wie in Ihrer aktuell ausgeführten Umgebung.

Klonen mit der neuesten Plattform

Sie können Ihre aktuelle Umgebung mit der neuesten Version der jeweils verwendeten Elastic Beanstalk-Plattform klonen. Diese Option ist nur verfügbar, wenn eine neuere Version der aktuellen Umgebungsplattform zur Verfügung steht.

Abbrechen des aktuellen Vorgangs

Stoppen Sie ein laufendes Umgebungs-Update. Das Stoppen eines Vorgangs kann dazu führen, dass einige der Instances in Ihrer Umgebung einen anderen Zustand haben als andere – je nachdem, wie weit der Prozess fortgeschritten ist. Diese Option ist nur verfügbar, wenn Ihre Umgebung aktualisiert wird.

Neustarten von App-Servern

Starten Sie den Webserver, der auf den Instances Ihrer Umgebung ausgeführt wird, neu. Mit dieser Option werden keine AWS Ressourcen beendet oder neu gestartet. Wenn Ihre Umgebung merkwürdiges Verhalten als Reaktion auf einige fehlerhafte Anfragen zeigt, kann durch einen Neustart des Anwendungsservers die Funktionalität vorübergehend wiederhergestellt werden, während Sie der Ursache auf den Grund gehen.

Neuerstellen der Umgebung

Beenden Sie alle Ressourcen in der laufenden Umgebung und erstellen Sie eine neue Umgebung mit den gleichen Einstellungen bereit. Dieser Vorgang dauert mehrere Minuten, ähnlich der Zeit,

die für die Bereitstellung einer neuen Umgebung von Grund auf benötigt wird. Alle Amazon RDS-Instances, die in der Datenschicht Ihrer Umgebung ausgeführt werden, werden während einer Neuerstellung gelöscht. Wenn Sie die Daten benötigen, erstellen Sie einen Snapshot. Sie können einen Snapshot manuell in der RDS-Konsole erstellen oder die Löschrichtlinie Ihrer Datenschicht konfigurieren, um vor dem Löschen der Instance automatisch einen Snapshot zu erstellen. Dies ist die Standardeinstellung beim Erstellen einer Datenschicht.

Beenden der Umgebung

Beenden Sie alle Ressourcen in der laufenden Umgebung und entfernen Sie die Umgebung aus der Anwendung. Wenn Sie über eine RDS-Instance verfügen, die in einer Datenebene ausgeführt wird und deren Daten beibehalten werden müssen, stellen Sie sicher, dass die Richtlinie zum Löschen von Datenbanken entweder auf `Snapshot` oder `Retain` festgelegt ist. Weitere Informationen finden Sie unter Datenbank-Lebenszyklus im Kapitel Konfigurieren von Umgebungen in diesem Leitfaden.

Erstellen einer Elastic Beanstalk-Umgebung

Mit dem folgenden Verfahren wird eine neue Umgebung gestartet, in der die Standardanwendung ausgeführt wird. Diese Schritte sind vereinfacht, um Ihre Umgebung mit Standardoptionswerten schnell zum Laufen zu bringen.

> **Hinweis zu Berechtigungen**
>
> Das Erstellen einer Umgebung erfordert die Berechtigungen in der verwalteten Elastic Beanstalk-Richtlinie für vollständigen Zugriff. Details dazu finden Sie unter Elastic Beanstalk-Benutzerrichtlinie.

Um eine Umgebung mit einer Anwendung (Konsole) zu starten

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Applications (Anwendungen). Wählen Sie eine bestehende Anwendung in der Liste aus. Sie können sich auch dafür entscheiden, eine zu erstellen, indem Sie den Anweisungen unter folgenVerwalten von -Anwendungen.
3. Wählen Sie auf der Seite mit der Anwendungsübersicht die Option Umgebung erstellen aus.

Dadurch wird der Assistent Create environment (Umgebung erstellen) gestartet. Der Assistent enthält eine Reihe von Schritten, mit denen Sie eine neue Umgebung erstellen können.

4. Wählen Sie für die Umgebungsebene die Webserver-Umgebung oder die Umgebungsebene Worker-Umgebung aus. Nach der Erstellung lässt sich die Umgebungsebene nicht mehr ändern.

   > **Note**
   >
   > Die .NET-Plattform auf Windows Server unterstützt die Worker-Umgebungsebene nicht.

   Die Standardfelder für Anwendungsinformationen basieren auf der Anwendung, die Sie zuvor ausgewählt haben.

   In der Gruppe Umgebungsinformationen wird der Umgebungsname standardmäßig auf der Grundlage des Anwendungsnamens voreingestellt. Wenn Sie einen anderen Umgebungsnamen bevorzugen, können Sie einen anderen Wert in das Feld eingeben. Sie können optional einen Domainnamen eingeben. Andernfalls generiert Elastic Beanstalk automatisch einen Wert. Sie können optional auch eine Beschreibung der Umgebung eingeben.

5. Wählen Sie in Platform (Plattform) die Plattform und die Plattformbranche aus, die/der mit der Sprache Ihrer Anwendung übereinstimmt.

   > **Note**
   >
   > Elastic Beanstalk unterstützt mehrere Versionen für die meisten Plattformen, die aufgelistet sind. Standardmäßig wählt die Konsole die empfohlene Version für die gewählte Plattform und Plattformvariante. Wenn Ihre Anwendung eine andere Version erfordert, können Sie diese hier auswählen. Informationen zu unterstützten Plattformversionen finden Sie unter the section called “Unterstützte Plattformen”.

6. Beim Anwendungscode haben Sie mehrere Möglichkeiten, fortzufahren.

   - Um die Standard-Beispielanwendung ohne Angabe des Quellcodes zu starten, wählen Sie Beispielanwendung. Diese Aktion wählt die Einzelseitenanwendung aus, die Elastic Beanstalk für die zuvor ausgewählte Plattform bereitstellt.
   - Wenn Sie eine Beispielanwendung aus diesem Handbuch heruntergeladen haben oder Ihren eigenen Quellcode für eine Anwendung haben, gehen Sie wie folgt vor.

a. Wählen Sie Code hochladen aus.
b. Wählen Sie als Nächstes Lokale Datei und dann unter Anwendung hochladen die Option Datei auswählen aus.
c. Das Betriebssystem Ihres Client-Computers bietet Ihnen eine Oberfläche zur Auswahl der lokalen Datei, die Sie heruntergeladen haben. Wählen Sie die Quellpaketdatei aus und fahren Sie fort.

7. Ihre Wahl für Voreinstellungen hängt von Ihrem Verwendungszweck für die Umgebung ab.

   - Wenn Sie eine Beispielumgebung erstellen, um mehr über Elastic Beanstalk oder eine Entwicklungsumgebung zu erfahren, wählen Sie Single Instance (kostenloses Kontingent berechtigt).
   - Wenn Sie eine Produktionsumgebung oder eine Umgebung erstellen, um mehr über Load Balancing zu erfahren, wählen Sie eine der Optionen für hohe Verfügbarkeit.

8. Wählen Sie Weiter aus.

Um den Servicezugriff zu konfigurieren

Als Nächstes benötigen Sie zwei Rollen. Eine Service-Rolle ermöglicht es Elastic Beanstalk, Ihre EC2-Instances zu überwachen und die Plattform Ihrer Umgebung zu aktualisieren. Eine EC2-Instance-Profilrolle ermöglicht Aufgaben wie das Schreiben von Logs und die Interaktion mit anderen Services.

Um die Service-Rolle zu erstellen oder auszuwählen

1. Wenn Sie zuvor eine Servicerolle erstellt haben und eine vorhandene auswählen möchten, wählen Sie den Wert aus der Dropdownliste Servicerolle aus und überspringen Sie die restlichen Schritte, um eine Servicerolle zu erstellen.
2. Wenn für die Servicerolle keine Werte aufgeführt sind oder Sie eine neue erstellen möchten, fahren Sie mit den nächsten Schritten fort.
3. Wählen Sie für Servicerolle die Option Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS -Service aus.
5. Wählen Sie als Anwendungsfall Elastic Beanstalk — Environment.
6. Wählen Sie Weiter aus.
7. Vergewissern Sie sich, dass die Berechtigungsrichtlinien Folgendes beinhalten, und wählen Sie dann Weiter aus:

- `AWSElasticBeanstalkEnhancedHealth`
- `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

8. Wählen Sie Rolle erstellen aus.
9. Kehren Sie zur Registerkarte Dienstzugriff konfigurieren zurück, aktualisieren Sie die Liste und wählen Sie dann die neu erstellte Servicerolle aus.

Um ein EC2-Instanzprofil zu erstellen oder auszuwählen

1. Wenn Sie zuvor ein EC2-Instanzprofil erstellt haben und ein vorhandenes auswählen möchten, wählen Sie den Wert aus der Dropdownliste für das EC2-Instanzprofil aus und überspringen Sie die restlichen Schritte, um ein EC2-Instanzprofil zu erstellen.
2. Wenn keine Werte für das EC2-Instanzprofil aufgeführt sind oder Sie ein neues erstellen möchten, fahren Sie mit den nächsten Schritten fort.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS -Service aus.
5. Wählen Sie als Anwendungsfall Elastic Beanstalk — Compute.
6. Wählen Sie Weiter aus.
7. Vergewissern Sie sich, dass die Berechtigungsrichtlinien Folgendes beinhalten, und wählen Sie dann Weiter aus:

   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`

8. Wählen Sie Rolle erstellen aus.
9. Kehren Sie zur Registerkarte Dienstzugriff konfigurieren zurück, aktualisieren Sie die Liste und wählen Sie dann das neu erstellte EC2-Instanzprofil aus.

Um die Konfiguration und Erstellung Ihrer Anwendung abzuschließen

1. (Optional) Wenn Sie zuvor ein EC2-Schlüsselpaar erstellt haben, können Sie es aus der Dropdownliste für das EC2-Schlüsselpaarfeld auswählen. Sie würden es verwenden, um sich sicher bei der Amazon EC2-Instance anzumelden, die Elastic Beanstalk für Ihre Anwendung bereitstellt. Wenn Sie diesen Schritt überspringen, können Sie jederzeit ein EC2-Schlüsselpaar

erstellen und zuweisen, nachdem die Umgebung erstellt wurde. Weitere Informationen finden Sie unter EC2 key pair.

2. Wählen Sie auf der Seite Configure service access (Servicezugriff konfigurieren) die Option Skip to Review (Zur Überprüfung springen).
3. Auf der Seite Review (Überprüfung) wird eine Zusammenfassung all Ihrer Auswahlmöglichkeiten angezeigt.

   Um Ihre Umgebung weiter anzupassen, wählen Sie neben dem Schritt, der alle Elemente enthält, die Sie konfigurieren möchten, die Option Edit (Bearbeiten). Sie können die folgenden Optionen nur während des Erstellens der Umgebung festlegen:

   - Environment name
   - Domainname
   - Version der Plattform
   - Prozessor
   - Load balancer type (Load-Balancer-Typ)
   - Stufe

   Sie können die folgenden Einstellungen nach der Umgebungserstellung zwar ändern, jedoch müssen dafür neue Instances oder andere Ressourcen bereitgestellt werden, deren Anwendung viel Zeit in Anspruch nehmen kann:

   - Instance-Typ, Root-Volume, key pair und AWS Identity and Access Management (IAM-) Rolle
   - Interne Amazon RDS-Datenbank
   - VPC

   Weitere Informationen zu allen verfügbaren Einstellungen finden Sie unter Der Assistent zum Erstellen einer neuen Umgebung.
4. Wählen Sie unten auf der Seite Submit (Senden), um die Erstellung Ihrer neuen Umgebung zu initialisieren.

Während Elastic Beanstalk Ihre Umgebung erstellt, werden Sie zu Elastic Beanstalk-Konsole umgeleitet. Wenn der Umgebungsstatus grün angezeigt wird, wählen Sie die URL neben dem Umgebungsnamen, um die laufende Anwendung anzuzeigen. Diese URL ist in der Regel über das

Internet zugänglich, es sei denn, Sie konfigurieren Ihre Umgebung so, dass eine benutzerdefinierte VPC mit einem internen Load Balancer verwendet wird.

Themen

- Der Assistent zum Erstellen einer neuen Umgebung
- Elastic Beanstalk-Umgebung klonen
- Elastic Beanstalk-Umgebung terminieren
- Elastic Beanstalk Beanstalk-Umgebungen mit der CLI erstellen AWS
- Elastic Beanstalk-Umgebungen mit der API erstellen
- Erstellen einer Launch-Now-URL
- Gruppen von Elastic Beanstalk Umgebungen erstellen oder aktualisieren

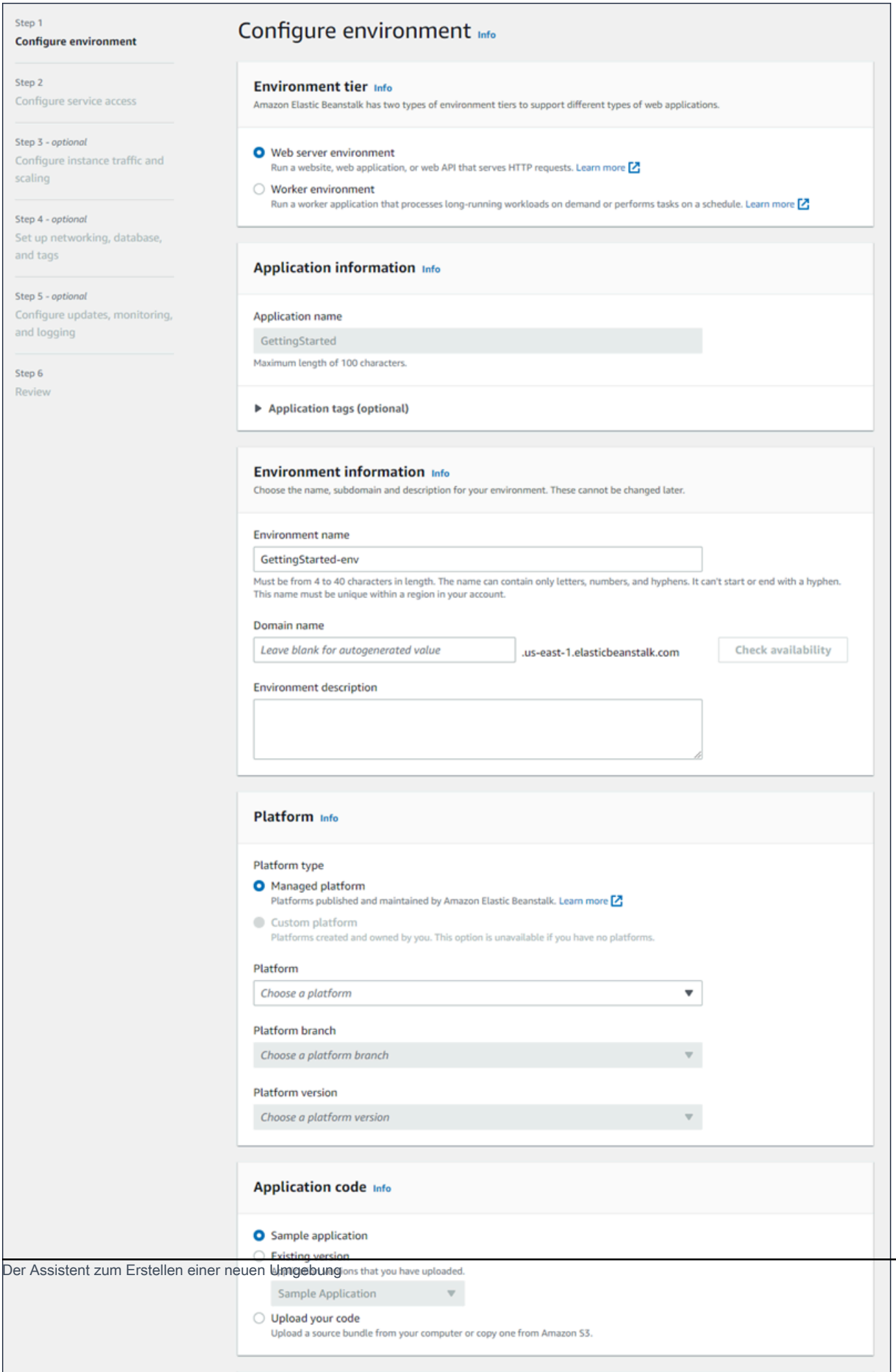
Der Assistent zum Erstellen einer neuen Umgebung

Dieses Thema beschreibt den Assistenten Create environment (Erstellen einer Umgebung) und alle Möglichkeiten, wie sie ihn zur Konfiguration der Umgebung nutzen können, die Sie erstellen möchten.

> **Note**
>
> Darin zeigen Erstellen einer Elastic Beanstalk-Umgebung wir, wie Sie den Assistenten zum Erstellen einer Umgebung starten und schnell eine Umgebung mit Standardwerten und empfohlenen Einstellungen erstellen. In diesem aktuellen Thema werden Sie durch alle Optionen geführt.

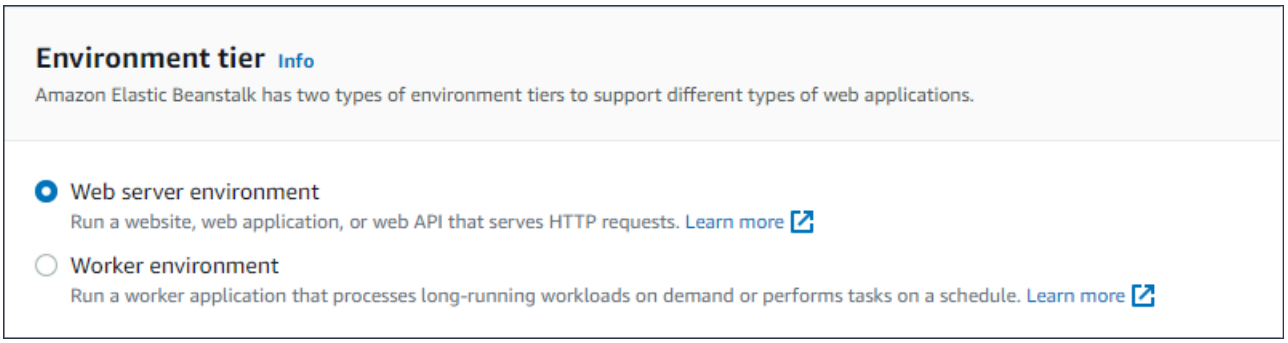
Assistentenseite

Der Assistent Create environment (Umgebung erstellen) enthält eine Reihe von Schritten, mit denen Sie eine neue Umgebung erstellen können.


Step 1
Configure environment
Step 2
Configure service access
Step 3 - optional
Configure instance traffic and scaling
Step 4 - optional
Set up networking, database, and tags
Step 5 - optional
Configure updates, monitoring, and logging
Step 6
Review
Configure environment Info
Environment tier Info
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.
Web server environment
Run a website, web application, or web API that serves HTTP requests. Learn more
Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. Learn more
Application information Info
Application name
GettingStarted
Maximum length of 100 characters.
Application tags (optional)
Environment information Info
Choose the name, subdomain and description for your environment. These cannot be changed later.
Environment name
GettingStarted-env
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.
Domain name
Leave blank for autogenerated value
.us-east-1.elasticbeanstalk.com
Check availability
Environment description
Platform Info
Platform type
Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. Learn more
Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.
Platform
Choose a platform
Platform branch
Choose a platform branch
Platform version
Choose a platform version
Application code Info
Sample application
Existing version
ons that you have uploaded.
Sample Application
Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

Stufe „Umwelt“

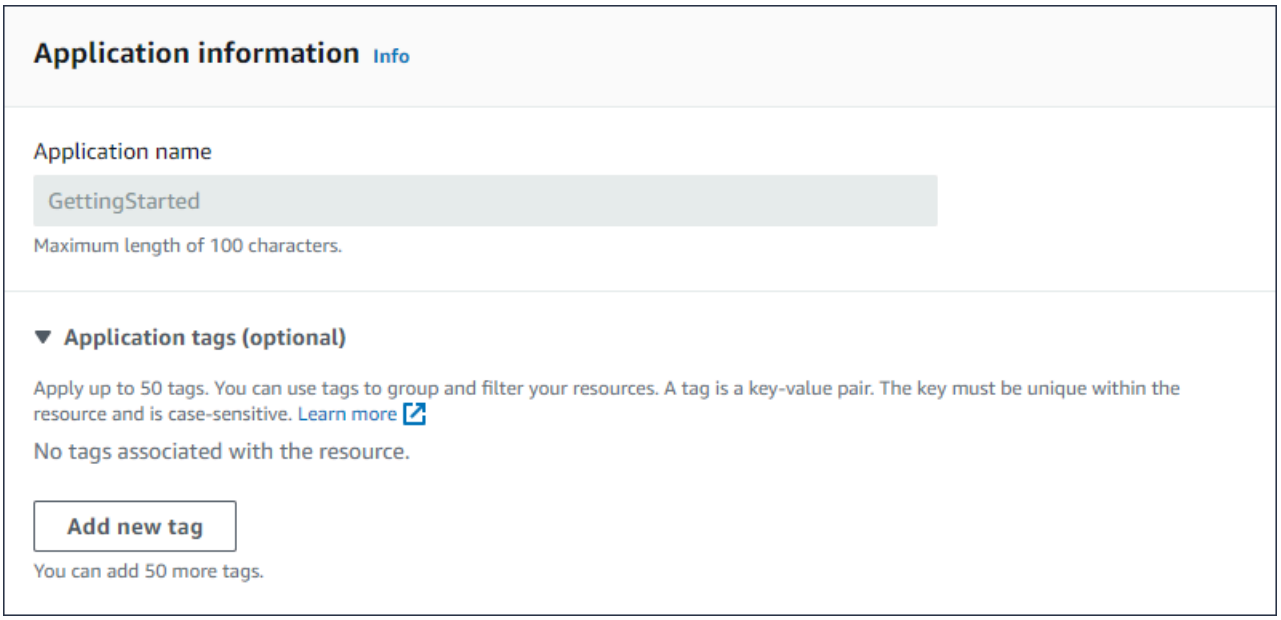
Als Environment tier (Umgebungsebene) wählen Sie die Umgebungsebenen Web server environment (Webserverumgebung) oder Worker environment (Worker-Umgebung) aus. Nach der Erstellung lässt sich die Umgebungsebene nicht mehr ändern.



> **Note**
>
> Die .NET-Plattform auf Windows Server unterstützt die Worker-Umgebungsebene nicht.

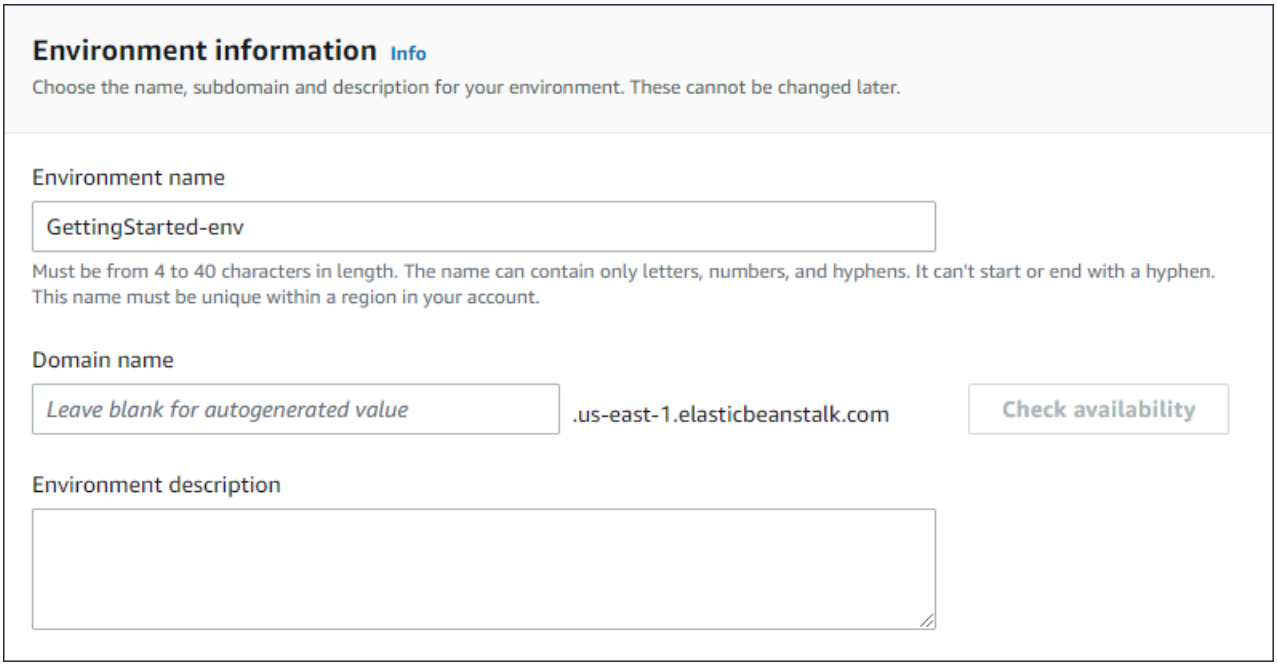
Informationen zur Anwendung

Wenn Sie den Assistenten gestartet haben, indem Sie auf der Seite Application overview (Anwendungsübersicht) die Option Create new environment (Neue Umgebung erstellen) ausgewählt haben, wird der Application name (Anwendungsname) vorab ausgefüllt. Andernfalls geben Sie einen Anwendungsnamen ein. Fügen Sie optional Anwendungs-Tags hinzu.



Informationen zur Umgebung

Legen Sie den Namen und die Domain der Umgebung fest und erstellen Sie eine Beschreibung für Ihre Umgebung. Beachten Sie, dass diese Einstellungen nach der Umgebungserstellung nicht mehr geändert werden können.



- Name – Geben Sie einen Namen für die Umgebung ein. Das Formular stellt einen generierten Namen bereit.
- Domain – (Webserverumgebungen) Geben Sie einen eindeutigen Domainnamen für Ihre Umgebung ein. Der Standardname ist der Name der Umgebung. Sie können einen anderen Domainnamen verwenden. Elastic Beanstalk verwendet diesen Namen, um einen eindeutigen CNAME für die Umgebung zu erstellen. Um zu überprüfen, ob der gewünschte Domainname verfügbar ist, klicken Sie auf Verfügbarkeit prüfen.
- Description – Geben Sie eine Beschreibung für diese Umgebung ein.

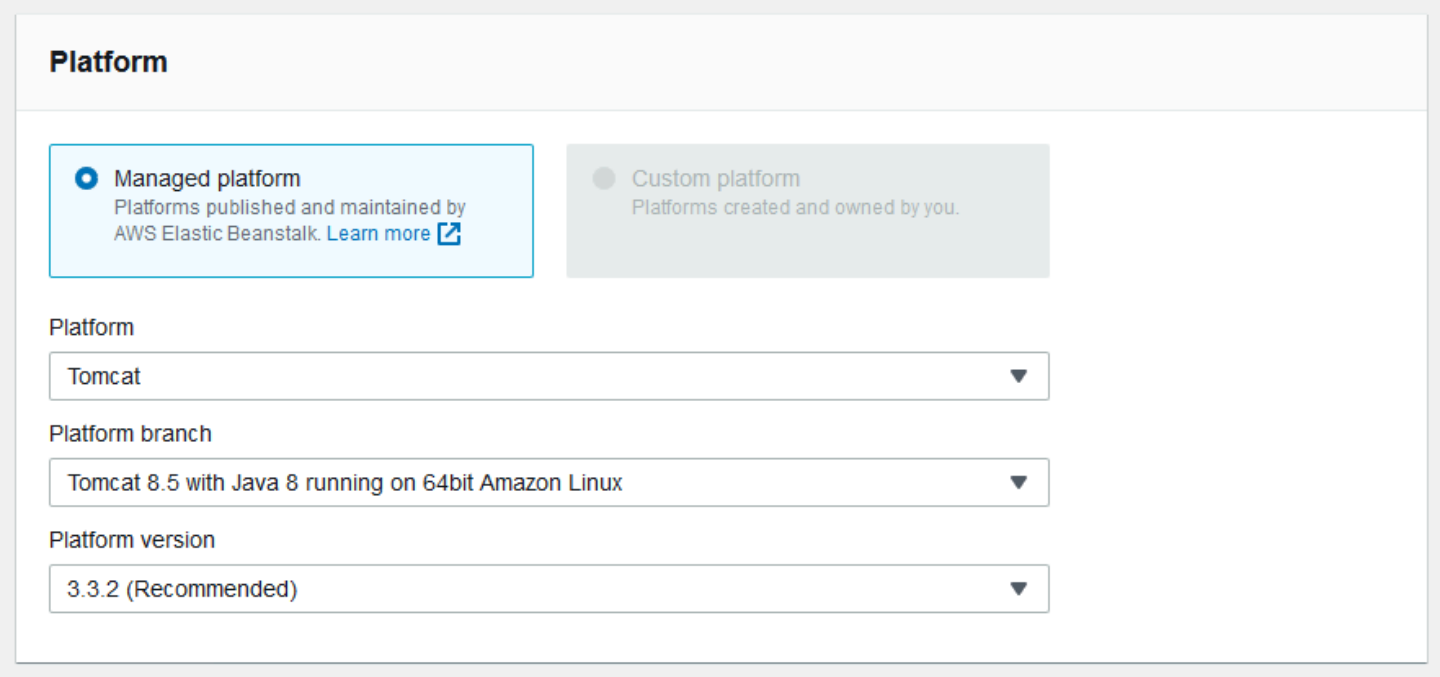
Auswählen der Plattform für die neue Umgebung

Sie können eine neue Umgebung aus zwei Arten von Plattformen erstellen:

- Verwaltete Plattform
- Benutzerdefinierte Plattform

Verwaltete Plattform

In den meisten Fällen verwenden Sie eine von Elastic Beanstalk verwaltete Plattform für Ihre neue Umgebung. Wenn der Assistent für die neue Umgebung gestartet wird, wählt dieser standardmäßig die Option Managed platform (Verwaltete Plattform) aus.



Wählen Sie eine Plattform, einen Plattformzweig innerhalb dieser Plattform und im Zweig eine bestimmte Plattformversion aus. Wenn Sie einen Plattformzweig auswählen, wird standardmäßig die empfohlene Version innerhalb des Zweigs ausgewählt. Darüber hinaus können Sie eine Plattformversion auswählen, die Sie zuvor verwendet haben.

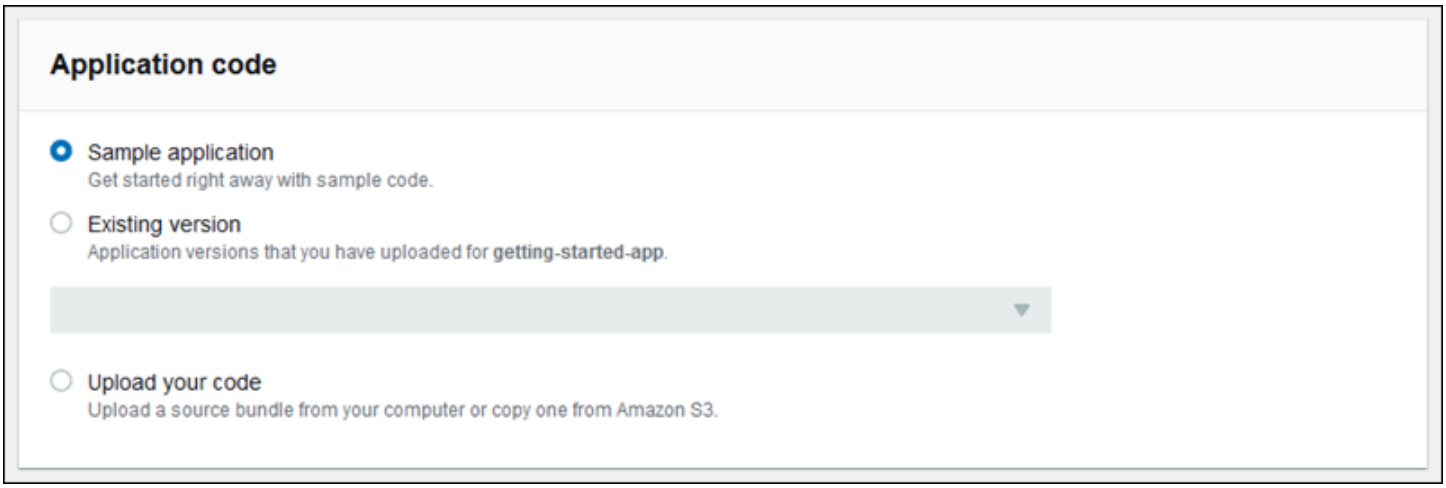
> **Note**
>
> Im Fall einer Produktionsumgebung sollten Sie eine Plattformversion in einem unterstützten Plattformzweig auswählen. Details zu den Statusarten von Plattformzweigen finden Sie in der Definition von Plattformzweig im the section called “Glossar zu Plattformen”.

Benutzerdefinierte Plattform

Wenn eine off-the-shelf Plattform Ihren Anforderungen nicht entspricht, können Sie auf einer benutzerdefinierten Plattform eine neue Umgebung erstellen. Um eine benutzerdefinierte Plattform anzugeben, wählen Sie die Option Custom platform (Benutzerdefinierte Plattform) und anschließend eine der verfügbaren benutzerdefinierten Plattformen aus. Wenn keine benutzerdefinierten Plattformen zur Verfügung stehen, ist diese Option nicht verfügbar.

Bereitstellen von Anwendungscode

Nachdem Sie die Plattform ausgewählt haben, stellen Sie im nächsten Schritt den Anwendungscode bereit.

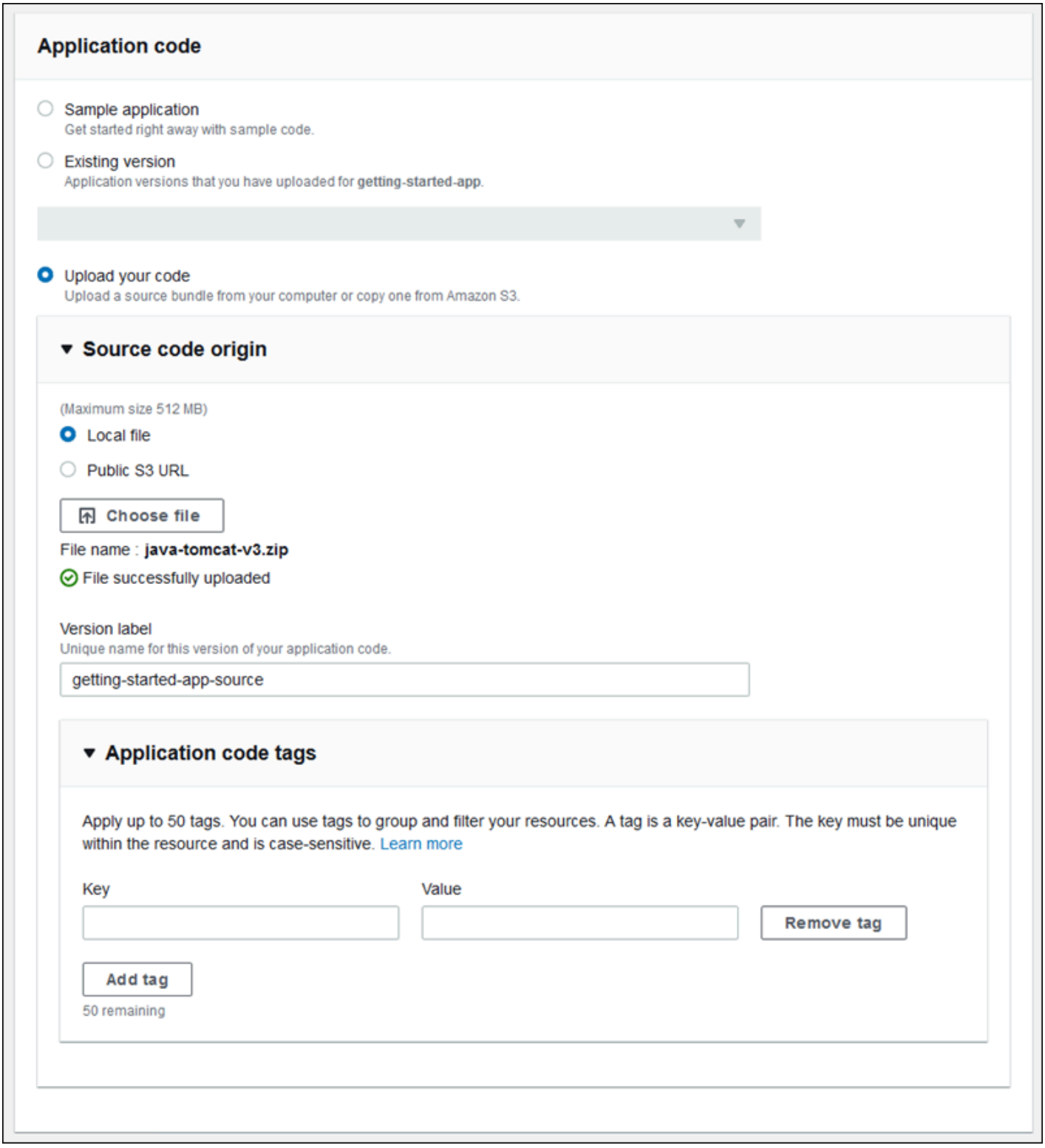


Sie haben mehrere Möglichkeiten:

- Sie können die Beispielanwendung verwenden, die Elastic Beanstalk für jede Plattform bereitstellt.
- Sie können Code verwenden, den Sie bereits für Elastic Beanstalk bereitgestellt haben. Wählen Sie Existing version (Bestehende Version) und die Anwendung im Bereich Application code (Anwendungscode) aus.
- Sie können neuen Code hochladen. Wählen Sie Upload your code (Code hochladen) und dann Upload (Hochladen) aus. Sie können den neuen Anwendungscode aus einer lokalen Datei hochladen oder die URL für den Amazon S3-Bucket mit Ihrem Anwendungscode angeben.

> **Note**
>
> Abhängig von der ausgewählten Plattformversion können Sie Ihre Anwendung in einem ZIP-basierten Quell-Bundle, in einer WAR-Datei oder in einer Klartext-Docker-Konfiguration hochladen. Die maximale Dateigröße beträgt 500 MB.

Wenn Sie neuen Code hochladen möchten, können Sie auch Tags angeben, die mit dem neuen Code verknüpft werden sollen. Weitere Informationen zum Markieren einer Anwendungsversion finden Sie unter the section called “Markieren von Anwendungsversionen”.



Um eine Umgebung schnell mithilfe von Standardkonfigurationsoptionen zu erstellen, können Sie jetzt Create environment (Umgebung erstellen) auswählen. Wählen Sie Configure more options

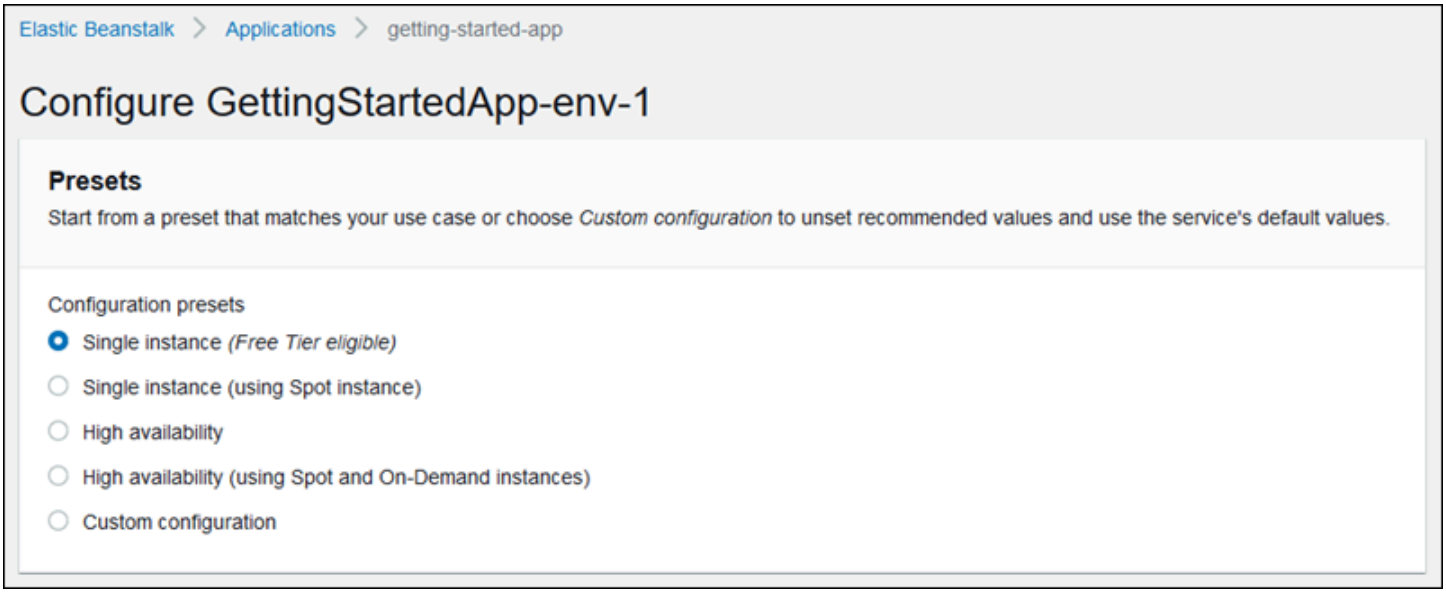
(Weitere Optionen konfigurieren) aus, um zusätzliche Konfigurationsänderungen anzugeben, wie in den folgenden Abschnitten beschrieben.

Seite „Assistentenkonfiguration“

Bei Auswahl von Configure more options (Weitere Optionen konfigurieren) wird im Assistenten die Seite Configure (Konfigurieren) angezeigt. Auf dieser Seite können Sie eine voreingestellte Konfiguration auswählen, die Plattformversion ändern, die Ihre Umgebung verwenden soll, oder bestimmte Konfigurationsoptionen für die neue Umgebung auswählen.

Auswahl einer voreingestellten Konfiguration

Im Abschnitt Presets (Voreinstellungen) der Seite stellt Elastic Beanstalk verschiedene voreingestellte Konfigurationen für verschiedene Anwendungsfälle bereit. Jede Voreinstellung enthält empfohlene Werte für mehrere Konfigurationsoptionen.



Die Voreinstellungen für High Availability (Hohe Verfügbarkeit) enthalten einen Load Balancer und werden für Produktionsumgebungen empfohlen. Wählen Sie diese aus, wenn Sie eine Umgebung wünschen, in der mehrere Instances ausgeführt werden können – für eine hohe Verfügbarkeit und Skalierbarkeit bei Lastenänderungen. Die Single Instance (Einzel-Instance)-Voreinstellungen werden in erster Linie für die Entwicklung empfohlen. Zwei der Voreinstellungen ermöglichen Spot-Instance-Anforderungen. Einzelheiten zur Kapazitätskonfiguration von Elastic Beanstalk finden Sie unter Auto Scaling-Gruppe.

Die letzte Voreinstellung, Custom configuration (Benutzerdefinierte Konfiguration), entfernt alle empfohlenen Werte außer den Rolleneinstellungen und verwendet die API-Standardeinstellungen.

Wählen Sie diese Option, wenn Sie ein Quell-Bundle mit Konfigurationsdateien bereitstellen, die Konfigurationsoptionen festlegen. Custom configuration (Benutzerdefinierte Konfiguration) ist auch automatisch ausgewählt, wenn Sie die Konfigurationsvoreinstellungen Low cost (Niedrige Kosten) oder High availability (Hohe Verfügbarkeit) ändern.

Anpassen Ihrer Konfiguration

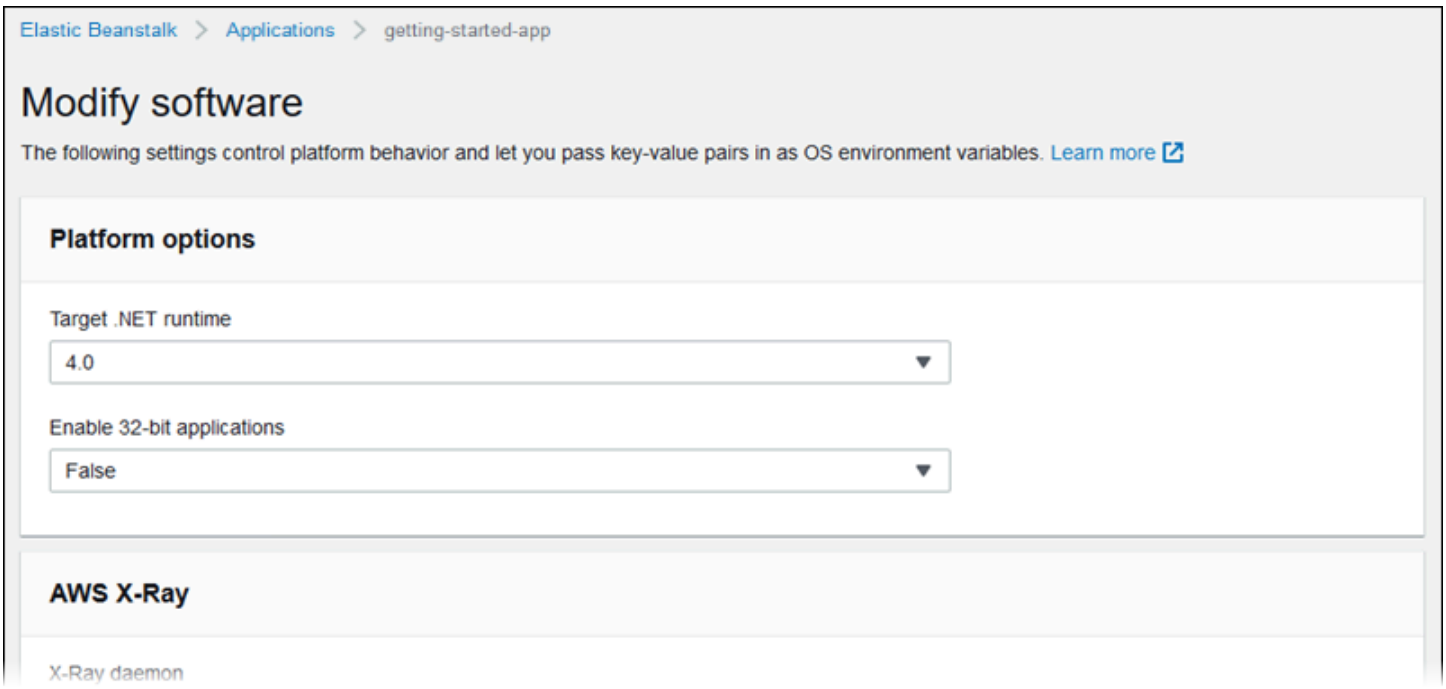
Zusätzlich zu (oder anstelle von) einer ausgewählten Konfigurationsvoreinstellung können Sie die Konfigurationsoptionen in Ihrer Umgebung feiner abstimmen. Der Assistent Configure (Konfigurieren) zeigt mehrere Konfigurationskategorien an. Jede Konfigurationskategorie zeigt eine Übersicht der Werte für eine Gruppe von Konfigurationseinstellungen an. Wählen Sie Edit (Bearbeiten) aus, um diese Einstellungsgruppe zu bearbeiten.

Konfigurationskategorien

- Softwareeinstellungen
- Instances
- Capacity (Kapazität)
- Load Balancer
- Fortlaufende Updates und Bereitstellungen
- Sicherheit
- Überwachen
- Verwaltete Aktualisierungen
- Benachrichtigungen
- Netzwerk
- Datenbank
- Tags (Markierungen)
- Worker-Umgebung

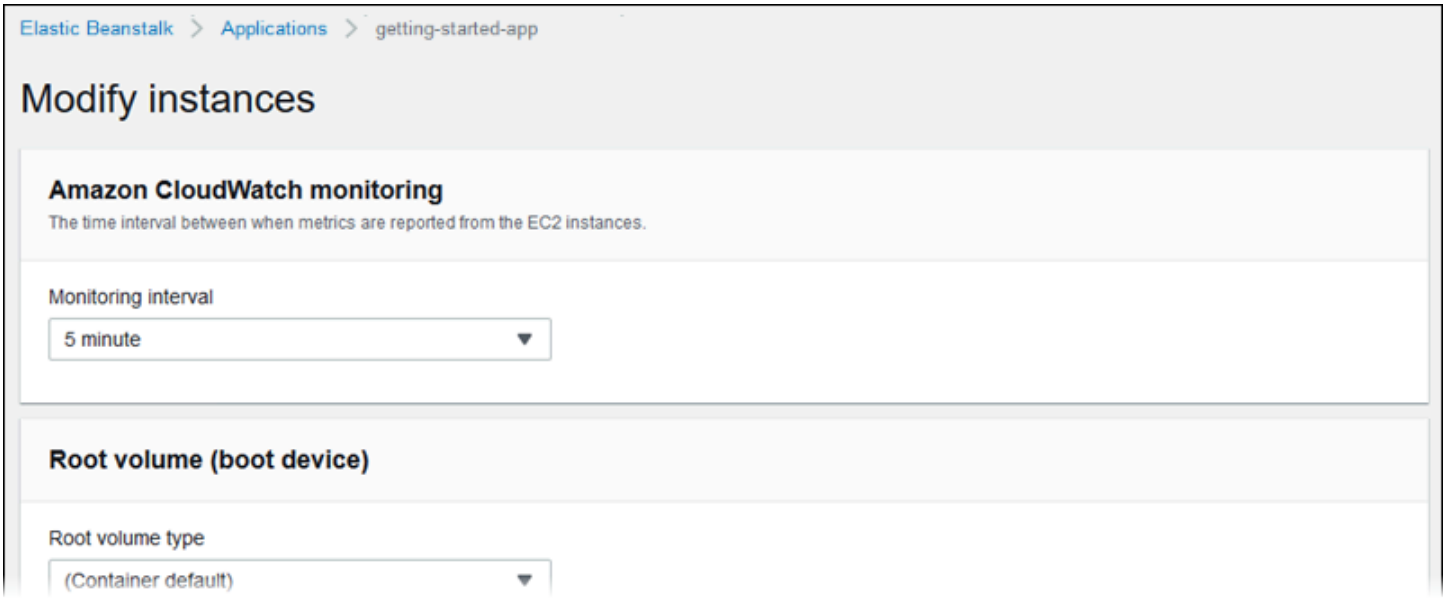
Softwareeinstellungen

Verwenden Sie die Seite Softwarekonfiguration ändern, um die Software auf den Amazon Elastic Compute Cloud (Amazon EC2) -Instances zu konfigurieren, auf denen Ihre Anwendung ausgeführt wird. Sie können Umgebungseigenschaften, AWS X-Ray Debugging, Speichern und Streamen von Instance-Protokollen sowie plattformspezifische Einstellungen konfigurieren. Details hierzu finden Sie unter the section called “Umgebungsvariablen und Softwareeinstellungen”.



Instances

Verwenden Sie die Konfigurationsseite Instances modifizieren, um die EC2 Amazon-Instances zu konfigurieren, auf denen Ihre Anwendung ausgeführt wird. Details hierzu finden Sie unter the section called “ EC2 Amazon-Instanzen”.



Capacity (Kapazität)

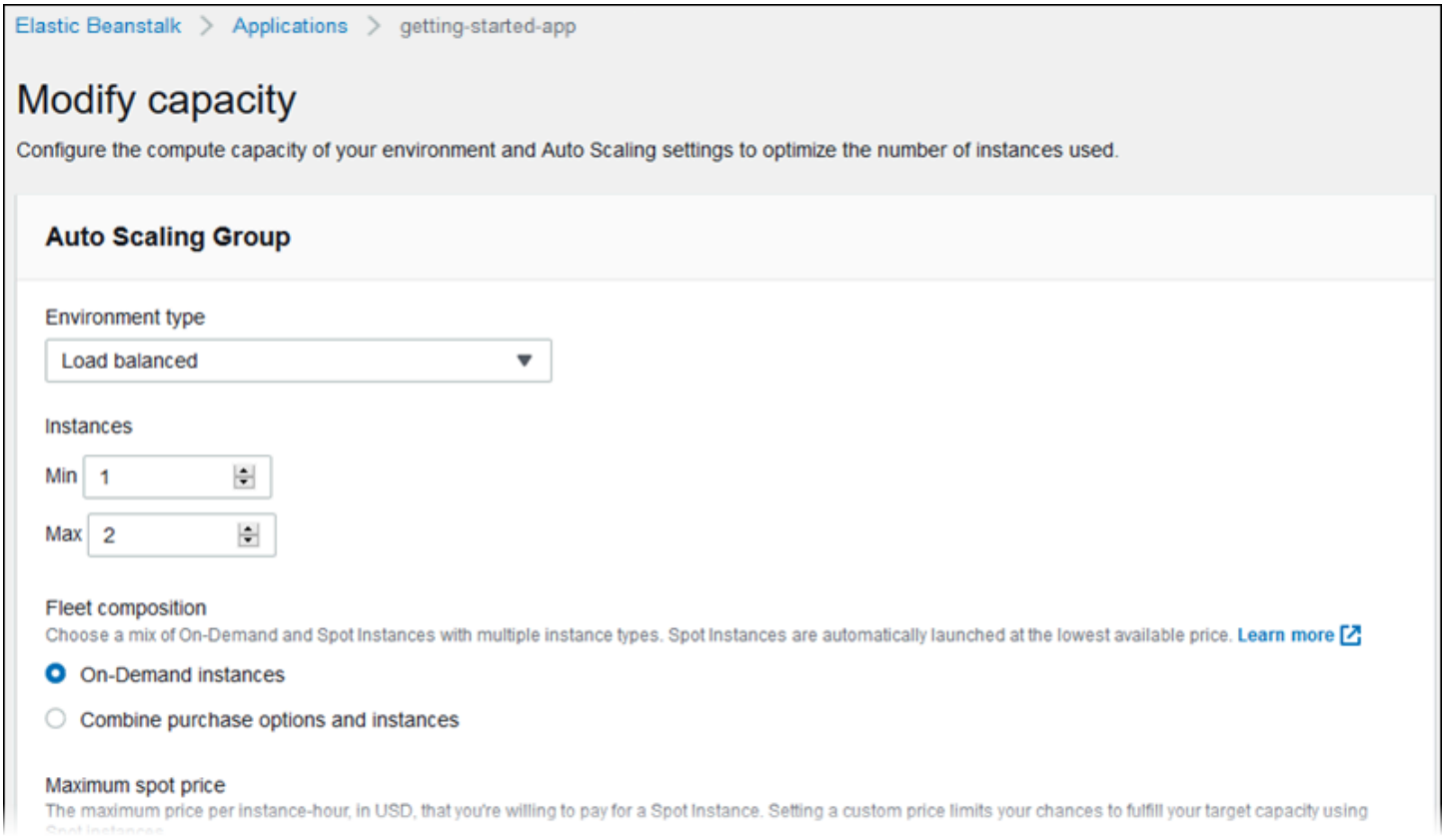
Auf der Konfigurationsseite Modify capacity (Kapazität ändern) können Sie die Rechenkapazität Ihrer Umgebung und die Einstellungen für Auto Scaling group (Auto Scaling-Gruppe konfigurieren,

um Anzahl und Typ der von Ihnen verwendeten Instances zu optimieren. Sie können die Umgebungskapazität auch auf der Grundlage von Auslösern oder Zeitplänen ändern.

Eine Umgebung mit Lastausgleich kann mehrere Instances für hohe Verfügbarkeit ausführen und Ausfallzeiten während Konfigurations-Updates und Bereitstellungen verhindern. In einer Umgebung mit Load Balancing entspricht der Domainname dem Load Balancer. In einer Umgebung mit einer Instance entspricht er einer Elastic IP-Adresse auf der Instance.

> **Warning**
>
> Eine Umgebung mit einer einzelnen Instance ist nicht geeignet für die Produktion. Falls die Instance während der Bereitstellung instabil wird, oder Elastic Beanstalk beendet wird und die Instance während eines Konfigurations-Update neu gestartet wird, steht Ihre Anwendung möglicherweise längere Zeit nicht zur Verfügung. Umgebungen mit einer einzelnen Instance können Sie für die Entwicklung, das Testen oder ein Staging nutzen. Für die Produktion verwenden Sie Umgebungen mit Load Balancing.

Weitere Informationen zu Umgebungskapazitätseinstellungen finden Sie unter the section called “Auto Scaling-Gruppe” and the section called “ EC2 Amazon-Instanzen”.

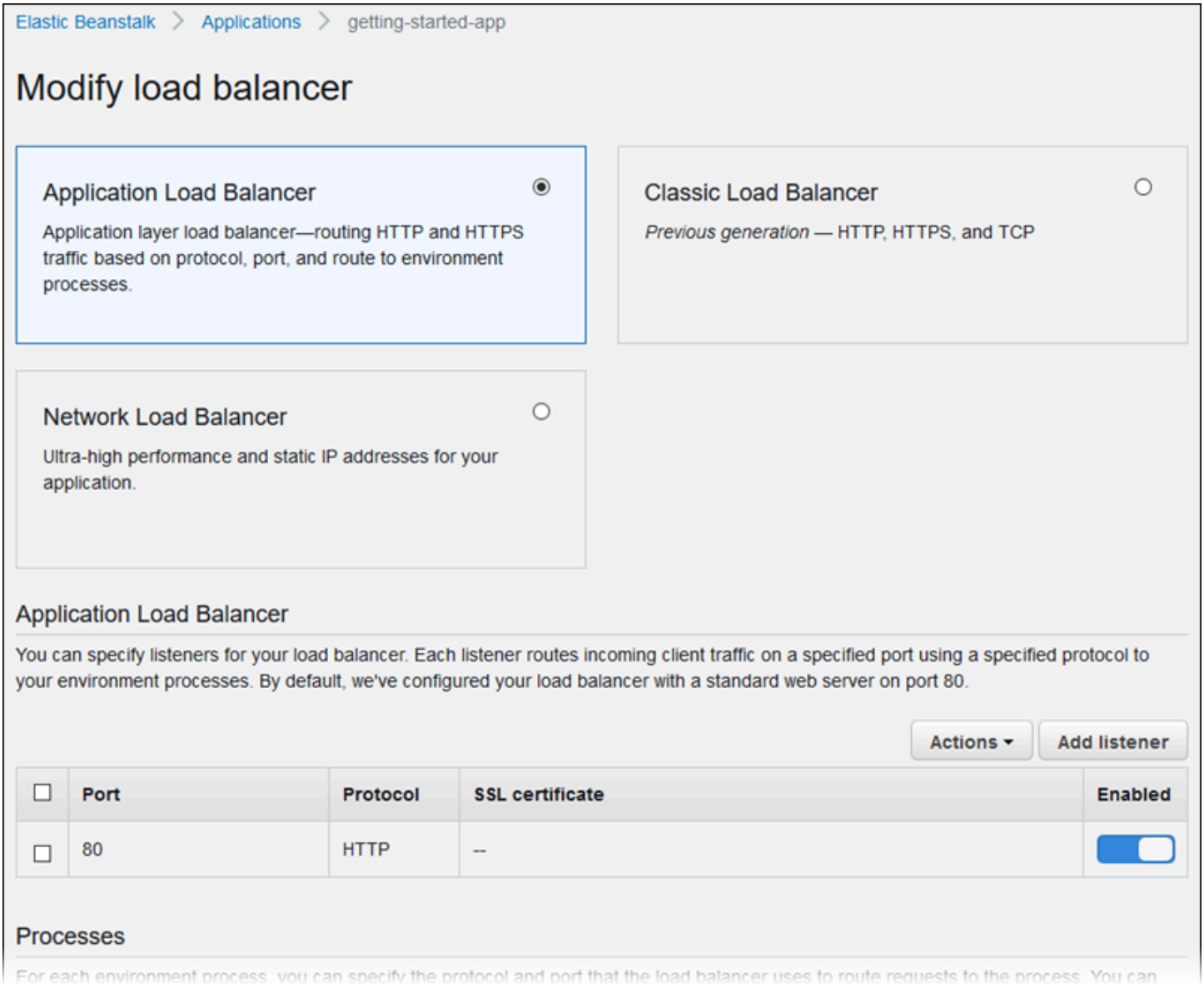


Load Balancer

Wählen Sie auf der Konfigurationskarte Modify load Balancer (Load Balancer ändern) einen Load Balancer-Typ aus und konfigurieren Sie die zugehörigen Einstellungen. In einer Umgebung mit Load Balancing ist der Load Balancer Ihrer Umgebung der Eintrittspunkt für den gesamten Datenverkehr, der Ihre Anwendung ansteuert. Elastic Beanstalk unterstützt verschiedene Arten von Load Balancern. Standardmäßig erstellt die Elastic Beanstalk-Konsole einen Application Load Balancer und konfiguriert ihn so, dass er HTTP-Datenverkehr auf Port 80 bedient.

> **Note**
>
> Sie können den Typ des Load Balancer Ihrer Umgebung nur während der Umgebungserstellung auswählen.

Weitere Informationen zu Typen und Einstellungen von Load Balancern finden Sie unter the section called “Load Balancer” und the section called “HTTPS”.

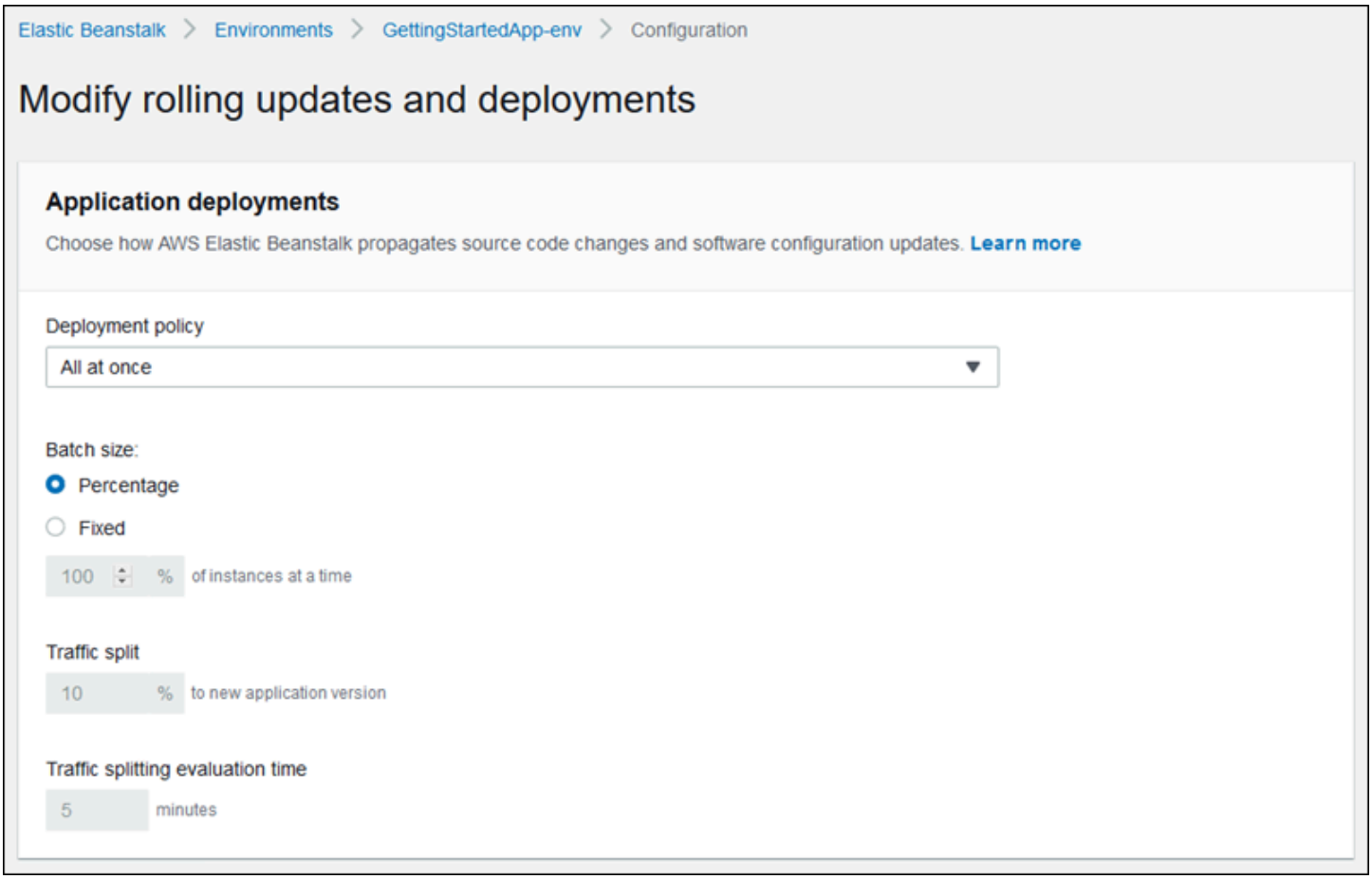


> **Note**
>
> Die Option „Classic Load Balancer (CLB)“ ist im Konsolenassistenten Umgebung erstellen deaktiviert. Wenn Sie eine bestehende Umgebung mit einem Classic Load Balancer konfiguriert haben, können Sie eine neue erstellen, indem Sie die bestehende Umgebung entweder mit der Elastic-Beanstalk-Konsole oder der EB CLI klonen. Sie haben auch die Möglichkeit, die EB-CLI zu verwenden oder eine neue Umgebung AWS CLIzu erstellen, die mit einem Classic Load Balancer konfiguriert ist. Diese Befehlszeilentools erstellen eine neue Umgebung mit einem CLB, auch wenn in Ihrem Konto noch keiner vorhanden ist.

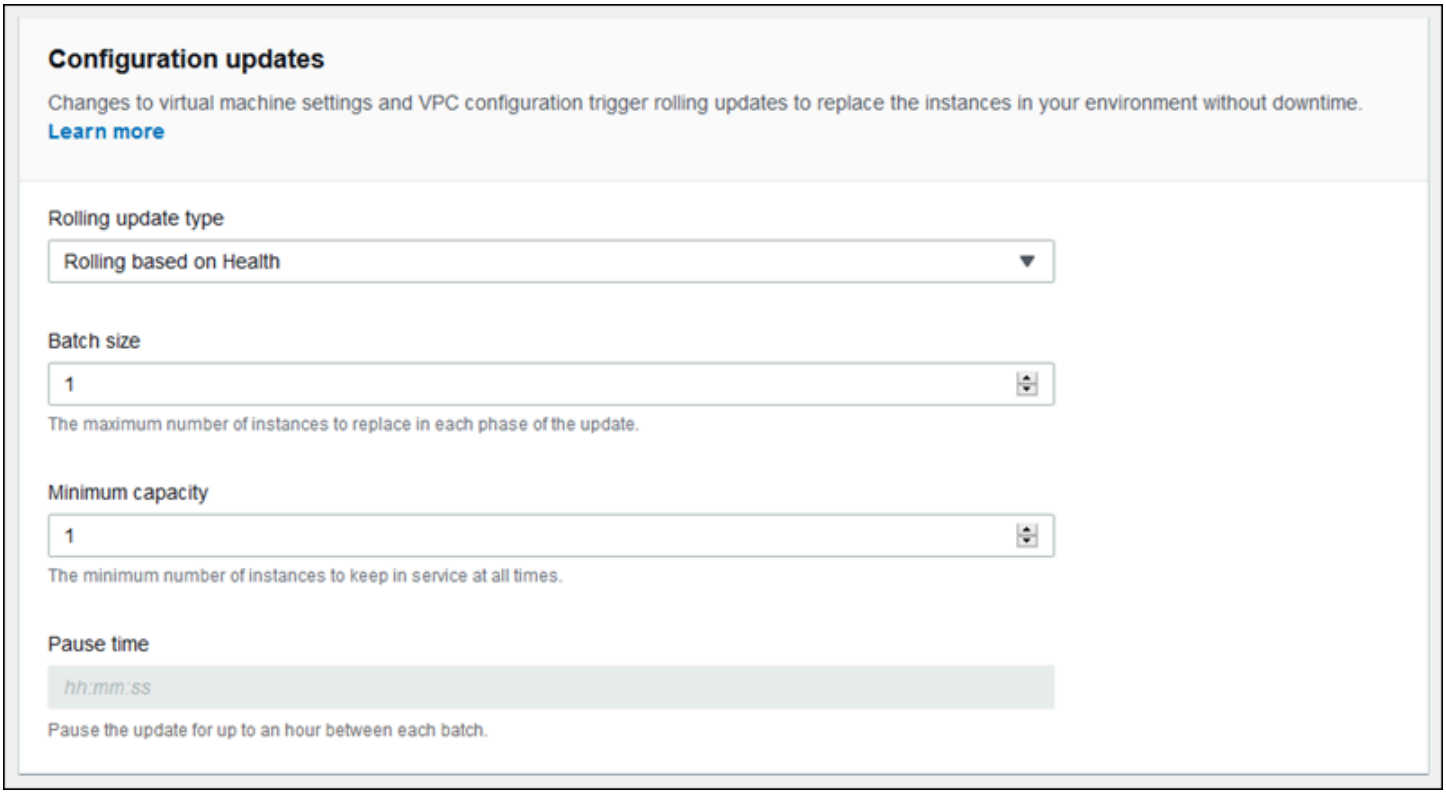
Fortlaufende Updates und Bereitstellungen

Auf der Konfigurationsseite Modify rolling updates and deployments (Rollierende Updates und Bereitstellungen) können Sie die Verarbeitung von Anwendungsbereitstellungen und Konfigurationsupdates für Ihre Umgebung durch Elastic Beanstalk konfigurieren.

Anwendungsbereitstellungen werden ausgeführt, wenn Sie ein aktuelles Anwendungsquellpaket hochladen und in Ihrer Umgebung bereitstellen. Weitere Informationen zur Konfigurierung von Bereitstellungen finden Sie unter the section called “Optionen für die Bereitstellung”.



Falls mit Konfigurationsänderungen die Startkonfiguration oder die VPC-Einstellungen geändert werden, müssen alle Instances in der Umgebung beendet und ersetzt werden. Weitere Informationen zur Festlegung des Update-Typs und anderer Optionen finden Sie unter the section called “Konfigurationsänderungen”.



Sicherheit

Auf der Seite Configure service access (Service-Zugang konfigurieren) können Sie die Sicherheitseinstellungen von Services und Instances konfigurieren.

Eine Beschreibung der Sicherheitskonzepte von Elastic Beanstalk finden Sie unter Berechtigungen.

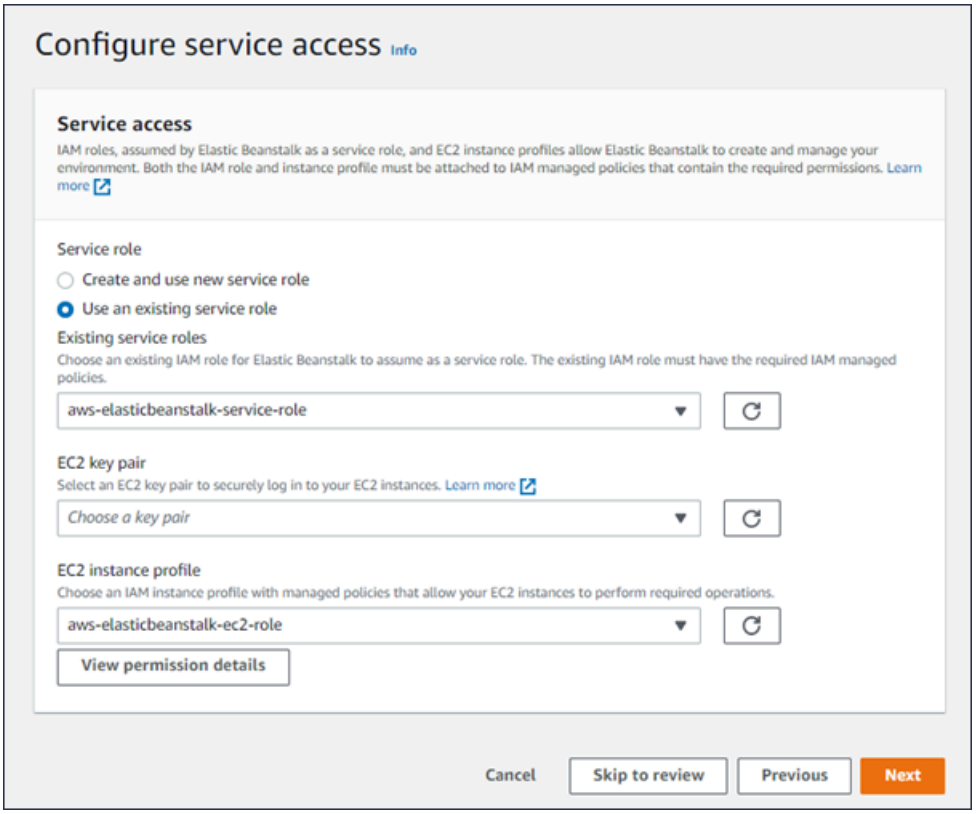
Wenn Sie zum ersten Mal eine Umgebung in der Elastic Beanstalk Beanstalk-Konsole erstellen, müssen Sie ein EC2 Instanzprofil mit einem Standardsatz von Berechtigungen erstellen. Wenn in der Dropdownliste für das EC2 Instanzprofil keine Werte zur Auswahl stehen, erweitern Sie das folgende Verfahren. Es enthält Schritte zum Erstellen einer Rolle, die Sie anschließend für das EC2 Instanzprofil auswählen können.

Erstellen Sie eine IAM-Rolle für das EC2 Instanzprofil

Um ein EC2 Instanzprofil zu erstellen oder auszuwählen

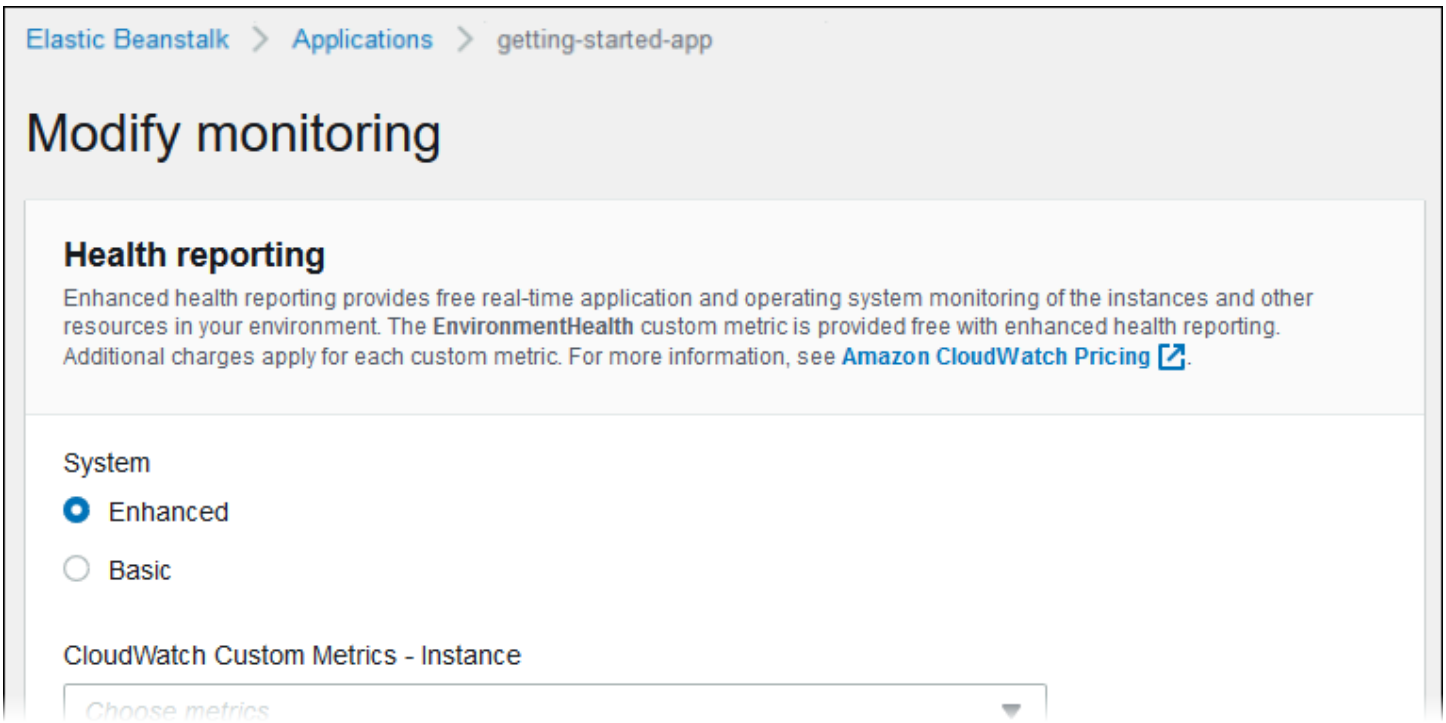
1. Wenn Sie bereits ein EC2 Instanzprofil erstellt haben und ein vorhandenes auswählen möchten, wählen Sie den Wert aus der Drop-down-Liste für das EC2 Instanzprofil aus und überspringen Sie die restlichen Schritte, um ein EC2 Instanzprofil zu erstellen.

2. Wenn keine Werte für das EC2 Instanzprofil aufgeführt sind oder Sie ein neues erstellen möchten, fahren Sie mit den nächsten Schritten fort.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS -Service aus.
5. Wählen Sie als Anwendungsfall Elastic Beanstalk — Compute.
6. Wählen Sie Weiter aus.
7. Vergewissern Sie sich, dass die Berechtigungsrichtlinien Folgendes beinhalten, und wählen Sie dann Weiter aus:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
8. Wählen Sie Rolle erstellen aus.
9. Kehren Sie zur Registerkarte Dienstzugriff konfigurieren zurück, aktualisieren Sie die Liste und wählen Sie dann das neu erstellte EC2 Instanzprofil aus.



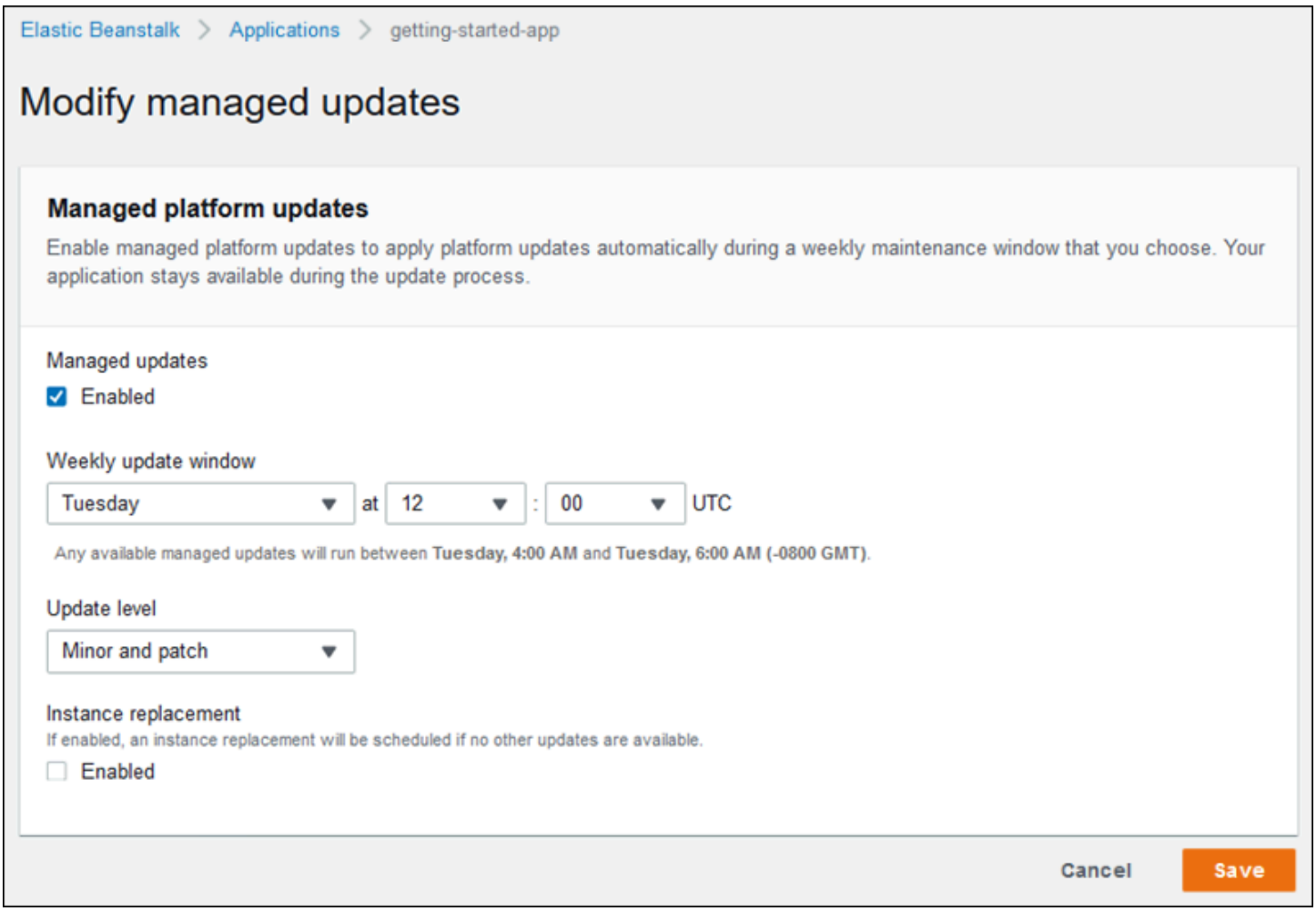
Überwachen

Auf der Konfigurationsseite Modify monitoring (Überwachung ändern) können Sie Integritätsberichte, Überwachungsregeln und das Integritätsereignisstreaming konfigurieren. Details hierzu finden Sie unter the section called “Aktivieren der erweiterten Zustandsberichte”, the section called “Regeln für den erweiterten Zustand” und the section called “Streamen des Umgebungszustands”.



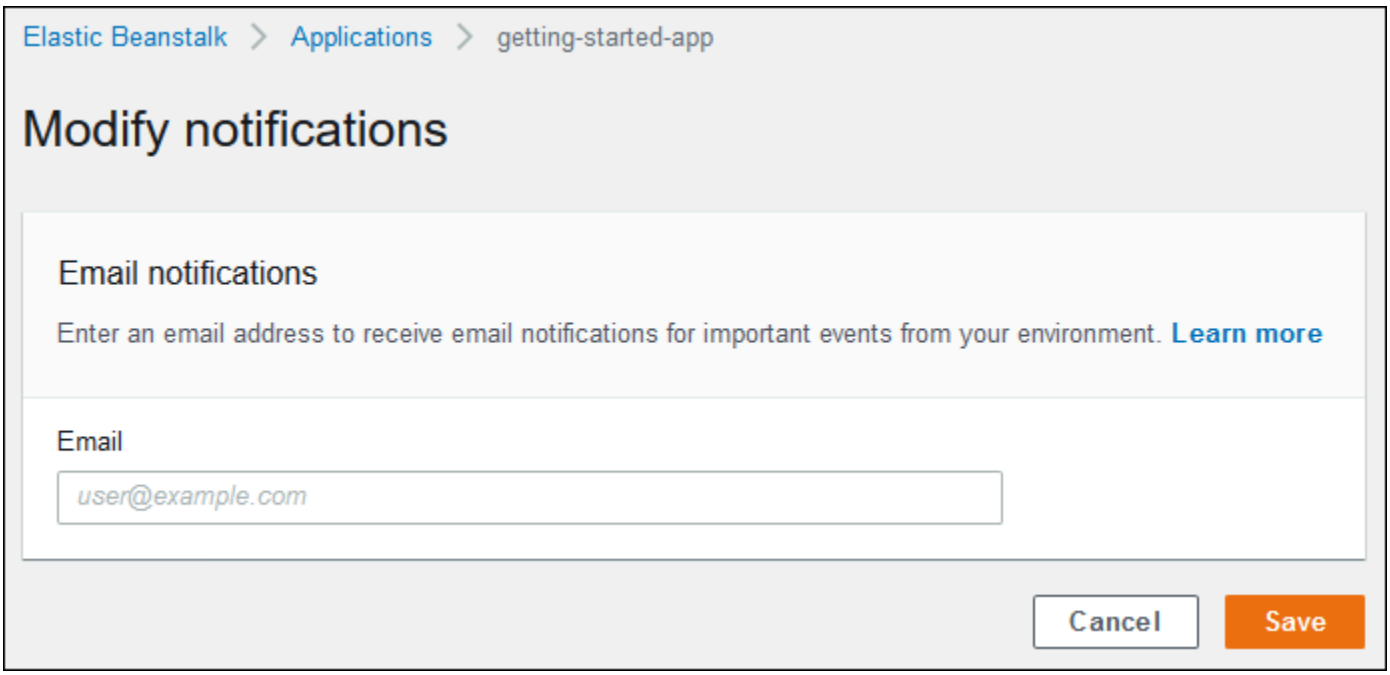
Verwaltete Aktualisierungen

Auf der Konfigurationsseite Modify managed updates (Verwaltete Aktualisierungen ändern) können Sie Updates für die verwaltete Plattform konfigurieren. Sie können festlegen, ob sie aktiviert werden sollen, den Zeitplan angeben und weitere Eigenschaften konfigurieren. Details hierzu finden Sie unter the section called “Verwaltete Aktualisierungen”.



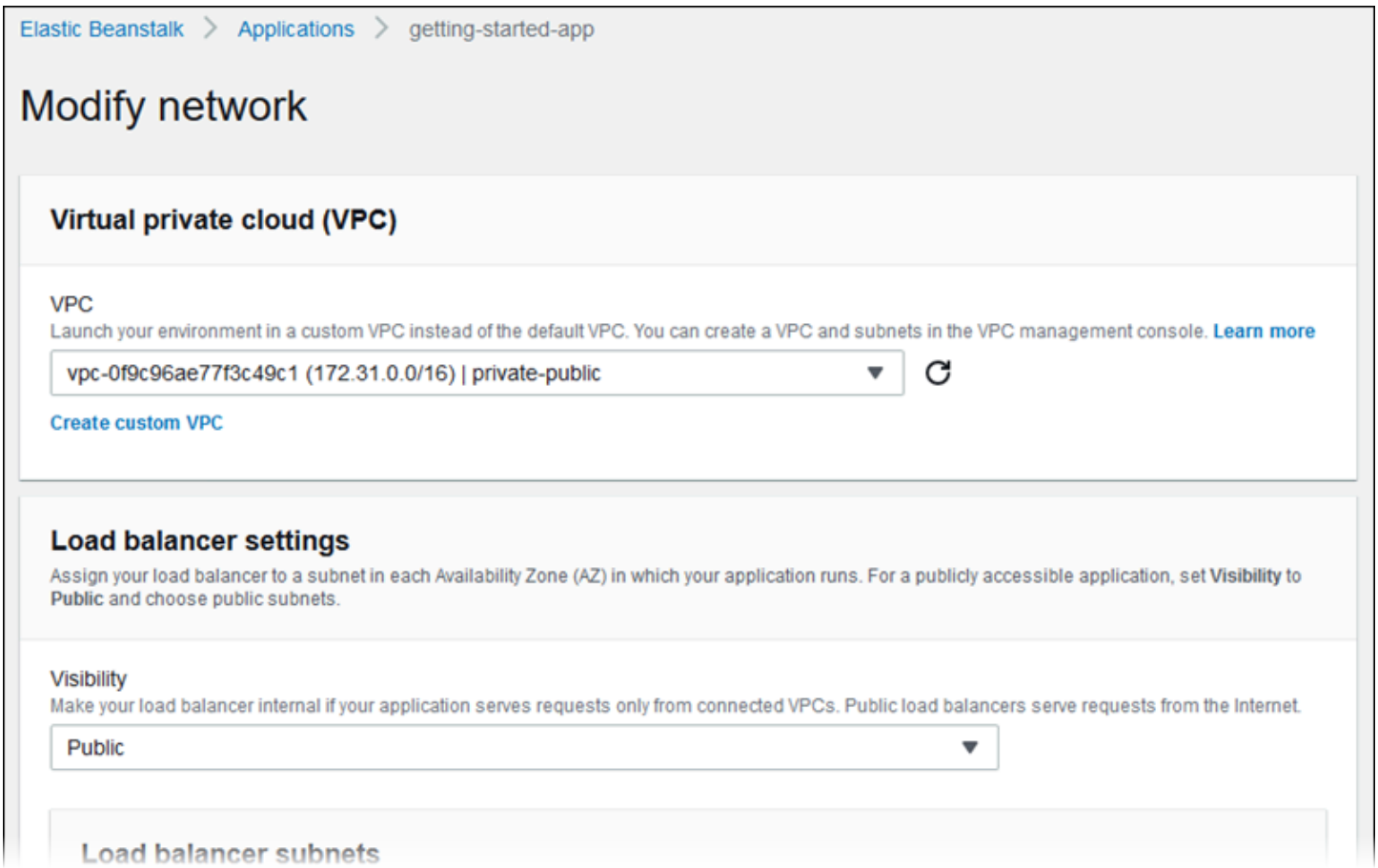
Benachrichtigungen

Auf der Konfigurationsseite Modify notifications (Benachrichtigungen ändern) können Sie eine E-Mail-Adresse angeben, um E-Mail-Benachrichtigungen für wichtige Ereignisse aus Ihrer Umgebung zu erhalten.



Netzwerk

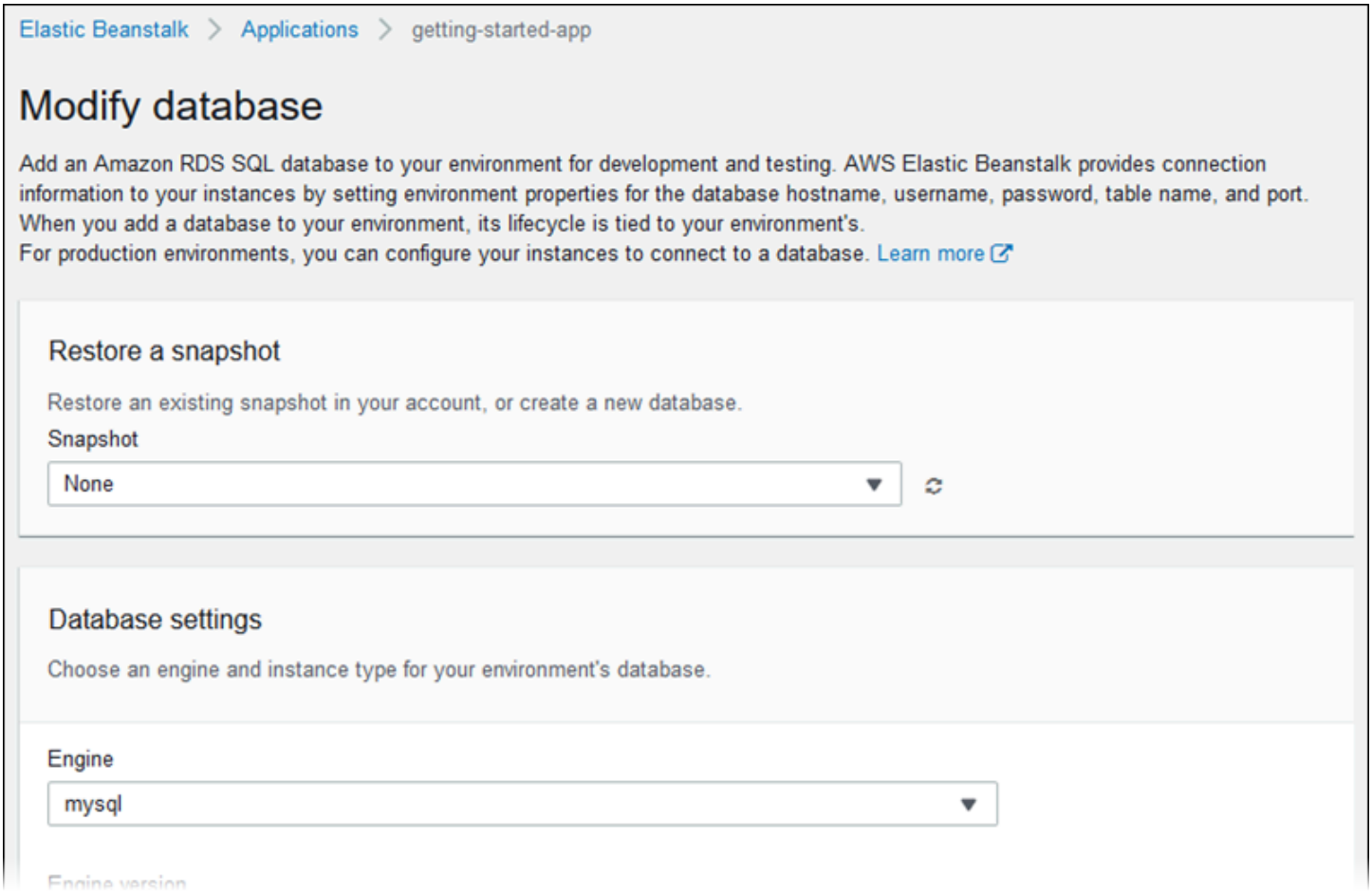
Wenn Sie eine benutzerdefinierte VPC erstellt haben, können Sie Ihre Umgebung auf der Konfigurationsseite Modify network (Netzwerk ändern) für ihre Verwendung konfigurieren. Wenn Sie keine VPC auswählen, verwendet Elastic Beanstalk die Standard-VPC und Subnetze.



Datenbank

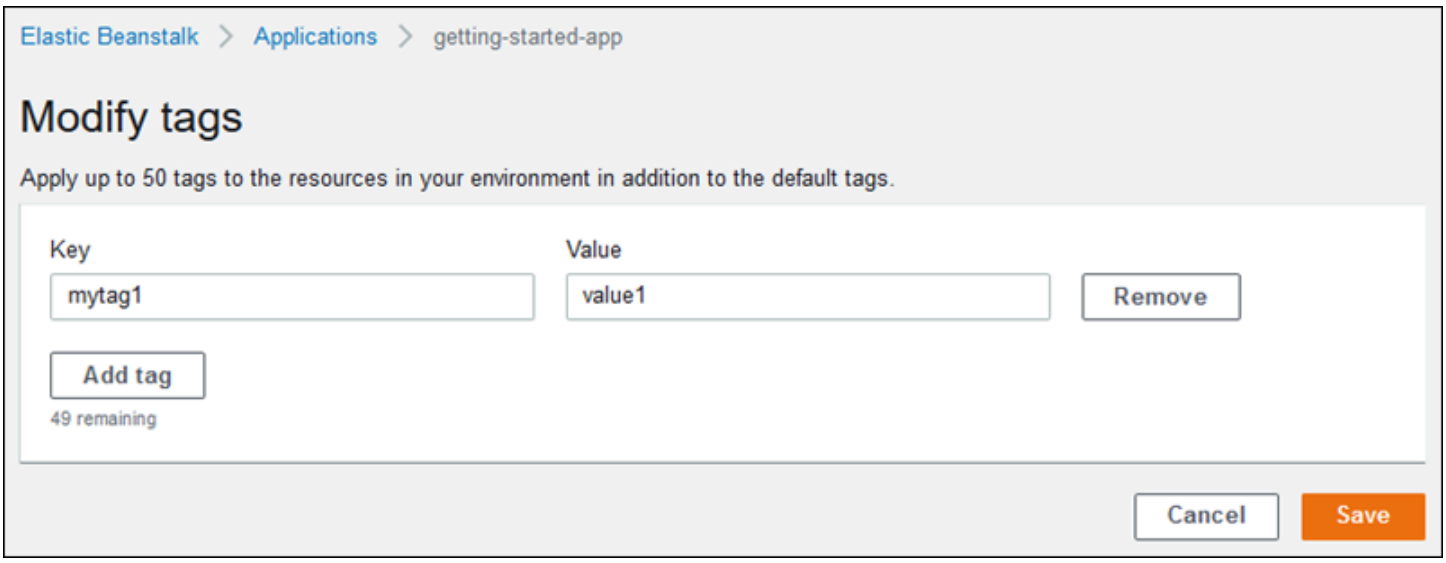
Verwenden Sie die Konfigurationsseite Modify database (Datenbank ändern), um eine Amazon Relational Database Service (Amazon RDS)-Datenbank zu Entwicklungs- und Testzwecken zu Ihrer Umgebung hinzuzufügen. Elastic Beanstalk bietet Verbindungsinformationen zu Ihren Instances, indem Umgebungseigenschaften für den Hostnamen, Benutzernamen, das Passwort, den Tabellennamen und den Port der Datenbank eingerichtet werden.

Details hierzu finden Sie unter the section called “Datenbank”.



Tags (Markierungen)

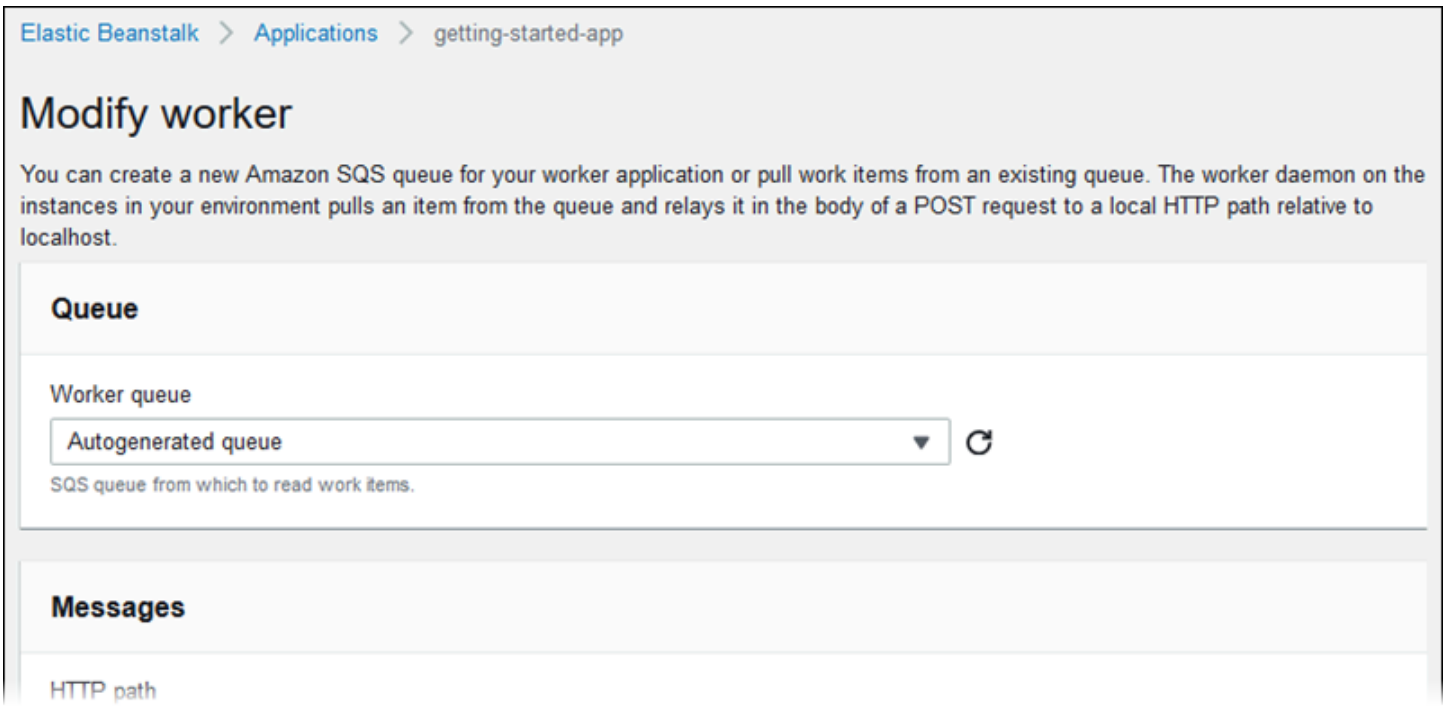
Auf der Konfigurationsseite Modify tags (Tags ändern) können Sie den Ressourcen in Ihrer Umgebung Tags hinzufügen. Weitere Informationen zu Umgebungs-Tags finden Sie unter Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren.



Worker-Umgebung

Wenn Sie eine Worker-Tier-Umgebung erstellen, können Sie auf der Konfigurationsseite Modify Worker (Worker ändern) die Worker-Umgebung konfigurieren. Der Worker-Daemon auf den Instances in Ihrer Umgebung ruft Elemente aus einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange ab und sendet sie als Post-Nachrichten an Ihre Worker-Anwendung. Sie können die Amazon SQS-Warteschlange auswählen, aus der der Worker-Daemon liest (automatisch generiert oder vorhanden). Sie können auch die Nachrichten konfigurieren, die der Worker-Daemon an Ihre Anwendung sendet.

Weitere Informationen finden Sie unter the section called “Worker-Umgebungen”.



Elastic Beanstalk-Umgebung klonen

Durch das Klonen einer vorhandenen Elastic Beanstalk-Umgebung können Sie diese als Grundlage für eine neue Umgebung einsetzen. Beispielsweise können Sie einen Klon erstellen, um eine neuere Version des ursprünglichen Lösungs-Stacks für die Umgebungsplattform zu verwenden. Elastic Beanstalk konfiguriert den Clone mit den Umgebungseinstellungen, die von der ursprünglichen Umgebung verwendet wurden. Indem Sie eine bestehende Umgebung klonen, anstatt eine neue Umgebung zu erstellen, müssen Sie Optionseinstellungen, Umgebungsvariablen und andere Einstellungen, die Sie mit dem Elastic Beanstalk-Service vorgenommen haben, nicht manuell konfigurieren. Elastic Beanstalk erstellt außerdem eine Kopie aller AWS Ressourcen, die mit der ursprünglichen Umgebung verknüpft sind.

Es ist wichtig, sich der folgenden Situationen bewusst zu sein:

- Während des Klonvorgangs kopiert Elastic Beanstalk keine Daten von Amazon RDS auf den Clone.
- Elastic Beanstalk bindet keine nicht verwalteten Ressourcenänderungen in den Klon ein. Änderungen an AWS Ressourcen, die Sie mit anderen Tools als der Elastic-Beanstalk-Konsole, den Befehlszeilen-Tools oder der API vorgenommen haben, gelten als nicht verwaltete Änderungen.

- Die Sicherheitsgruppen für Ingress gelten als nicht verwaltete Änderungen. Bei geklonten Elastic Beanstalk Beanstalk-Umgebungen werden die Sicherheitsgruppen für eingehenden Zugriff nicht übertragen, sodass die Umgebung für den gesamten Internetverkehr geöffnet bleibt. Sie müssen die Sicherheitsgruppen für eingehende Zugriffe für die geklonte Umgebung neu einrichten.

Sie können eine Umgebung nur auf eine andere Plattformversion desselben Plattformzweigs klonen. Ein anderer Plattformzweig ist nicht garantiert kompatibel. Um einen anderen Plattformzweig zu verwenden, müssen Sie manuell eine neue Umgebung erstellen, Ihren Anwendungscode bereitstellen und alle erforderlichen Änderungen an Code und Optionen vornehmen, um sicherzustellen, dass Ihre Anwendung auf dem neuen Plattformzweig ordnungsgemäß funktioniert.

AWS Verwaltungskonsole

> **Important**
>
> Bei geklonten Elastic Beanstalk Beanstalk-Umgebungen werden die Sicherheitsgruppen für eingehenden Zugriff nicht übertragen, sodass die Umgebung für den gesamten Internetverkehr geöffnet bleibt. Sie müssen die Sicherheitsgruppen für eingehende Zugriffe für die geklonte Umgebung neu einrichten.
> Sie können sehen, welche Ressourcen möglicherweise nicht geklont wurden, indem Sie den Drift-Status Ihrer Umgebungskonfiguration überprüfen. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter Drift auf einem gesamten CloudFormation Stack erkennen.

So klonen Sie eine Umgebung

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Seite „Umgebungsübersicht“ Aktionen aus.
4. Wählen Sie Umgebung klonen.
5. Überprüfen Sie auf der Seite Clone Environment (Umgebung klonen) anhand der Informationen unter Original Environment (Ursprüngliche Umgebung), ob Sie die korrekte Umgebung für die Klonerstellung ausgewählt haben.

6. Im Abschnitt New Environment (Neue Umgebung) können Sie optional die Werte für Environment name (Umgebungsname), Environment URL (Umgebungs-URL), Description (Beschreibung), Platform version (Plattformversion) und Service role (Servicerolle) ändern, die Elastic Beanstalk automatisch basierend auf der ursprünglichen Umgebung festlegt.

   > **Note**
   >
   > Wenn es sich bei der in der ursprünglichen Umgebung verwendeten Plattformversion nicht um die für die Verwendung in der Plattformvariante empfohlene Version handelt, werden Sie gewarnt, dass eine andere Plattformversion empfohlen wird. Wählen Sie Platform version (Plattformversion), damit die empfohlene Plattformversion in der Liste angezeigt wird, z. B.3.3.2 (Recommended) (3.3.2 (Empfohlen)).

7. Wenn Sie fertig sind, klicken Sie auf Clone (Klonen).

Elastic Beanstalk Command Line Interface (EB CLI)

> **Important**
>
> Bei geklonten Elastic Beanstalk Beanstalk-Umgebungen werden die Sicherheitsgruppen für eingehenden Zugriff nicht übertragen, sodass die Umgebung für den gesamten Internetverkehr geöffnet bleibt. Sie müssen die Sicherheitsgruppen für eingehende Zugriffe für die geklonte Umgebung neu einrichten.
> Sie können sehen, welche Ressourcen möglicherweise nicht geklont wurden, indem Sie den Drift-Status Ihrer Umgebungskonfiguration überprüfen. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter Drift auf einem gesamten CloudFormation Stack erkennen.

Mit dem Befehl eb clone klonen Sie eine ausgeführte Umgebung, wie hier gezeigt.

```
~/workspace/my-app$ eb clone my-env1
Enter name for Environment Clone
(default is my-env1-clone): my-env2
Enter DNS CNAME prefix
(default is my-env1-clone): my-env2
```

Sie können im Klonbefehl den Namen der Quellumgebung angeben. Wenn Sie dies nicht tun, wird die Standardumgebung des aktuellen Projektordners geklont. Sie werden von der EB CLI aufgefordert, einen Namen und ein DNS-Präfix für die neue Umgebung anzugeben.

Standardmäßig wird die neue Umgebung von eb clone mit der neuesten verfügbaren Version der Quellumgebungsplattform erstellt. Mit der Option `--exact` erzwingen Sie – selbst wenn eine neuere Version vorhanden ist – in der EB CLI die Verwendung der gleichen Version.

```
~/workspace/my-app$ eb clone --exact
```

Weitere Informationen zu diesem Befehl finden Sie unter eb clone.

Elastic Beanstalk-Umgebung terminieren

Sie können eine laufende AWS Elastic Beanstalk Umgebung mit der Elastic Beanstalk Beanstalk-Konsole beenden. Auf diese Weise vermeiden Sie, dass Gebühren für ungenutzte Ressourcen anfallen. AWS

> **Note**
>
> Sie können zu einem späteren Zeitpunkt problemlos eine neue Umgebung mit derselben Version starten.

Wenn Sie über Daten aus einer Umgebung verfügen, die Sie beibehalten möchten, setzen Sie die Richtlinie zum Löschen von Datenbanken auf `Retain`, bevor Sie die Umgebung beenden. Dies hält die Datenbank außerhalb von Elastic Beanstalk betriebsbereit. Danach müssen sich alle Elastic-Beanstalk-Umgebungen als externe Datenbank damit verbinden. Wenn Sie die Daten sichern möchten, ohne die Datenbank betriebsbereit zu halten, legen Sie die Löschrichtlinie so fest, dass eine Snapshot der Datenbank erstellt wird, bevor die Umgebung beendet wird. Weitere Informationen finden Sie unter Datenbank-Lebenszyklus im Kapitel Konfigurieren von Umgebungen in diesem Leitfaden.

Elastic Beanstalk kann Ihre Umgebung möglicherweise nicht terminieren. Ein häufiger Grund ist, dass die Sicherheitsgruppe einer anderen Umgebung von der Sicherheitsgruppe der Umgebung abhängt, die Sie beenden möchten. Anweisungen zur Vermeidung dieses Problems finden Sie EC2 Sicherheitsgruppen auf der EC2Instanzen-Seite dieses Handbuchs.

> **Important**
>
> Wenn Sie eine Umgebung beenden, müssen Sie auch alle CNAME-Zuordnungen löschen, die Sie erstellt haben, damit andere Kunden einen verfügbaren Hostnamen wiederverwenden können. Achten Sie darauf, DNS-Einträge zu löschen, die auf Ihre beendete Umgebung verweisen, um zu verhindern, dass ein DNS-Eintrag hängenbleibt. Ein hängender DNS-Eintrag kann den Internetdatenverkehr, der für Ihre Domain bestimmt ist, Sicherheitslücken aussetzen. Er kann auch andere Risiken bergen.
> Weitere Informationen finden Sie unter Schutz vor hängenden Delegierungsdatensätzen in Route 53 im Entwicklerhandbuch von Amazon Route 53. Weitere Informationen über fehlerhafte DNS-Einträge finden Sie auch im AWS Sicherheits-Blog unter Enhanced Domain Protections for Amazon CloudFront Requests.

Elastic-Beanstalk-Konsole

So beenden Sie eine Umgebung

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

   > **Note**
   >
   > Wenn Sie Ihre Umgebung beenden, wird der CNAME, der der beendeten Umgebung zugeordnet ist, freigegeben und kann von jedem verwendet werden.

   Es dauert einige Minuten, bis Elastic Beanstalk die AWS Ressourcen beendet, die in der Umgebung ausgeführt werden.

AWS CLI

So beenden Sie eine Umgebung

- Führen Sie den folgenden Befehl aus.

```
$ aws elasticbeanstalk terminate-environment --environment-name my-env
```

API

So beenden Sie eine Umgebung

- Rufen Sie `TerminateEnvironment` mit folgendem Parameter auf:

  `EnvironmentName` = `SampleAppEnv`

```
https://elasticbeanstalk.us-west-2.amazon.com/?EnvironmentName=SampleAppEnv
&Operation=TerminateEnvironment
&AuthParams
```

Elastic Beanstalk Beanstalk-Umgebungen mit der CLI erstellen AWS

Einzelheiten zu den AWS CLI Befehlen für Elastic Beanstalk finden Sie in der AWS CLI Befehlsreferenz.

1. Überprüfen Sie, ob der CNAME für die Umgebung verfügbar ist.

```
$ aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
{
    "Available": true,
    "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

2. Stellen Sie sicher, dass die Anwendungsversion vorhanden ist.

```
$ aws elasticbeanstalk describe-application-versions --application-name my-app --
version-label v1
```

Wenn Sie noch nicht über eine Anwendungsversion Ihrer Quelle verfügen, erstellen Sie sie. Der folgende Befehl erstellt beispielsweise eine Anwendungsversion aus einem Quell-Bundle im Amazon Simple Storage Service (Amazon S3).

```
$ aws elasticbeanstalk create-application-version --application-name my-app --
version-label v1 --source-bundle S3Bucket=amzn-s3-demo-bucket,S3Key=my-source-
bundle.zip
```

3. Erstellen Sie eine Konfigurationsvorlage für die Anwendung.

```
$ aws elasticbeanstalk create-configuration-template --application-name my-app --
template-name v1 --solution-stack-name "64bit Amazon Linux 2015.03 v2.0.0 running
 Ruby 2.2 (Passenger Standalone)"
```

4. Erstellen Sie die Umgebung.

```
$ aws elasticbeanstalk create-environment --cname-prefix my-cname --application-
name my-app --template-name v1 --version-label v1 --environment-name v1clone --
option-settings file://options.txt
```

Optionseinstellungen sind in der Datei options.txt definiert:

```
[
    {
        "Namespace": "aws:autoscaling:launchconfiguration",
        "OptionName": "IamInstanceProfile",
        "Value": "aws-elasticbeanstalk-ec2-role"
    }
]
```

Die oben genannte Optionseinstellung definiert das IAM-Instance-Profil. Sie können die ARN oder den Profilnamen angeben.

5. Stellen Sie fest, ob die neue Umgebung grün und einsatzbereit ist.

```
$ aws elasticbeanstalk describe-environments --environment-names my-env
```

Wenn die neue Umgebung nicht grün und einsatzbereit ist, sollten Sie entscheiden, ob Sie den Vorgang wiederholen oder die Umgebung im aktuellen Zustand zur Untersuchung lassen

möchten. Achten Sie darauf, die Umgebung zu beenden, nachdem Sie fertig sind, und nicht verwendete Ressourcen zu bereinigen.

> **Note**
>
> Sie können die Zeitüberschreitung anpassen, wenn die Umgebung nicht in einem angemessenen Zeitraum startet.

Elastic Beanstalk-Umgebungen mit der API erstellen

1. Rufen Sie `CheckDNSAvailability` mit folgendem Parameter auf:

   - `CNAMEPrefix` = `SampleApp`

   Example

   ```
   https://elasticbeanstalk.us-east-2.amazonaws.com/?CNAMEPrefix=sampleapplication
   &Operation=CheckDNSAvailability
   &AuthParams
   ```

2. Rufen Sie `DescribeApplicationVersions` mit folgenden Parametern auf:

   - `ApplicationName` = `SampleApp`
   - `VersionLabel` = `Version2`

   Example

   ```
   https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
   &VersionLabel=Version2
   &Operation=DescribeApplicationVersions
   &AuthParams
   ```

3. Rufen Sie `CreateConfigurationTemplate` mit folgenden Parametern auf:

   - `ApplicationName` = `SampleApp`
   - `TemplateName` = `MyConfigTemplate`

- SolutionStackName = 64bit%20Amazon%20Linux%202015.03%20v2.0.0%20running%20Ruby%202.2%20(Passenger%20Standalone)

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&TemplateName=MyConfigTemplate
&Operation=CreateConfigurationTemplate
&SolutionStackName=64bit%20Amazon%20Linux%202015.03%20v2.0.0%20running%20Ruby
%202.2%20(Passenger%20Standalone)
&AuthParams
```

4. Rufen Sie CreateEnvironment mit einem der folgenden Parametersätze auf.

   a. Verwenden Sie Folgendes für eine Webserverumgebungsebene:

   - EnvironmentName = SampleAppEnv2
   - VersionLabel = Version2
   - Description = description
   - TemplateName = MyConfigTemplate
   - ApplicationName = SampleApp
   - CNAMEPrefix = sampleapplication
   - OptionSettings.member.1.Namespace = aws:autoscaling:launchconfiguration
   - OptionSettings.member.1.OptionName = IamInstanceProfile
   - OptionSettings.member.1.Value = aws-elasticbeanstalk-ec2-role

   Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&EnvironmentName=SampleAppEnv2
&TemplateName=MyConfigTemplate
&CNAMEPrefix=sampleapplication
&Description=description
&Operation=CreateEnvironment
&OptionSettings.member.1.Namespace=aws%3Aautoscaling%3Alaunchconfiguration
```

```
&OptionSettings.member.1.OptionName=IamInstanceProfile
&OptionSettings.member.1.Value=aws-elasticbeanstalk-ec2-role
&AuthParams
```

b. Verwenden Sie Folgendes für eine Worker-Umgebungsebene:

- EnvironmentName = SampleAppEnv2
- VersionLabel = Version2
- Description = description
- TemplateName = MyConfigTemplate
- ApplicationName = SampleApp
- Tier = Worker
- OptionSettings.member.1.Namespace = aws:autoscaling:launchconfiguration
- OptionSettings.member.1.OptionName = IamInstanceProfile
- OptionSettings.member.1.Value = aws-elasticbeanstalk-ec2-role
- OptionSettings.member.2.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.2.OptionName = WorkerQueueURL
- OptionSettings.member.2.Value = sqsd.elasticbeanstalk.us-east-2.amazonaws.com
- OptionSettings.member.3.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.3.OptionName = HttpPath
- OptionSettings.member.3.Value = /
- OptionSettings.member.4.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.4.OptionName = MimeType
- OptionSettings.member.4.Value = application/json
- OptionSettings.member.5.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.5.OptionName = HttpConnections
- OptionSettings.member.5.Value = 75
- OptionSettings.member.6.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.6.OptionName = ConnectTimeout
- OptionSettings.member.6.Value = 10
- OptionSettings.member.7.Namespace = aws:elasticbeanstalk:sqsd

- OptionSettings.member.7.OptionName = InactivityTimeout
- OptionSettings.member.7.Value = 10
- OptionSettings.member.8.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.8.OptionName = VisibilityTimeout
- OptionSettings.member.8.Value = 60
- OptionSettings.member.9.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.9.OptionName = RetentionPeriod
- OptionSettings.member.9.Value = 345600

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&EnvironmentName=SampleAppEnv2
&TemplateName=MyConfigTemplate
&Description=description
&Tier=Worker
&Operation=CreateEnvironment
&OptionSettings.member.1.Namespace=aws%3Aautoscaling%3Alaunchconfiguration
&OptionSettings.member.1.OptionName=IamInstanceProfile
&OptionSettings.member.1.Value=aws-elasticbeanstalk-ec2-role
&OptionSettings.member.2.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.2.OptionName=WorkerQueueURL
&OptionSettings.member.2.Value=sqsd.elasticbeanstalk.us-east-2.amazonaws.com
&OptionSettings.member.3.Namespace=aws%3elasticbeanstalk%3sqsd
&OptionSettings.member.3.OptionName=HttpPath
&OptionSettings.member.3.Value=%2F
&OptionSettings.member.4.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.4.OptionName=MimeType
&OptionSettings.member.4.Value=application%2Fjson
&OptionSettings.member.5.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.5.OptionName=HttpConnections
&OptionSettings.member.5.Value=75
&OptionSettings.member.6.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.6.OptionName=ConnectTimeout
&OptionSettings.member.6.Value=10
&OptionSettings.member.7.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.7.OptionName=InactivityTimeout
&OptionSettings.member.7.Value=10
```

```
&OptionSettings.member.8.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.8.OptionName=VisibilityTimeout
&OptionSettings.member.8.Value=60
&OptionSettings.member.9.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.9.OptionName=RetentionPeriod
&OptionSettings.member.9.Value=345600
&AuthParams
```

Erstellen einer Launch-Now-URL

Sie können eine benutzerdefinierte URL erstellen, damit jeder schnell eine vordefinierte Webanwendung in AWS Elastic Beanstalk bereitstellen und ausführen kann. Diese URL wird als Launch-Now-URL bezeichnet. Mit einer Launch-Now-URL können Sie zum Beispiel eine Webanwendung demonstrieren, die zur Ausführung auf Elastic Beanstalk erstellt wurde. Mit einer Launch-Now-URL können Sie Parameter verwenden, um die erforderlichen Informationen zum Assistenten zum Erstellen von Anwendungen im Vorfeld hinzuzufügen. Nachdem Sie dem Assistenten diese Informationen hinzugefügt haben, kann jeder mit dem URL-Link in nur wenigen Schritten eine Elastic-Beanstalk-Umgebung mit Ihrer Webanwendungsquelle starten. Das bedeutet, dass Benutzer den Speicherort des Anwendungs-Quell-Bundles nicht manuell hochladen oder angeben müssen. Sie müssen dem Assistenten auch keine zusätzlichen Informationen bereitstellen.

Eine Launch-Now-URL liefert Elastic Beanstalk die minimalen Informationen, die erforderlich sind, um eine Anwendung zu erstellen: Anwendungsnamen, Lösungs-Stack, Instance-Typ und Umgebungstyp. Elastic Beanstalk nutzt Standardwerte für andere Konfigurationsdetails, die nicht explizit in Ihrer benutzerdefinierten Launch-Now-URL angegeben wurden.

Eine Launch-Now-URL verwendet die Standard-URL-Syntax. Weitere Informationen finden Sie unter RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax.

URL-Parameter

Die URL muss die folgenden Parameter enthalten, bei denen die Groß- und Kleinschreibung beachtet werden muss:

- Region — Geben Sie eine AWS Region an. Eine Liste der Regionen, die von Elastic Beanstalk unterstützt werden, finden Sie unter AWS Elastic Beanstalk -Endpunkte und Kontingente in der Allgemeine AWS-Referenz.
- applicationName – Geben Sie den Namen Ihrer Anwendung an. Elastic Beanstalk zeigt den Anwendungsnamen in der Elastic Beanstalk-Konsole an, um sie von anderen Anwendungen

zu unterscheiden. Standardmäßig bildet der Name der Anwendung auch die Basis für den Umgebungsnamen und die Umgebungs-URL.

- platform (Plattform) – Geben Sie die Plattformversion für die Umgebung an. Nutzen Sie dazu eine der folgenden Methoden und codieren Sie Ihre Auswahl dann in einer URL:
  - Geben Sie einen Plattform-ARN ohne Version an. Elastic Beanstalk wählt die neueste Plattformversion der entsprechenden Plattform-Hauptversion aus. Um beispielsweise die neueste Python-3.6-Plattformversion auszuwählen, geben Sie Folgendes an `Python 3.6 running on 64bit Amazon Linux`.
  - Geben Sie den Plattformnamen an. Elastic Beanstalk wählt die neueste Plattformversion der neuesten Sprach-Laufzeitumgebung der Plattform (zum Beispiel, `Python`) aus.

  Eine Beschreibung aller verfügbaren Plattformen und deren Versionen finden Sie unter Von Elastic Beanstalk unterstützte Plattformen.

  Sie können das AWS Command Line Interface(AWS CLI) verwenden, um eine Liste aller verfügbaren Plattformversionen mit ihren jeweiligen Versionen zu erhalten ARNs. Der `list-platform-versions`-Befehl listet detaillierte Informationen über alle verfügbaren Plattformversionen auf. Verwenden Sie das `--filters`-Argument, um die Liste nach unten zu verschieben. Sie können die Liste beispielsweise so eingrenzen, dass nur die Plattformversionen einer bestimmten Sprache angezeigt werden.

  Das folgende Beispiel fragt alle Python-Plattformversionen ab und übergibt die Ausgabe über eine Reihe von Befehlen. Das Ergebnis ist eine Liste der Plattformversionen ARNs (ohne den */*_version_ Schwanz) in einem für Menschen lesbaren Format ohne URL-Kodierung.

```
$ aws elasticbeanstalk list-platform-versions --filters
 'Type="PlatformName",Operator="contains",Values="Python"' | grep PlatformArn | awk -
F '"' '{print $4}' | awk -F '/' '{print $2}'
Preconfigured Docker - Python 3.4 running on 64bit Debian
Preconfigured Docker - Python 3.4 running on 64bit Debian
Python 2.6 running on 32bit Amazon Linux
Python 2.6 running on 32bit Amazon Linux 2014.03
...
Python 3.6 running on 64bit Amazon Linux
```

  Das folgenden Beispiel fügt dem letzten Beispiel einen Perl-Befehl hinzu, um die Ausgabe als URL zu codieren.

```
$ aws elasticbeanstalk list-platform-versions --filters
 'Type="PlatformName",Operator="contains",Values="Python"' | grep PlatformArn | awk
 -F '"' '{print $4}' | awk -F '/' '{print $2}' | perl -MURI::Escape -ne 'chomp;print
 uri_escape($_),"\n"'
Preconfigured%20Docker%20-%20Python%203.4%20running%20on%2064bit%20Debian
Preconfigured%20Docker%20-%20Python%203.4%20running%20on%2064bit%20Debian
Python%202.6%20running%20on%2032bit%20Amazon%20Linux
Python%202.6%20running%20on%2032bit%20Amazon%20Linux%202014.03
...
Python%203.6%20running%20on%2064bit%20Amazon%20Linux
```

Eine Launch-Now-URL kann optional die folgenden Parameter enthalten. Wenn Sie die optionalen Parameter nicht in Ihre Launch-Now-URL einschließen, verwendet Elastic Beanstalk Standardwerte zum Erstellen und Ausführen Ihrer Anwendung. Wenn Sie den sourceBundleUrlParameter nicht angeben, verwendet Elastic Beanstalk die Standard-Beispielanwendung für die angegebene Plattform.

- sourceBundleUrl— Geben Sie den Speicherort Ihres Quellpakets für Webanwendungen im URL-Format an. Wenn Sie beispielsweise Ihr Quellpaket in einen Amazon S3 S3-Bucket hochgeladen haben, können Sie den Wert des sourceBundleUrlParameters als angeben`https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject`.

  > **Note**
  >
  > Sie können den Wert des sourceBundleUrlParameters als HTTP-URL angeben, aber der Webbrowser des Benutzers konvertiert die Zeichen nach Bedarf, indem er die HTML-URL-Kodierung anwendet.

- environmentType – Gibt an, ob es sich um eine skalierbare Umgebung mit Lastausgleich oder um eine Single-Instance-Umgebung handelt. Weitere Informationen finden Sie unter Typen von Umgebungen. Sie können entweder `LoadBalancing` oder `SingleInstance` als Parameterwert angeben.
- tierName – Geben Sie an, ob die Umgebung eine Webanwendung unterstützt, die Webanfragen oder eine Webanwendung unterstützt, die Hintergrundaufträge ausführt. Weitere Informationen finden Sie unter Elastic-Beanstalk-Worker-Umgebungen. Sie können entweder `WebServer` oder `Worker` angeben.

- instanceType – Geben Sie einen Server mit den Merkmalen (einschließlich Speichergröße und CPU-Leistung) an, die für Ihre Anwendung am besten geeignet sind. Weitere Informationen zu EC2 Amazon-Instance-Familien und -Typen finden Sie unter Instance-Typen im EC2 Amazon-Benutzerhandbuch. Weitere Informationen zu den verfügbaren Instance-Typen in allen Regionen finden Sie unter Verfügbare Instance-Typen im EC2 Amazon-Benutzerhandbuch.
- withVpc – Geben Sie an, ob die Umgebung in einer Amazon VPC erstellt werden soll. Sie können entweder `true` oder `false` angeben. Weitere Informationen zum Verwenden von Elastic Beanstalk mit Amazon VPC finden Sie unter Verwenden von Elastic Beanstalk mit Amazon VPC.
- withRds – Geben Sie an, ob eine Amazon RDS-Datenbank-Instance mit dieser Umgebung erstellt werden soll. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon RDS. Sie können entweder `true` oder `false` angeben.
- rds DBEngine — Geben Sie die Datenbank-Engine an, die Sie für Ihre EC2 Amazon-Instances in dieser Umgebung verwenden möchten. Sie können `mysql`, `oracle-sel`, `sqlserver-ex`, `sqlserver-web` oder `sqlserver-se`festlegen. Der Standardwert ist `mysql`.
- rds DBAllocated Storage — Geben Sie die zugewiesene Datenbankspeichergröße in Gigabyte (GB) an. Sie können die folgenden Werte angeben:
  - MySQL – `5` bis `1024`. Der Standardwert ist `5`.
  - Oracle – `10` bis `1024`. Der Standardwert ist `10`.
  - Microsoft SQL Server Express Edition – `30`.
  - Microsoft SQL Server Web Edition – `30`.
  - Microsoft SQL Server Standard Edition – `200`.
- rds DBInstance Class — Geben Sie den Typ der Datenbank-Instance an. Der Standardwert ist `db.t2.micro` (`db.m1.large` für eine Umgebung, die nicht in einer Amazon VPC ausgeführt wird). Eine Liste der von Amazon RDS unterstützten Datenbank-Instance-Klassen finden Sie unter DB-Instance-Klasse im Benutzerhandbuch für Amazon Relational Database Service.
- RDSMulti AZDatabase — Geben Sie an, ob Elastic Beanstalk die Datenbank-Instance in mehreren Availability Zones erstellen muss. Sie können entweder `true` oder `false` angeben. Weitere Informationen zu Multi-AZ-Bereitstellungen mit Amazon Relational Database Service (RDS), finden Sie unter Regionen und Availability Zones im Benutzerhandbuch für Amazon Relational Database Service.
- DBDeletionrds-Richtlinie — Geben Sie an, ob die Datenbank-Instance bei Beendigung der Umgebung gelöscht oder ein Snapshot erstellt werden soll. Sie können entweder `Delete` oder `Snapshot` angeben.

Beispiel

Es folgt ein Beispiel für eine Launch-Now-URL. Nachdem Sie Ihre eigene erstellt haben, können Sie sie an Ihre Benutzer weitergeben. Sie können z. B. die URL auf einer Webseite oder in Schulungsmaterial einbetten. Wenn Benutzer eine Anwendung mit der Launch-Now-URL erstellen, benötigt der Elastic Beanstalk-Assistent zum Erstellen einer Anwendung keine zusätzlichen Eingaben.

```
https://console.aws.amazon.com/elasticbeanstalk/home?region=us-west-2#/newApplication?
applicationName=YourCompanySampleApp
      &platform=PHP%207.3%20running%20on%2064bit%20Amazon%20Linux&sourceBundleUrl=
        http://s3.amazonaws.com/amzn-s3-demo-bucket/
myobject&environmentType=SingleInstance&tierName=WebServer
      &instanceType=m1.small&withVpc=true&withRds=true&rdsDBEngine=

 postgres&rdsDBAllocatedStorage=6&rdsDBInstanceClass=db.m1.small&rdsMultiAZDatabase=
          true&rdsDBDeletionPolicy=Snapshot
```

So verwenden Sie die Launch-Now-URL

1. Wählen Sie die Launch-Now-URL aus.
2. Wenn die Elastic-Beanstalk-Konsole geöffnet wird, wählen Sie auf der Seite Create a web app (Webanwendung erstellen) die Option Review and launch (Prüfen und starten) aus, um die Einstellungen anzuzeigen, die Elastic Beanstalk zum Erstellen der Anwendung und Starten der Umgebung für die Anwendungsausführung verwendet.
3. Wählen Sie auf der Seite Configure (Konfigurieren) die Option Create app (Anwendung erstellen) aus, um die Anwendung zu erstellen.

Gruppen von Elastic Beanstalk Umgebungen erstellen oder aktualisieren

Mit der AWS Elastic Beanstalk ComposeEnvironmentsAPI können Sie Gruppen von Elastic Beanstalk Beanstalk-Umgebungen in einer einzigen Anwendung erstellen und aktualisieren. Jede Umgebung in der Gruppe kann eine separate Komponente einer serviceorientierten Architekturanwendung ausführen. Die `Compose Environments`-API nimmt eine Liste der Anwendungsversionen und einen optionalen Gruppennamen. Elastic Beanstalk erstellt eine Umgebung für jede Anwendungsversion oder, wenn die Umgebungen bereits vorhanden ist, stellt die Anwendungsversionen darin bereit.

Erstellen Sie Links zwischen Elastic Beanstalk-Umgebungen zur Benennung einer Umgebung in Abhängigkeit von einer anderen. Beim Erstellen einer Gruppe von Umgebungen mit der `Compose Environments`-API erstellt Elastic Beanstalk abhängige Umgebungen nur, nachdem deren Abhängigkeiten aktiv sind. Weitere Informationen zu Umgebungs-Links finden Sie unter Verknüpfungen zwischen Elastic Beanstalk-Umgebungen erstellen.

Die `Compose Environments`-API nutzt ein Umgebungsmanifest zum Speichern von Konfigurationsdetails, die von Gruppen von Umgebungen gemeinsam verwendet werden. Jede Komponentenanwendung muss über eine `env.yaml`-Konfigurationsdatei im Quell-Bundle der Anwendung verfügen, in der die Parameter angegeben werden, mit denen die Umgebung erstellt wird.

`Compose Environments` erfordert, dass `EnvironmentName` und `SolutionStack` im Umgebungsmanifest für jede Komponentenanwendung angegeben werden.

Sie können die `Compose Environments` API mit der Elastic Beanstalk Beanstalk-Befehlszeilenschnittstelle (EB CLI) AWS CLI, dem oder einem SDK verwenden. EB CLI-Anweisungen finden Sie unter Verwalten mehrerer Elastic Beanstalk-Umgebungen als Gruppe mit der EB CLI.

Verwenden der **`Compose Environments`**-API

Beispielsweise könnten Sie eine Anwendung namens `Media Library` erstellen, mit der Benutzer Bilder und Videos hochladen und verwalten können, die in Amazon Simple Storage Service (Amazon S3) gespeichert sind. Die Anwendung verfügt über eine Frontend-Umgebung, `front`, für die Ausführung einer Webanwendung, mit der Benutzer einzelne Dateien hoch- und herunterladen, ihre Bibliothek anzeigen Batch-Verarbeitungsjobs initiieren können.

Anstatt die Jobs direkt zu verarbeiten, fügt die Frontend-Anwendung Jobs zu einer Amazon SQS-Warteschlange hinzu. Die zweite Umgebung, `worker` zieht Jobs aus der Warteschlange und verarbeitet sie. `worker` verwendet einen G2-Instance-Typ, der eine Hochleistungs-GPU hat, während `front` auf einem kosteneffizienteren, allgemeinen Instance-Type ausgeführt werden kann.

Sie organisieren den Projektordner `Media Library` in separate Verzeichnisse für jede Komponente, wobei jedes Verzeichnis eine Umgebungs-Definitionsdatei (`env.yaml`) mit den Quellcode enthält:

```
~/workspace/media-library
|-- front
|   `-- env.yaml
```

```
`-- worker
    `-- env.yaml
```

Die folgenden Auflistung zeigt die `env.yaml`-Datei für jede Komponentenanwendung.

~/workspace/media-library/front/env.yaml

```
EnvironmentName: front+
EnvironmentLinks:
  "WORKERQUEUE" : "worker+"
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentTier:
  Name: WebServer
  Type: Standard
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.4 running Java 8
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: m4.large
```

~/workspace/media-library/worker/env.yaml

```
EnvironmentName: worker+
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentTier:
  Name: Worker
  Type: SQS/HTTP
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.4 running Java 8
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: g2.2xlarge
```

Nach dem Erstellen einer Anwendungsversion für die Frontend- (`front-v1`) und Worker- (`worker-v1`) Anwendungskomponenten, die Sie mit der `Compose Environments`-API mit den Versionsnamen erstellen. In diesem Beispiel verwenden wir die AWS CLI zum Aufrufen der API.

```
# Create application versions for each component:
~$ aws elasticbeanstalk create-application-version --application-name media-
library --version-label front-v1 --process --source-bundle S3Bucket="amzn-s3-demo-
bucket",S3Key="front-v1.zip"
  {
    "ApplicationVersion": {
```

```
        "ApplicationName": "media-library",
        "VersionLabel": "front-v1",
        "Description": "",
        "DateCreated": "2015-11-03T23:01:25.412Z",
        "DateUpdated": "2015-11-03T23:01:25.412Z",
        "SourceBundle": {
            "S3Bucket": "amzn-s3-demo-bucket",
            "S3Key": "front-v1.zip"
        }
    }
  }
~$ aws elasticbeanstalk create-application-version --application-name media-library
 --version-label worker-v1 --process --source-bundle S3Bucket="amzn-s3-demo-
bucket",S3Key="worker-v1.zip"
  {
    "ApplicationVersion": {
        "ApplicationName": "media-library",
        "VersionLabel": "worker-v1",
        "Description": "",
        "DateCreated": "2015-11-03T23:01:48.151Z",
        "DateUpdated": "2015-11-03T23:01:48.151Z",
        "SourceBundle": {
            "S3Bucket": "amzn-s3-demo-bucket",
            "S3Key": "worker-v1.zip"
        }
    }
  }
# Create environments:
~$ aws elasticbeanstalk compose-environments --application-name media-library --group-
name dev --version-labels front-v1 worker-v1
```

Die dritte Anfrage erstellt zwei Umgebungen, `front-dev` und `worker-dev`. Die API erstellt die Namen der Umgebungen, indem der `EnvironmentName`, der in der `env.yaml`-Datei angegeben ist, mit der `group name`-Option im `Compose Environments`-Aufruf angegeben wird, getrennt durch einen Bindestrich. Die Gesamtlänge dieser beiden Optionen und der Bindestrich dürfen nicht größer als die maximal zulässige Umgebungsnamenlänge von 23 Zeichen.

Die Anwendung in der `front-dev`-Umgebung kann auf den Namen der Amazon SQS-Warteschlange zugreifen, die der `worker-dev`-Umgebung angefügt ist, indem die `WORKERQUEUE`-Variable gelesen wird. Weitere Informationen zu Umgebungs-Links finden Sie unter Verknüpfungen zwischen Elastic Beanstalk-Umgebungen erstellen.

Verwalten mehrerer Elastic Beanstalk-Umgebungen als Gruppe mit der EB CLI

Sie können die EB-CLI verwenden, um Gruppen von AWS Elastic Beanstalk Umgebungen zu erstellen, in denen jeweils eine separate Komponente einer serviceorientierten Architekturanwendung ausgeführt wird. Die EB CLI verwaltet solche Gruppen mithilfe der ComposeEnvironmentsAPI.

> **Note**
>
> Umgebungsgruppen unterscheiden sich von dem Konzept mehrerer Container in einer Multicontainer Docker-Umgebung. Mit Umgebungsgruppen wird jede Komponente Ihrer Anwendung in einer separaten Elastic Beanstalk-Umgebung ausgeführt, mit jeweils einem eigenen Satz von Amazon EC2-Instances. Jede Komponente kann separat skaliert werden. Mit Multicontainer Docker kombinieren Sie mehrere Komponenten einer Anwendung in einer einzigen Umgebung. Alle Komponenten verwenden den gleichen Satz an Amazon EC2-Instances, wobei jede Instance mehrere Docker-Container ausführt. Wählen Sie eine dieser Architekturen entsprechend den Anforderungen Ihrer Anwendung aus.
> Details zu Multicontainer Docker finden Sie unter Verwendung des von ECS verwalteten Docker-Plattformzweigs in Elastic Beanstalk.

Organisieren Sie Ihre Anwendungskomponenten in der folgenden Ordnerstruktur:

```
~/project-name
|-- component-a
|   `-- env.yaml
`-- component-b
    `-- env.yaml
```

Jeder Unterordner enthält den Quellcode für eine unabhängige Komponente einer Anwendung, die in ihrer eigenen Umgebung ausgeführt, sowie eine Umgebungsdefinitionsdatei mit dem Namen `env.yaml`. Weitere Informationen zum Format `env.yaml` finden Sie unter Umgebungsmanifest (`env.yaml`).

Zur Verwendung der `Compose Environments`-API führen Sie zuerst eb init aus dem Projektordner aus und geben Sie die einzelnen Komponenten mit dem Namen des Ordners an, der sie mit der `--modules`-Option enthält:

```
~/workspace/project-name$ eb init --modules component-a component-b
```

Sie werden von der EB CLI aufgefordert, jede Komponente zu konfigurieren. Anschließend wird das `.elasticbeanstalk`-Verzeichnis in jedem Komponentenordner erstellt. EB CLI erstellt keine Konfigurationsdateien im übergeordneten Verzeichnis.

```
~/project-name
|-- component-a
|   |-- .elasticbeanstalk
|   `-- env.yaml
`-- component-b
    |-- .elasticbeanstalk
    `-- env.yaml
```

Führen Sie im nächsten Schritt den eb create-Befehl mit einer Liste von zu erstellenden Umgebungen aus, einen für jede Komponente:

```
~/workspace/project-name$ eb create --modules component-a component-b --env-group-
suffix group-name
```

Mit diesem Befehl wird eine Umgebung für jede Komponente erstellt. Die Namen der Umgebungen werden erstellt, indem der `EnvironmentName`, der in der `env.yaml`-Datei angegeben ist, mit dem Gruppennamen zusammengefügt wird (durch einen Bindestrich getrennt). Die Gesamtlänge dieser beiden Optionen und der Bindestrich dürfen nicht größer als die maximal zulässige Umgebungsnamenlänge von 23 Zeichen.

Verwenden Sie den eb deploy-Befehl, um die Umgebung zu aktualisieren:

```
~/workspace/project-name$ eb deploy --modules component-a component-b
```

Sie können jede Komponente einzeln oder als Gruppe aktualisieren. Geben Sie die Komponenten an, die Sie mit der `--modules`-Option aktualisieren möchten.

Die EB CLI speichert den Gruppennamen, den Sie mit eb create im `branch-defaults`-Abschnitt der EB CLI-Konfigurationsdatei unter `/.elasticbeanstalk/config.yml` verwendet haben. Zum Bereitstellen Ihrer Anwendung für eine andere Gruppe verwenden Sie die `--env-group-suffix`-Option, wenn Sie eb deploy ausführen. Wenn die Gruppe noch nicht vorhanden ist, erstellt die EB CLI eine neue Gruppe von Umgebungen:

```
~/workspace/project-name$ eb deploy --modules component-a component-b --env-group-suffix group-2-name
```

Zum Beenden der Umgebungen führen Sie eb terminate im Ordner für jedes Modul aus. Standardmäßig zeigt die EB CLI eine Fehlermeldung an, wenn Sie versuchen, eine Umgebung zu beenden, von der eine andere laufende Umgebung abhängt. Beenden Sie zuerst die abhängige Umgebung oder überschreiben Sie mit der `--ignore-links`-Option das Standardverhalten:

```
~/workspace/project-name/component-b$ eb terminate --ignore-links
```

Anwendungen in Elastic Beanstalk-Umgebungen bereitstellen

Sie können die AWS Elastic Beanstalk Konsole verwenden, um ein aktualisiertes Quellpaket hochzuladen und es in Ihrer Elastic Beanstalk Beanstalk-Umgebung bereitzustellen, oder Sie können eine zuvor hochgeladene Version erneut bereitstellen.

Jede Bereitstellung wird mit einer Bereitstellungs-ID identifiziert. Die Bereitstellung IDs beginnt bei jeder Änderung der Bereitstellung 1 und der Instanzkonfiguration und wird um eins erhöht. Wenn Sie erweiterte Zustandsberichte aktiviert haben, zeigt Elastic Beanstalk die Bereitstellungs-ID in der Zustandskonsole und in der EB CLI an, wenn der Instance-Zustand gemeldet wird. Mit der Bereitstellungs-ID können Sie den Zustand Ihrer Umgebung bestimmen, wenn ein fortlaufendes Update fehlschlägt.

Elastic Beanstalk stellt mehrere Bereitstellungsrichtlinien und -einstellungen zur Verfügung. Weitere Informationen zum Konfigurieren einer Richtlinie und zusätzlicher Einstellungen finden Sie unter the section called “Optionen für die Bereitstellung”. In der folgenden Tabelle sind die Richtlinien und die Arten von Umgebungen aufgeführt, die sie unterstützen.

Unterstützte Bereitstellungsrichtlinien

Bereitstellungsrichtlinie	Umgebungen mit Load Balancing	Umgebungen mit einer Instance	Ältere Windows Server-Umgebungen†
Alle auf einmal	✓ Ja	✓ Ja	✓ Ja
Fortlaufend	✓ Ja	✗ Nein	✓ Ja
Fortlaufend mit zusätzlichem Stapel	✓ Ja	✗ Nein	✗ Nein

Bereitstellungsrichtlinie	Umgebungen mit Load Balancing	Umgebungen mit einer Instance	Ältere Windows Server-Umgebungen†
Unveränderlich	✓ Ja	✓ Ja	✗ Nein
Datenverkehrsaufteilung	✓ Ja (Application Load Balancer)	✗ Nein	✗ Nein

† In dieser Tabelle ist eine Legacy-Windows Server-Umgebung eine Umgebung, die auf einer Windows Server-Plattformkonfiguration mit einer früheren IIS-Version als IIS 8.5 basiert.

> ⚠ Warning
>
> Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle gesammelten Amazon EC2 Burst-Guthaben verloren. Dies geschieht in den folgenden Fällen:
>
> - Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
> - Unveränderliche Updates
> - Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Auswählen einer Bereitstellungsrichtlinie

Die Auswahl der richtigen Bereitstellungsrichtlinie für Ihre Anwendung ist eine Abstimmung einiger Überlegungen und hängt von Ihren speziellen Anforderungen ab. Die Seite the section called “Optionen für die Bereitstellung” enthält weitere Informationen zu den einzelnen Richtlinien und eine detaillierte Beschreibung der Funktionsweise einiger von ihnen.

Die folgende Liste bietet zusammenfassende Informationen zu den verschiedenen Bereitstellungsrichtlinien und zugehörige Überlegungen.

- Alles auf einmal – Die schnellste Bereitstellungsmethode. Geeignet, wenn Sie einen kurzen Serviceverlust akzeptieren können und eine schnelle Bereitstellung für Sie wichtig ist. Mit dieser Methode stellt Elastic Beanstalk die neue Anwendungsversion für jede Instance bereit. Anschließend muss der Webproxy oder der Anwendungsserver möglicherweise neu gestartet werden. Infolgedessen könnte Ihre Anwendung für kurze Zeit für Benutzer nicht verfügbar sein (oder eine geringe Verfügbarkeit haben).

- Fortlaufend – Vermeidet Ausfallzeiten und minimiert Einschränkungen bei der Verfügbarkeit auf Kosten einer längeren Bereitstellungszeit. Geeignet, wenn Sie keinerlei vollständigen Serviceverlust akzeptieren können. Mit dieser Methode wird Ihre Anwendung in Ihrer Umgebung in je einem Stapel von Instances bereitgestellt. Während der gesamten Bereitstellung wird die meiste Bandbreite beibehalten.
- Fortlaufend mit zusätzlichem Stapel – Vermeidet jegliche Verfügbarkeitseinschränkung auf Kosten einer noch längeren Bereitstellungszeit als bei der Methode Fortlaufend. Geeignet, wenn Sie während der gesamten Bereitstellung die gleiche Bandbreite beibehalten müssen. Mit dieser Methode startet Elastic Beanstalk einen zusätzlichen Stapel von Instances und führt dann eine fortlaufende Bereitstellung durch. Das Starten des zusätzlichen Stapels erfordert Zeit und stellt sicher, dass die gleiche Bandbreite während der gesamten Bereitstellung beibehalten wird.
- Unveränderlich – Eine langsamere Bereitstellungsmethode, mit der Ihre neue Anwendungsversion immer auf neuen Instances bereitgestellt wird, anstatt vorhandene Instances zu aktualisieren. Sie bietet darüber hinaus den Vorteil eines schnellen und sicheren Rollbacks für den Fall, dass die Bereitstellung fehlschlägt. Mit dieser Methode führt Elastic Beanstalk ein unveränderliches Update durch, um Ihre Anwendung bereitzustellen. In einem unveränderbaren Update wird eine zweite Auto Scaling-Gruppe in Ihrer Umgebung gestartet und die neue Version verarbeitet Datenverkehr zusammen mit der alten Version, bis die neue Instance die Zustandsprüfungen besteht.
- Datenverkehrsaufteilung – Eine Bereitstellungsmethode für Canary-Tests. Geeignet, wenn Sie die Integrität Ihrer neuen Anwendungsversion mit einem Teil des eingehenden Datenverkehrs testen möchten, während der Rest des Datenverkehrs, der von der alten Anwendungsversion bedient wird, beibehalten werden soll.

In der folgenden Tabelle werden die Eigenschaften der Bereitstellungsmethoden verglichen.

Bereitstellungsmethoden

Method	Auswirkungen einer fehlgeschlagenen Bereitstellung	Bereitstellungszeit	Ohne Ausfallzeit	Keine DNS-Änderung	Rollback Vorgang	Code bereitgestellt für
Alle auf einmal	Ausfallzeit	🕒	X Nein	✓ Ja	Erneute manuell Bereitstellung	Vorhandene Instances

Method	Auswirkungen einer fehlgeschlagenen Bereitstellung	Bereitstellungszeit	Ohne Ausfallzeit	Keine DNS-Änderung	Rollback Vorgang	Code bereitgestellt für
Fortlaufend	Einzelne Stapel außer Betrieb; alle erfolgreichen Stapel vor dem Ausfall werden mit neuer Anwendungsversion ausgeführt	🕒	✓ Ja	✓ Ja	Erneute manuell Bereitstellung	Vorhandene Instances
Fortlaufend mit zusätzlichem Stapel	Minimal, wenn der erste Stapel fehlschlägt, ansonsten ähnlich wie Rolling (Fortlaufend)	🕒	✓ Ja	✓ Ja	Erneute manuell Bereitstellung	Neue und vorhandene Instances
Unveränderlich	Minimal	🕒	✓ Ja	✓ Ja	Neue Instanc beende	Neue Instances
Datenverkehrsaufteilung	Prozentsatz des Clientdatenverkehrs, der an die vorübergehend beeinträchtigte neue Version weitergeleitet wird	🕒	✓ Ja	✓ Ja	Datenverkehr umleiten und neue Instanc beende	Neue Instances
Blau/Grün	Minimal	🕒	✓ Ja	✗ Nein	URL austausen	Neue Instances

† Variiert je nach Stapelgröße.

†† Variiert je nach Einstellung der Option Auswertungszeit.

Bereitstellen einer neuen Anwendungsversion

Sie können Bereitstellungen über das Dashboard Ihrer Umgebung durchführen.

So stellen Sie eine neue Anwendungsversion in einer Elastic Beanstalk-Umgebung bereit:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Upload and Deploy (Hochladen und Bereitstellen).
4. Verwenden Sie das Formular auf dem Bildschirm, um das Quellpaket der Anwendung hochzuladen.
5. Wählen Sie Bereitstellen.

Erneute Bereitstellung einer früheren Version

Sie können auch eine zuvor hochgeladene Version Ihrer Anwendung in einer der Umgebungen aus der Anwendungsversionsseite bereitstellen.

So stellen Sie eine vorhandene Anwendungsversion in einer vorhandenen Umgebung bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen)aus.
4. Wählen Sie die Anwendungsversion aus, die bereitgestellt werden soll.
5. Wählen Sie Actions (Aktionen) und anschließend Deploy (Bereitstellen).
6. Wählen Sie eine Umgebung und anschließend Deploy (Bereitstellen).

Weitere Möglichkeiten zur Bereitstellung Ihrer Anwendung

Wenn Sie häufig bereitstellen, sollten Sie die Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) zum Verwalten Ihrer Umgebungen verwenden. Die EB-CLI erstellt neben Ihrem Quellcode ein

Repository. Sie kann außerdem ein Quellpaket erstellen, es nach Elastic Beanstalk hochladen und mit einem einzigen Befehl bereitstellen.

Für Bereitstellungen, die von Änderungen der Ressourcenkonfiguration oder einer neuen Version, die nicht neben der alten Version ausgeführt werden kann, abhängen, können Sie eine neue Umgebung mit der neuen Version starten und einen CNAME-Austausch für eine Blau/Grün-Bereitstellung durchführen.

Um Ihre Build-, Test- und Bereitstellungsprozesse zu automatisieren, können Sie Continuous Integration und Continuous Deployment (CI/CD) mit Ihrer Elastic Beanstalk Beanstalk-Umgebung implementieren. Weitere Informationen finden Sie unter Implementierung der CI/CD Integration mit Ihrer Elastic Beanstalk Beanstalk-Umgebung.

Bereitstellungsrichtlinien und -einstellungen

AWS Elastic Beanstalk bietet mehrere Optionen für die Verarbeitung von Bereitstellungen, darunter Bereitstellungsrichtlinien (Alle auf einmal, fortlaufend, fortlaufend mit zusätzlichem Batch, Unveränderlich und Datenverkehrsaufteilung) und Optionen, mit denen Sie die Batchgröße und das Verhalten bei der Integritätsprüfung während Bereitstellungen konfigurieren können. In Ihrer Umgebung werden standardmäßig Bereitstellungen verwendet. all-at-once Wenn Sie die Umgebung über die EB CLI erstellt haben und es sich um eine skalierbare Umgebung handelt (d. h., Sie haben die Option `--single` nicht angegeben), verwendet sie fortlaufende Bereitstellungen.

Bei fortlaufenden Bereitstellungen teilt Elastic Beanstalk die EC2 Amazon-Instances der Umgebung in Batches auf und stellt die neue Version der Anwendung jeweils für einen Stapel bereit. Auf den verbleibenden Instances in der Umgebung wird weiterhin die alte Version der Anwendung ausgeführt. Während der fortlaufenden Bereitstellung werden Anforderungen sowohl von Instances mit der alten Anwendungsversion als auch von Instances mit der neuen Anwendungsversion (in den bereits aktualisierten Stapeln) verarbeitet. Details hierzu finden Sie unter the section called “Funktionsweise fortlaufender Bereitstellungen”.

Um auch während der Bereitstellungen die volle Kapazität zu wahren, können Sie die Umgebung so konfigurieren, dass erst ein neuer Instance-Batch gestartet wird, bevor andere Instances außer Betrieb genommen werden. Diese Option wird als fortlaufende Bereitstellung mit zusätzlichem Batch bezeichnet. Nach erfolgter Bereitstellung beendet Elastic Beanstalk den zusätzlichen Instance-Batch.

Bei unveränderlichen Bereitstellungen wird eine unveränderliche Aktualisierung ausgeführt. Dabei wird zusätzlich zu den Instances mit der alten Anwendungsversion eine ganze Gruppe neuer Instances mit der neuen Anwendungsversion in einer separaten Auto Scaling-Gruppe gestartet.

Die Probleme, die von teilweise abgeschlossenen fortlaufenden Bereitstellungen verursacht werden können, treten bei unveränderlichen Bereitstellungen nicht auf. Falls die neuen Instances die Zustandsprüfungen nicht bestehen, werden sie von Elastic Beanstalk beendet – und die ursprünglichen Instances sind unverändert vorhanden.

Mit Datenverkehrsaufteilungs-Bereitstellungen können Sie Canary-Tests als Teil Ihrer Anwendungsbereitstellung durchführen. In einer Bereitstellung mit Datenverkehrsaufteilung startet Elastic Beanstalk eine vollständige Reihe neuer Instances, genau wie bei einer unveränderlichen Bereitstellung. Anschließend leitet er einen bestimmten Prozentsatz des eingehenden Clientdatenverkehrs für einen bestimmten Evaluierungszeitraum an die neue Anwendungsversion weiter. Wenn die neuen Instances fehlerfrei bleiben, leitet Elastic Beanstalk den gesamten Datenverkehr an sie weiter und beendet die alten Instances. Wenn die neuen Instances keine Zustandsprüfungen bestehen oder wenn Sie die Bereitstellung abbrechen, verschiebt Elastic Beanstalk den Datenverkehr zurück zu den alten Instances und beendet die neuen Instances. Der Service wird nie unterbrochen. Details hierzu finden Sie unter the section called “Funktionsweise von Bereitstellungen mit Datenverkehrsaufteilung”.

> **Warning**
>
> Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle gesammelten Amazon EC2 Burst-Guthaben verloren. Dies geschieht in den folgenden Fällen:
>
> - Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
> - Unveränderliche Updates
> - Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Im Falle, dass die Anwendung nicht alle Zustandsprüfungen besteht, aber mit einem niedrigeren Status ordnungsgemäß funktioniert, können Sie die Option Healthy threshold ändern und so zulassen, dass Instances die Zustandsprüfungen auch mit einem niedrigeren Status (z. B. `Warning`) bestehen. Um bei fehlgeschlagenen Bereitstellungen aufgrund von nicht bestandenen Zustandsprüfungen eine Aktualisierung unabhängig vom Status zu erzwingen, aktivieren Sie die Option Ignore health check (Zustandsprüfung ignorieren).

Sofern Sie eine Batchgröße für fortlaufende Aktualisierungen angeben, verwendet Elastic Beanstalk diesen Wert auch für den fortlaufenden Anwendungsneustart. Verwenden Sie fortlaufende Neustarts,

wenn die Proxy- und Anwendungsserver, die auf den Instances der Umgebung ausgeführt werden, ohne Ausfallzeit neu gestartet werden müssen.

Konfigurieren von Anwendungsbereitstellungen

Aktivieren und konfigurieren Sie in der Environment Management Console die Stapelbereitstellung von Anwendungsversionen, indem Sie die Option Updates and Deployments (Updates und Bereitstellungen) auf der Seite Configuration (Konfiguration) der Umgebung ändern.

So konfigurieren Sie Bereitstellungen (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Rolling updates and deployments (Fortlaufende Updates und Bereitstellungen) die Option Edit (Bearbeiten).
5. Wählen Sie unter Application Deployments (Anwendungsbereitstellungen) eine Option für Deployment policy (Bereitstellungsrichtlinie) sowie Batch-Einstellungen und Optionen für die Zustandsprüfung aus.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Im Bereich Application deployments (Anwendungsbereitstellungen) der Seite Rolling updates and deployments (Fortlaufende Updates und Bereitstellungen) sind die folgenden Optionen für Anwendungsbereitstellungen verfügbar:

- Deployment policy (Bereitstellungsrichtlinie) – Wählen Sie eine der folgenden Bereitstellungsoptionen aus:
  - All at once (Alle auf einmal) – Die neue Version wird auf allen Instances gleichzeitig bereitgestellt. Während der Bereitstellung sind alle Instances der Umgebung kurzzeitig außer Betrieb.
  - Rolling (Fortlaufend) – Die neue Version wird in Stapeln bereitgestellt. Während der Bereitstellung ist der jeweilige Batch außer Betrieb, sodass die Kapazität der Umgebung um die Anzahl der Instances in einem Batch reduziert wird.

- Rolling with additional batch (Fortlaufend mit zusätzlichem Stapel) – Die neue Version wird in Stapeln bereitgestellt, aber zuerst wird ein neuer Instance-Stapel gestartet, um die volle Kapazität auch während der Bereitstellung zu gewährleisten.
- Immutable (Unveränderlich) – Die neue Version wird mithilfe einer unveränderlichen Aktualisierung auf einer neuen Instance-Gruppe bereitgestellt.
- Datenverkehrsaufteilung – Stellen Sie die neue Version für eine neue Gruppe von Instances bereit und teilen Sie den eingehenden Clientdatenverkehr vorübergehend zwischen der vorhandenen und der neuen Anwendungsversion auf.

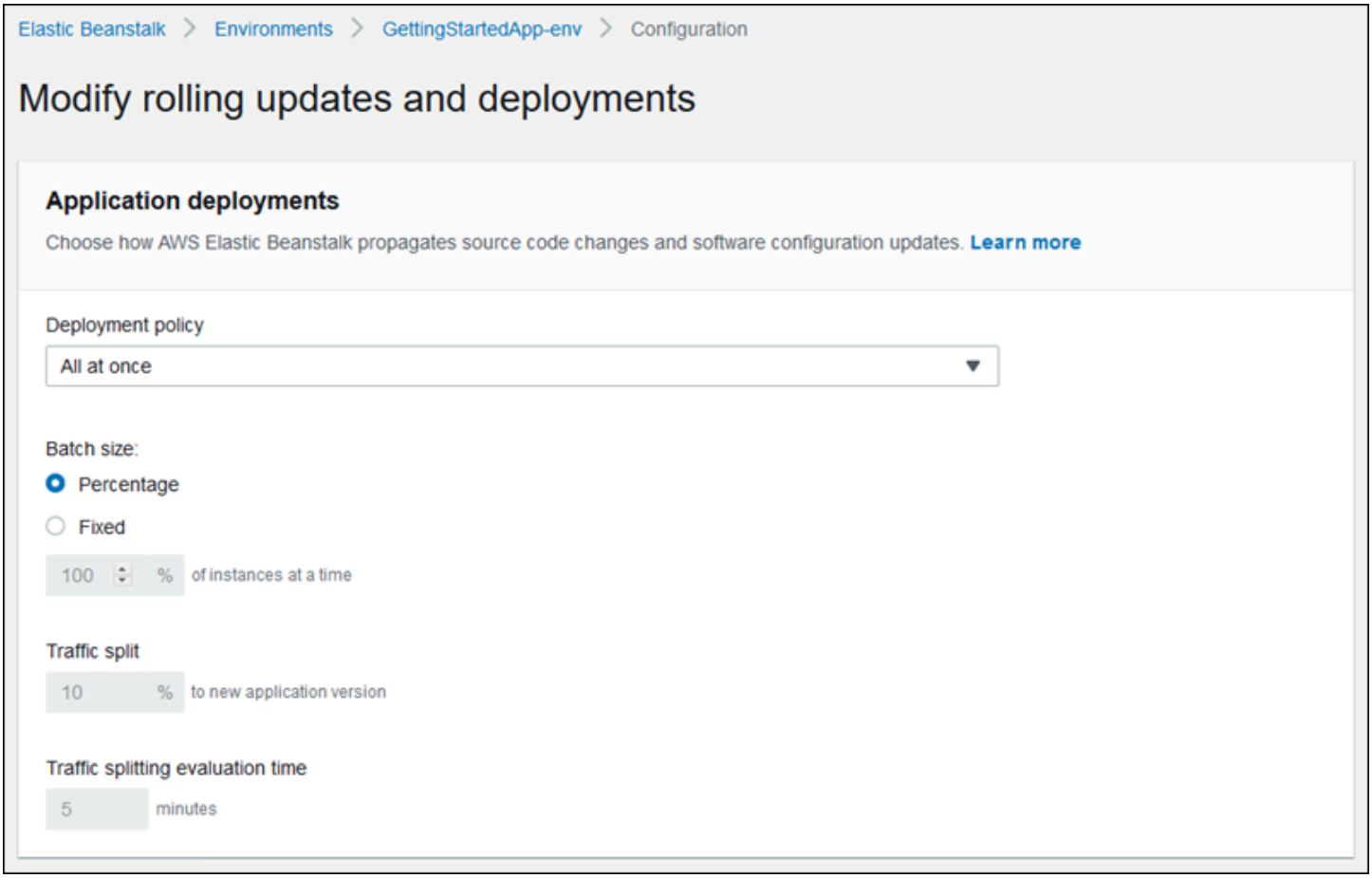
Für die Bereitstellungsrichtlinien Rolling (Fortlaufend) und Rolling with additional Batch (Fortlaufend mit zusätzlichem Stapel) können Sie Folgendes konfigurieren:

- Batch size (Stapelgröße) – Die Größe der Gruppe von Instances für die Bereitstellung in jedem Stapel.

  Wählen Sie Prozent, um einen Prozentsatz der Gesamtzahl der EC2 Instances in der Auto Scaling Scaling-Gruppe zu konfigurieren (bis zu 100 Prozent), oder wählen Sie Fest, um eine feste Anzahl von Instances zu konfigurieren (bis zur maximalen Anzahl von Instanzen in der Auto Scaling Scaling-Konfiguration Ihrer Umgebung).

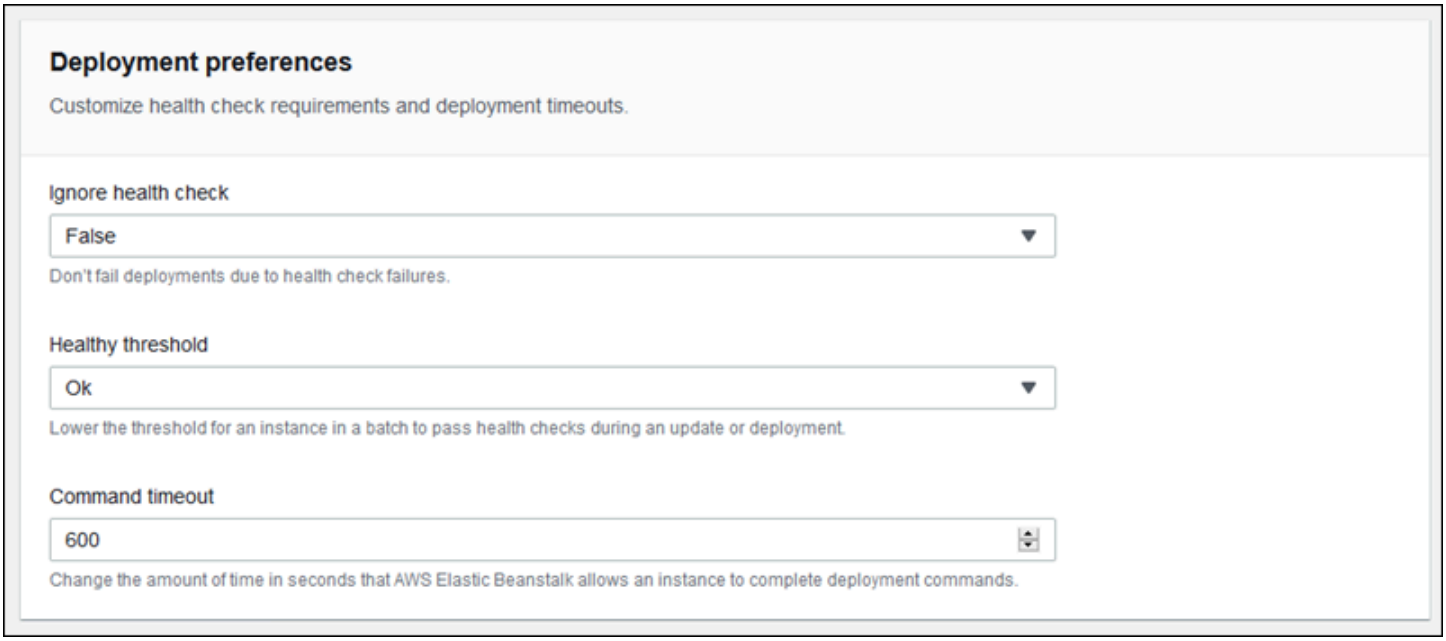
Sie können für die Bereitstellungsrichtlinie Traffic Splitting (Datenverkehrsaufteilung) Folgendes konfigurieren:

- Traffic split (Datenverkehrsaufteilung) – Der anfängliche Prozentsatz des eingehenden Clientdatenverkehrs, den Elastic Beanstalk auf Umgebungs-Instances verlagert, auf denen die von Ihnen bereitgestellte neue Anwendungsversion ausgeführt wird.
- Auswertungszeit für Datenverkehrsaufteilung – Der Zeitraum (in Minuten), den Elastic Beanstalk nach einer anfänglich fehlerfreien Bereitstellung wartet, bevor der gesamte eingehende Clientdatenverkehr auf die von Ihnen bereitgestellte neue Anwendungsversion verschoben wird.



Der Abschnitt Deployment Preferences (Bereitstellungspräferenzen) enthält die Optionen für Zustandsprüfungen.

- Ignore health check (Zustandsprüfung ignorieren) – Verhindert den Rollback einer Bereitstellung, falls ein Batch nicht innerhalb des unter Command timeout (Befehls-Timeout) angegebenen Werts als stabil eingestuft wird.
- Healthy threshold (Schwellenwert für den fehlerfreien Zustand) – Damit kann der Grenzwert, bei dem eine Instance als stabil eingestuft wird, für fortlaufende Bereitstellungen, fortlaufende Aktualisierungen und unveränderliche Aktualisierungen herabgesetzt werden.
- Command timeout (Befehls-Timeout) – Gibt an, wie viele Sekunden darauf gewartet wird, dass eine Instance als stabil eingestuft wird, bevor die Bereitstellung abgebrochen oder – bei aktivierter Ignore health check (Zustandsprüfung ignorieren)-Option – mit dem nächsten Batch begonnen wird.



Funktionsweise fortlaufender Bereitstellungen

Im Rahmen der Batch-Verarbeitung trennt Elastic Beanstalk alle Instances im Batch vom Load Balancer, stellt die neue Anwendungsversion bereit und ordnet die Instances wieder zu. Wenn Sie Connection Draining aktivieren, entfernt Elastic Beanstalk bestehende Verbindungen von den EC2 Amazon-Instances in jedem Batch, bevor mit der Bereitstellung begonnen wird.

Nachdem die Instances im Batch wieder dem Load Balancer zugeordnet wurden, wartet Elastic Load Balancing, bis die Instances die mindestens erforderlichen Elastic Load Balancing-Zustandsprüfungen (der Healthy check count threshold-Wert) bestanden haben, anschließend wird dann der Datenverkehr an sie weitergeleitet. Falls keine URL für die Zustandsprüfung konfiguriert ist, kann das sehr schnell gehen, da eine Instance die Zustandsprüfung bereits besteht, sobald sie eine TCP-Verbindung annimmt. Ist eine URL für die Zustandsprüfung definiert, leitet der Load Balancer den Datenverkehr erst an die aktualisierten Instances weiter, wenn diese als Antwort auf eine `200 OK`-Anforderung den Statuscode `HTTP GET` an die URL für die Zustandsprüfung senden.

Elastic Beanstalk wartet, bis alle Instances in einem Batch stabil sind, bevor mit dem nächsten Batch begonnen wird. Beim Basis-Health-Reporting hängt der Status einer Instance vom Status der Elastic Load Balancing-Zustandsprüfung ab. Wenn alle Instances im Batch genügend Zustandsprüfungen bestanden haben, um von Elastic Load Balancing als stabil eingestuft zu werden, ist der Batch abgeschlossen. Sind erweiterte Zustandsberichte aktiviert, berücksichtigt Elastic Beanstalk noch einige weitere Faktoren, darunter das Ergebnis eingehender Anforderungen. Mit erweiterten Zustandsberichten müssen alle Instances 12 aufeinanderfolgende Zustandsprüfungen mit dem Status

"OK" innerhalb von zwei Minuten für Webserverumgebungen und 18 Zustandsprüfungen innerhalb von drei Minuten für Worker-Umgebungen bestehen.

Die Bereitstellung schlägt fehl, wenn ein Instance-Batch nicht innerhalb des Timeout-Werts für Befehle als stabil eingestuft wird. Nach einer fehlgeschlagenen Bereitstellung überprüfen Sie den Zustand der Instances in Ihrer Umgebung nach Informationen über die Ursache des Fehlers. Führen Sie dann einen Rollback mit einer anderen Bereitstellung mit einer fixen oder funktionierenden, guten Version Ihrer Anwendung durch.

Schlägt eine Bereitstellung nach dem erfolgreichen Abschluss von einem oder mehreren Stapeln fehl, wird auf den aktualisierten Stapeln die neue Anwendungsversion und auf ausstehenden Stapeln weiterhin die alte Version ausgeführt. Die auf den Umgebungs-Instances ausgeführte Version wird auf der Zustandsprüfungsseite in der Konsole angezeigt. Auf dieser Seite wird die Bereitstellungs-ID der zuletzt für die Instances der Umgebung ausgeführte Bereitstellung angegeben. Wenn Sie Instances aus der fehlgeschlagenen Bereitstellung beenden, ersetzt Elastic Beanstalk diese durch Instances, auf denen die Anwendungsversion der letzten erfolgreichen Bereitstellung ausgeführt wird.

Funktionsweise von Bereitstellungen mit Datenverkehrsaufteilung

Mit Datenverkehrsaufteilungs-Bereitstellungen können Sie Canary-Tests durchführen. Sie leiten einen Teil des eingehenden Clientdatenverkehrs an die neue Anwendungsversion weiter, um die Integrität der Anwendung zu überprüfen, bevor Sie die neue Version einsetzen und den gesamten Datenverkehr dahin weiterleiten.

Während einer Bereitstellung mit Datenverkehrsaufteilung erstellt Elastic Beanstalk einen neuen Satz von Instance in einer separaten temporären Auto Scaling-Gruppe. Elastic Beanstalk weist dann den Load Balancer an, einen bestimmten Prozentsatz des eingehenden Datenverkehrs Ihrer Umgebung an die neuen Instances weiterzuleiten. Anschließend verfolgt Elastic Beanstalk für einen festgelegten Zeitraum die Integrität dieser neuen Instances. Ist alles in Ordnung, verschiebt Elastic Beanstalk den verbleibenden Datenverkehr auf die neuen Instances und fügt sie der ursprünglichen Auto Scaling-Gruppe der Umgebung hinzu, wodurch die alten Instances ersetzt werden. Anschließend bereinigt Elastic Beanstalk die alten Instances und entfernt die temporäre Auto Scaling-Gruppe.

> **Note**
>
> Die Kapazität der Umgebung ändert sich während einer Bereitstellung mit Datenverkehrsaufteilung nicht. Elastic Beanstalk startet die gleiche Anzahl von Instances in der temporären Auto Scaling-Gruppe wie in der ursprünglichen Auto Scaling-Gruppe zum Zeitpunkt des Bereitstellungsbeginns. Für die Dauer der Bereitstellung wird dann eine

konstante Anzahl von Instances in beiden Auto Scaling-Gruppen verwaltet. Berücksichtigen Sie diese Tatsache bei der Konfiguration der Evaluierungszeit für die Datenverkehrsaufteilung der Umgebung.

Das Rollback der Bereitstellung auf die vorherige Anwendungsversion erfolgt schnell und wirkt sich nicht auf den Dienst für den Clientdatenverkehr aus. Wenn die neuen Instances keine Zustandsprüfungen bestehen oder wenn Sie die Bereitstellung abbrechen, verschiebt Elastic Beanstalk den Datenverkehr zurück zu den alten Instances und beendet die neuen Instances. Sie können jede Bereitstellung abbrechen, indem Sie die Übersichtsseite der Umgebung in der Elastic-Beanstalk-Konsole verwenden und unter Environment actions (Umgebungsaktionen) die Option Abort current operation (Aktuellen Vorgang abbrechen) auswählen. Sie können auch die AbortEnvironmentUpdateAPI oder den entsprechenden Befehl aufrufen. AWS CLI

Für die Bereitstellung zur Datenverkehrsaufteilung ist ein Application Load Balancer erforderlich. Elastic Beanstalk verwendet diesen Load Balancer-Typ standardmäßig, wenn Sie Ihre Umgebung mit der Elastic Beanstalk-Konsole oder der EB CLI erstellen.

Namespaces für die Bereitstellungsoption

Sie können die Konfigurationsoptionen im Namespace `aws:elasticbeanstalk:command` zum Konfigurieren Ihrer Bereitstellungen verwenden. Wenn Sie die Richtlinie für die Datenverkehrsaufteilung auswählen, stehen zusätzliche Optionen für diese Richtlinie im Namespace `aws:elasticbeanstalk:trafficsplitting` zur Verfügung.

Mit der Option `DeploymentPolicy` legen Sie den Bereitstellungstyp fest. Folgende Werte werden unterstützt:

- `AllAtOnce` – Deaktiviert fortlaufende Bereitstellungen; die Bereitstellung erfolgt stets auf allen Instances gleichzeitig.
- `Rolling` – Aktiviert standardmäßig fortlaufende Bereitstellungen.
- `RollingWithAdditionalBatch` – Startet vor der Bereitstellung einen zusätzlichen Instance-Batch, um die volle Kapazität zu gewährleisten.
- `Immutable` – Führt bei jeder Bereitstellung eine unveränderliche Aktualisierung aus.
- `TrafficSplitting` – Führt Bereitstellungen mit Datenverkehrsaufteilung aus, um Ihre Anwendungsbereitstellungen einem Canary-Test zu unterziehen.

Wenn Sie fortlaufende Bereitstellungen aktivieren, legen Sie die Optionen `BatchSize` und `BatchSizeType` fest, um die Größe der einzelnen Batches zu definieren. Um beispielsweise 25 % aller Instances in den einzelnen Stapeln bereitzustellen, geben Sie folgende Optionen und Werte an.

Example.ebextensions/rolling-updates.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Rolling
    BatchSizeType: Percentage
    BatchSize: 25
```

Um unabhängig von der Anzahl ausgeführter Instances jeweils fünf Instances in den einzelnen Batches bereitzustellen und einen zusätzlichen Batch mit fünf Instances, auf denen bereits die neue Version ausgeführt wird, vor der Außerbetriebnahme einer anderen Instance bereitzustellen, geben Sie folgende Optionen und Werte an.

Example.ebextensions/rolling-additionalbatch.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: RollingWithAdditionalBatch
    BatchSizeType: Fixed
    BatchSize: 5
```

Um für jede Bereitstellung mit dem Zustandsprüfungs-Grenzwert Warning (Warnung) eine unveränderliche Aktualisierung auszuführen und die Bereitstellung fortzusetzen, auch wenn die Instances im Stapel die Zustandsprüfungen nicht innerhalb von 15 Minuten bestanden haben, geben Sie folgende Optionen und Werte an.

Example.ebextensions/immutable-ignorehealth.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Immutable
    HealthCheckSuccessThreshold: Warning
    IgnoreHealthCheck: true
    Timeout: "900"
```

Geben Sie die folgenden Optionen und Werte an, um Bereitstellungen mit Datenverkehrsaufteilung durchzuführen, 15 Prozent des Clientdatenverkehrs an die neue Anwendungsversion weiterzuleiten und die Integrität für 10 Minuten zu bewerten.

Example.ebextensions/traffic-splitting.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: TrafficSplitting
  aws:elasticbeanstalk:trafficsplitting:
    NewVersionPercent: "15"
    EvaluationTime: "10"
```

Die EB-CLI und Elastic-Beanstalk-Konsole wenden die für die vorhergehenden Optionen empfohlenen Werte an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter Empfohlene Werte.

Blau/Grün-Bereitstellungen mit Elastic Beanstalk

Da AWS Elastic Beanstalk bei der Aktualisierung Ihrer Anwendungsversionen ein direktes Update durchgeführt wird, kann es sein, dass Ihre Anwendung für Benutzer für einen kurzen Zeitraum nicht verfügbar ist. Um dies zu vermeiden, führen Sie eine blue/green Bereitstellung durch. Stellen Sie dazu die neue Version in einer separaten Umgebung bereit und tauschen Sie dann die beiden Umgebungen CNAMEs aus, um den Datenverkehr sofort auf die neue Version umzuleiten.

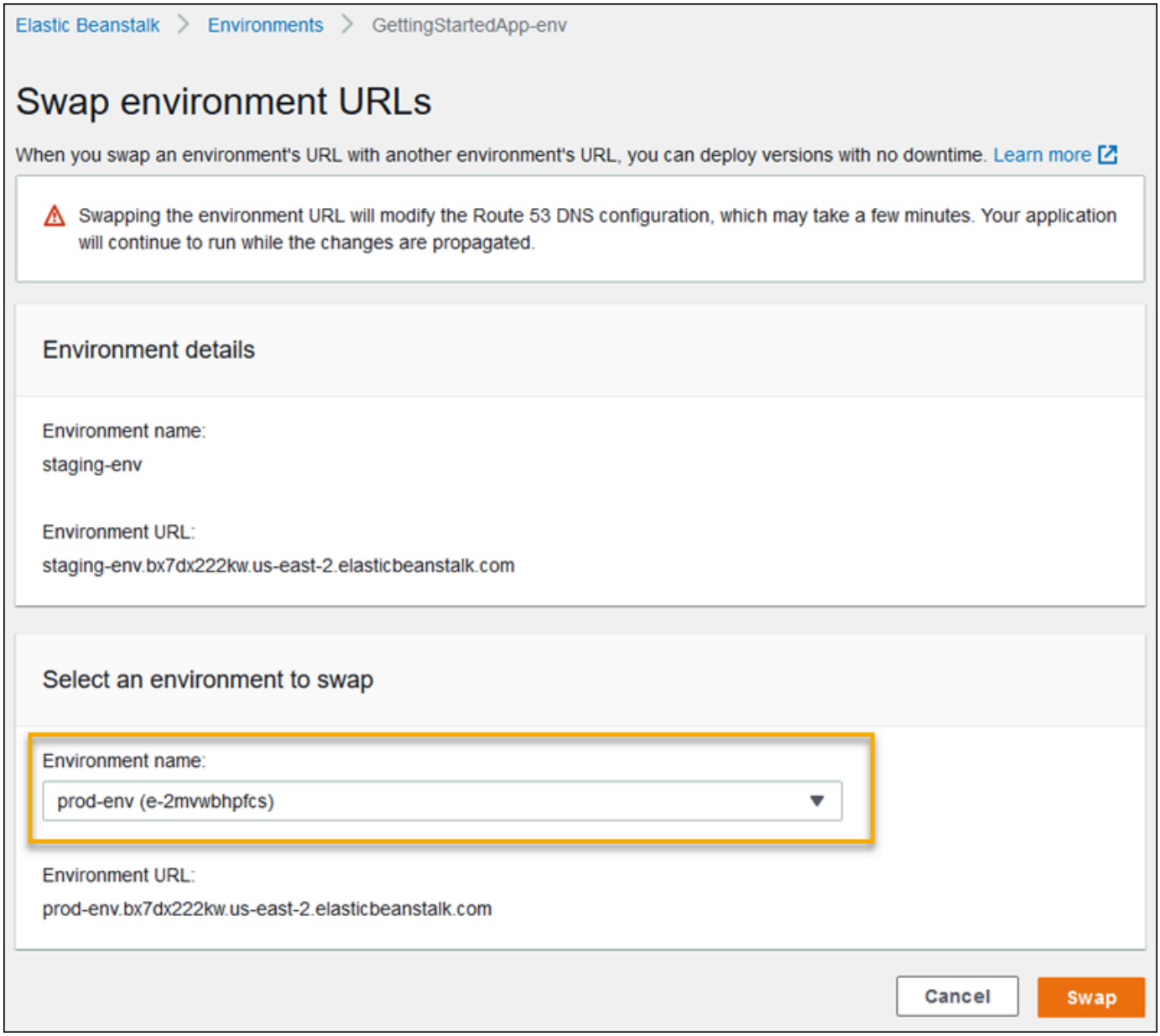
Eine blue/green Bereitstellung ist auch erforderlich, wenn Sie eine Umgebung auf eine inkompatible Plattformversion aktualisieren möchten. Weitere Informationen finden Sie unter the section called “Plattformaktualisierungen”.

Für Blau/Grün-Bereitstellungen muss Ihre Umgebung unabhängig von Ihrer Produktionsdatenbank ausgeführt werden, wenn Ihre Anwendung eine verwendet. Wenn Ihre Umgebung eine Datenbank enthält, die Elastic Beanstalk in Ihrem Namen erstellt hat, werden die Datenbank und die Verbindung der Umgebung nur beibehalten, wenn Sie bestimmte Aktionen ausführen. Wenn Sie eine Datenbank behalten möchten, verwenden Sie eine der Elastic-Beanstalk-Datenbanklebenszyklusoptionen. Sie können die Option „Beibehalten“ wählen, um die Datenbank und Umgebung nach dem Entkoppeln der Datenbank betriebsbereit zu halten. Weitere Informationen finden Sie unter Datenbank-Lebenszyklus im Kapitel Konfigurieren von Umgebungen in diesem Leitfaden.

Anweisungen zum Konfigurieren Ihrer Anwendung für die Verbindung mit einer Amazon-RDS-Instance, die nicht von Elastic Beanstalk verwaltet wird, finden Sie unter Verwenden von Elastic Beanstalk mit Amazon RDS.

Um eine blue/green Bereitstellung durchzuführen

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Klonen Sie Ihre aktuelle Umgebung oder starten Sie eine neue Umgebung mit der gewünschten Plattformversion.
3. Stellen Sie die neue Anwendungsversion in der neuen Umgebung bereit.
4. Testen Sie die neue Version auf der neuen Umgebung.
5. Wählen Sie auf der Seite mit der Umgebungsübersicht die Option Aktionen und anschließend Umgebung austauschen aus. URLs
6. Wählen Sie unter Environment name (Umgebungsname) die aktuelle Umgebung aus.



7. Wählen Sie Austauschen.

Elastic Beanstalk tauscht die CNAME-Datensätze der alten und neuen Umgebungen aus und leitet so den Datenverkehr von der alten auf die neue Version um.

Nachdem Elastic Beanstalk den Austausch abgeschlossen hat, stellen Sie sicher, dass die neue Umgebung reagiert, wenn Sie versuchen, eine Verbindung zur alten Umgebungs-URL herzustellen. Beenden Sie Ihre alte Umgebung jedoch nicht, bis die DNS-Änderungen übertragen werden und Ihre alten DNS-Datensätze ablaufen. DNS-Server löschen nicht immer alte Einträge aus ihrem Cache basierend auf der Time-to-Live (TTL), die Sie für Ihre DNS-Einträge festlegen.

Implementierung der CI/CD Integration mit Ihrer Elastic Beanstalk Beanstalk-Umgebung

Elastic Beanstalk lässt sich in viele CI/CD Tools integrieren, um Ihren Arbeitsablauf bei der Anwendungsentwicklung zu automatisieren. CI/CD Methoden ermöglichen es Ihnen, Ihre Anwendungen mit minimalem manuellem Eingriff automatisch zu erstellen, zu testen und bereitzustellen. Continuous delivery/deployment (CD) erweitert die kontinuierliche Integration (CI) durch die Automatisierung des Bereitstellungsprozesses. Sie können mithilfe von AWS Services wie CodePipeline oder Tools von Drittanbietern wie Jenkins optimierte Bereitstellungspipelines erstellen und so konsistente, zuverlässige Bereitstellungen in GitLab Ihren Elastic Beanstalk Beanstalk-Umgebungen sicherstellen.

Themen

- AWS Quellen für den Einstieg
- Weitere Ressourcen
- Verwenden von GitHub Aktionen für die Bereitstellung auf Elastic Beanstalk

AWS Quellen für den Einstieg

In der folgenden Liste sind CI/CD Tools und die entsprechenden AWS Ressourcen aufgeführt, die step-by-step Anleitungen zur Erstellung automatisierter Deployment-Pipelines für Elastic Beanstalk Beanstalk-Umgebungen bieten:

- AWS CodePipeline— In diesem Tutorial „AWS Erste Schritte“ im Resource Center erfahren Sie, wie Sie eine kontinuierliche Deployment-Pipeline zu Elastic Beanstalk von GitHub , S3 oder einrichten. AWS CodeCommit
- GitHub Aktionen — Unter erfahren SieVerwenden von GitHub Aktionen für die Bereitstellung auf Elastic Beanstalk, wie Sie YAML-basierte Workflows konfigurieren, um direkt von dort aus eine kontinuierliche Deployment-Pipeline zu Elastic Beanstalk einzurichten. GitHub
- GitLab— Dieser Blogbeitrag zur AWS DevOps Entwicklerproduktivität zeigt, wie GitLab kontinuierliche Pipelines konfiguriert werden, um Node.js -Anwendungen in Elastic Beanstalk Docker-Umgebungen bereitzustellen.
- Azure DevOps — Dieser AWS Blogbeitrag von.NET führt Sie durch die Implementierung einer kontinuierlichen Bereitstellungspipeline von einem DevOps Azure-Git-Repository zu Elastic Beanstalk mithilfe von Azure-Pipelines.

Weitere Ressourcen

Die folgenden Tools und Ressourcen von Drittanbietern können Ihnen bei der Implementierung automatisierter Deployment-Pipelines für Elastic Beanstalk Beanstalk-Umgebungen helfen:

- Jenkins — Das AWS EBDeployment Jenkins-Plugin ermöglicht die direkte Bereitstellung in Elastic Beanstalk Beanstalk-Umgebungen von Ihrer Jenkins-Job-Konfigurationsseite aus.
- Circle CI: — The Orbs for Elastic Beanstalk bieten wiederverwendbare Konfigurationspakete für die Bereitstellung und Skalierung von Anwendungen in Elastic Beanstalk.
- Bitbucket-Pipelines — Der Artikel Elastic Beanstalk-Anwendung mithilfe von Bitbucket-Pipelines bereitstellen bietet ein grundlegendes Konfigurationsbeispiel für die Implementierung von Bitbucket-Pipelines mit Elastic Beanstalk.

Verwenden von GitHub Aktionen für die Bereitstellung auf Elastic Beanstalk

GitHub Mit Aktionen können Sie Ihre Anwendung automatisch auf Elastic Beanstalk bereitstellen, wenn Sie Codeänderungen in Ihr Repository übertragen. Die Elastic Beanstalk Deploy-Aktion bietet eine einfache YAML-Schnittstelle, die das Erstellen von Anwendungsversionen, das Hochladen von Quellpaketen auf Amazon S3 und das Deployment in Ihrer Elastic Beanstalk Beanstalk-Umgebung übernimmt.

Beispiel-ArbeitsFlows

Mit dem folgenden Beispiel-Workflow wird bei jedem Push zum Branch eine Anwendung in einer Elastic Beanstalk Beanstalk-Umgebung bereitgestellt. `main` Erstellen Sie eine `.yml` Datei in Ihrem Repository unter. `.github/workflows/`

Example GitHub Aktionsablauf für die Bereitstellung von Elastic Beanstalk

```
name: Deploy to Elastic Beanstalk

on:
  push:
    branches:
      - main

permissions:
  id-token: write
  contents: read
```

```
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v4
        with:
          role-to-assume: arn:aws:iam::123456789012:role/my-github-actions-role
          aws-region: us-east-1

      - name: Deploy to Elastic Beanstalk
        uses: aws-actions/aws-elasticbeanstalk-deploy@v1.0.0
        with:
          aws-region: us-east-1
          application-name: my-application
          environment-name: my-application-env
```

Dieser Workflow checkt Ihr Repository aus, verwendet OpenID Connect (OIDC) zur Authentifizierung AWS über die Aktion „AWS Anmeldeinformationen konfigurieren“ und stellt dann Ihre Anwendung auf Elastic Beanstalk bereit. Die Aktion Deploy packt Ihre Repository-Inhalte, lädt das Quellpaket auf Amazon S3 hoch, erstellt eine neue Anwendungsversion und erstellt oder aktualisiert Ihre Umgebung. Standardmäßig wird darauf gewartet, dass die Bereitstellung abgeschlossen ist und die Umgebung wieder in einen fehlerfreien Zustand zurückkehrt.

Weitere Konfigurationsoptionen und erweiterte Beispiele finden Sie in der README-Datei zur Elastic Beanstalk Deploy-Aktion. GitHub

Weitere Ressourcen

- Elastic Beanstalk: Aktion bereitstellen auf GitHub
- Aktion „ AWS Anmeldeinformationen konfigurieren“ für GitHub
- Konfiguration von OpenID Connect in Amazon Web Services (GitHubDokumentation)

Konfigurationsänderungen

Wenn Sie die Einstellungen der Konfigurationsoptionen im Abschnitt Konfiguration der Environment Management Console ändern, AWS Elastic Beanstalk wird die Änderung auf alle betroffenen

Ressourcen übertragen. Zu diesen Ressourcen gehören der Load Balancer, der den Traffic an die EC2 Amazon-Instances verteilt, auf denen Ihre Anwendung ausgeführt wird, die Auto Scaling Scaling-Gruppe, die diese Instances verwaltet, und die EC2 Instances selbst.

Viele Konfigurationsänderungen können in eine ausgeführte Umgebung übernommen werden, ohne vorhandene Instances zu ersetzen. Beispielsweise löst das Festlegen einer URL für die Zustandsprüfung ein Umgebungsaktualisierung aus, um die Load Balancer-Einstellungen zu ändern. Dies verursacht jedoch keine Ausfallzeit, weil die Instances, auf denen die Anwendung ausgeführt wird, während des Aktualisierungsvorgangs weiterhin Anforderungen verarbeiten.

Falls mit Konfigurationsänderungen die Startkonfiguration oder die VPC-Einstellungen geändert werden, müssen alle Instances in der Umgebung beendet und ersetzt werden. Wenn Sie beispielsweise den Instance-Typ oder die SSH-Schlüsseleinstellung für Ihre Umgebung ändern, müssen die EC2 Instances beendet und ersetzt werden. Elastic Beanstalk bietet verschiedene Richtlinien, die festlegen, wie dieser Ersatz durchgeführt wird.

- Rolling updates (Fortlaufende Aktualisierungen) – Elastic Beanstalk wendet Ihre Konfigurationsänderungen in Stapeln an, wobei eine Mindestanzahl von Instances jederzeit ausgeführt wird und für Datenverkehr verfügbar ist. Dies verhindert Ausfallzeiten während des Aktualisierungsvorgangs. Details hierzu finden Sie unter Fortlaufende Updates.
- Immutable updates (Unveränderliche Aktualisierungen) – Elastic Beanstalk startet eine temporäre Auto Scaling-Gruppe außerhalb Ihrer Umgebung mit einem separaten Satz von Instances, die in der neuen Konfiguration ausgeführt werden. Anschließend platziert Elastic Beanstalk diese Instances hinter dem Load Balancer Ihrer Umgebung. Alte und neue Instances dienen beide dem Datenverkehr, bis die neuen Instances die Zustandsprüfungen bestehen. Elastic Beanstalk verschiebt zu diesem Zeitpunkt die neuen Instances in die Auto Scaling-Gruppe Ihrer Umgebung und beendet die temporäre Gruppe und alte Instances. Details hierzu finden Sie unter Unveränderliche Updates.
- Disabled (Deaktiviert) – Elastic Beanstalk macht keinen Versuch, Ausfallzeiten zu vermeiden. Es beendet die vorhandenen Instances Ihrer Umgebung und ersetzt sie durch neue Instances, die mit der neuen Konfiguration ausgeführt werden.

> **Warning**
>
> Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle gesammelten Amazon EC2 Burst-Guthaben verloren. Dies geschieht in den folgenden Fällen:

- Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
- Unveränderliche Updates
- Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Unterstützte Aktualisierungs-Typen

Einstellung für fortlaufende Aktualisierung	Umgebungen mit Load Balancing	Umgebungen mit einer Instance	Ältere Windows Server-Umgebungen†
Disabled	✓ Ja	✓ Ja	✓ Ja
Rolling Based on Health	✓ Ja	✗ Nein	✓ Ja
Rolling Based on Time	✓ Ja	✗ Nein	✓ Ja
Immutable (Unveränderlich)	✓ Ja	✓ Ja	✗ Nein

† Für die Zwecke dieser Tabelle ist eine Legacy-Windows Server-Umgebung eine Umgebung, die auf einer Windows Server-Plattformkonfiguration mit einer früheren IIS-Version als IIS 8.5 basiert.

Themen

- Fortlaufende Updates der Elastic Beanstalk-Umgebungskonfiguration
- Unveränderliche Umgebungsaktualisierungen

Fortlaufende Updates der Elastic Beanstalk-Umgebungskonfiguration

Wenn eine Konfigurationsänderung vorsieht, dass Instances ersetzt werden, kann Elastic Beanstalk das Update in Stapeln durchführen, um Ausfallzeiten zu vermeiden, während die Änderung verbreitet wird. Während eines fortlaufenden Updates wird die Kapazität nur um die Größe eines einzigen Stapels reduziert, den Sie konfigurieren können. Elastic Beanstalk schaltet einen Stapel an Instances

offline, beendet diese und startet dann einen Stapel mit der neuen Konfiguration. Nachdem der neue Stapel beginnt, Anfragen zu bedienen, fährt Elastic Beanstalk mit dem nächsten Stapel fort.

Fortlaufende Konfigurations-Update-Stapel können in regelmäßigen Abständen (zeitbasiert) verarbeitet werden – mit einer Verzögerung zwischen jedem Stapel oder basierend auf dem Zustand. Für zeitbasierte fortlaufende Updates können Sie die Dauer konfigurieren, die Elastic Beanstalk wartet, nachdem der Start eines Instance-Stapels abgeschlossen ist, bevor mit dem nächsten Stapel fortgefahren wird. Durch diese Pausenzeit kann Ihre Anwendung einen Bootsrap durchführen und mit dem Verarbeiten von Anforderungen beginnen.

Bei zustandsbasierten fortlaufenden Updates wartet Elastic Beanstalk, bis Instances in einem Stapel die Zustandsprüfungen bestanden haben, bevor mit dem nächsten Stapel fortgefahren wird. Der Zustand einer Instance wird durch das System für Zustandsberichte bestimmt. Dieses kann grundlegend oder erweitert sein. Beim grundlegenden Zustandssystem wird ein Stapel als stabil betrachtet, wenn alle Instances darin die Elastic Load Balancing (ELB)-Zustandsprüfungen bestehen.

Bei erweiterten Zustandsberichten müssen alle Instances in einem Stapel mehrere aufeinanderfolgende Zustandsprüfungen bestehen, bevor Elastic Beanstalk mit dem nächsten Stapel fortfährt. Zusätzlich zu den ELB-Zustandsprüfungen, bei denen nur Ihre Instances geprüft werden, werden bei der erweiterten Zustandsprüfung Anwendungsprotokolle und der Status der anderen Ressourcen Ihrer Umgebung überprüft. In einer Webserverumgebung mit erweiterten Zustandsberichten müssen alle Instances im Laufe von zwei Minuten zwölf Zustandsprüfungen bestehen (18 Prüfungen über drei Minuten für Worker-Umgebungen). Wenn eine Instance die Zustandsprüfung nicht besteht, wird die Anzahl zurückgesetzt.

Wenn ein Stapel während des Timeouts des fortlaufenden Updates keinen stabilen Zustand erreicht (Standard: 30 Minuten), wird das Update abgebrochen. Das Timeout beim fortlaufenden Update ist eine Konfigurationsoption, die im `aws:autoscaling:updatepolicy:rollingupdate`-Namespace verfügbar ist. Wenn Ihre Anwendung Zustandsprüfungen nicht mit dem `Ok`-Status besteht, sie auf einer anderen Stufe jedoch stabil ist, können Sie die `HealthCheckSuccessThreshold`-Option im `aws:elasticbeanstalk:healthreporting:system`-Namespace so festlegen, dass die Stufe geändert wird, auf der Elastic Beanstalk eine Instance als stabil betrachtet.

Wenn das fortlaufende Update fehlschlägt, startet Elastic Beanstalk ein weiteres, um zur vorherigen Konfiguration zurückzukehren. Ein fortlaufendes Update kann aufgrund fehlerhafter Zustandsprüfungen fehlschlagen oder wenn Sie durch das Starten neuer Instances die Kontingente auf Ihrem Konto überschreiten. Wenn Sie beispielsweise ein Kontingent für die Anzahl der EC2

Amazon-Instances erreichen, kann das fortlaufende Update fehlschlagen, wenn versucht wird, einen Stapel neuer Instances bereitzustellen. In diesem Fall schlägt das Rollback ebenfalls fehl.

Bei einem fehlgeschlagenen Rollback wird der Aktualisierungsvorgang beendet und Ihre Umgebung verbleibt in einem instabilen Zustand. Nicht verarbeitete Stapel führen nach wie vor Instances mit der alten Konfiguration aus, während erfolgreich abgeschlossene Stapel über die neue Konfiguration verfügen. Um eine Umgebung nach einem fehlgeschlagenen Rollback zu reparieren, beheben Sie zuerst das zugrundeliegende Problem, durch das das Update fehlschlägt, und initiieren Sie anschließend ein weiteres Umgebungs-Update.

Eine alternative Methode ist, die neue Version Ihrer Anwendung für eine andere Umgebung bereitzustellen und dann einen CNAME-Austausch durchzuführen, um den Datenverkehr ohne Ausfallzeit weiterzuleiten. Weitere Informationen finden Sie unter Blau/Grün-Bereitstellungen mit Elastic Beanstalk.

Fortlaufende Updates im Vergleich zu fortlaufenden Bereitstellungen

Rollende Updates erfolgen, wenn Sie Einstellungen ändern, die die Bereitstellung neuer EC2 Amazon-Instances für Ihre Umgebung erfordern. Dazu gehören Änderungen an der Auto Scaling-Gruppenkonfiguration, wie Instance-Typ- und Schlüsselpaar-Einstellungen, sowie Änderungen an den VPC-Einstellungen. Bei einem fortlaufenden Update wird jeder Instance-Stapel beendet, bevor ein neuer Stapel bereitgestellt wird, der den alten ersetzen soll.

Fortlaufende Bereitstellungen treten auf, wenn Sie Ihre Anwendung bereitstellen. Sie können in der Regel durchgeführt werden, ohne dass Instances in Ihrer Umgebung ersetzt werden. Elastic Beanstalk schaltet jeden Stapel offline, stellt die neue Anwendungsversion bereit und nimmt sie dann wieder in Betrieb.

Eine Ausnahme liegt dann vor, wenn Sie Einstellungen ändern, bei denen der Instance-Austausch gleichzeitig mit der Bereitstellung einer neuen Anwendungsversion erfolgen muss. Wenn Sie beispielsweise die Schlüsselnamen-Einstellungen in einer Konfigurationsdatei in Ihrem Quell-Bundle ändern und es in Ihrer Umgebung bereitstellen, lösen Sie ein fortlaufendes Update aus. Anstatt Ihre neue Anwendungsversion für jeden Stapel an vorhandenen Instances bereitzustellen, wird ein neuer Instance-Stapel mit der neuen Konfiguration bereitgestellt. In diesem Fall erfolgt keine separate Bereitstellung, da die neuen Instances mit der neuen Anwendungsversion aufgerufen werden.

Jedes Mal, wenn neue Instances als Teil eines Umgebungs-Updates bereitgestellt werden, gibt es eine Bereitstellungsphase, in der der Quellcode Ihrer Anwendung für die neuen Instances bereitgestellt wird. Darüber hinaus werden alle Konfigurationseinstellungen, die das Betriebssystem

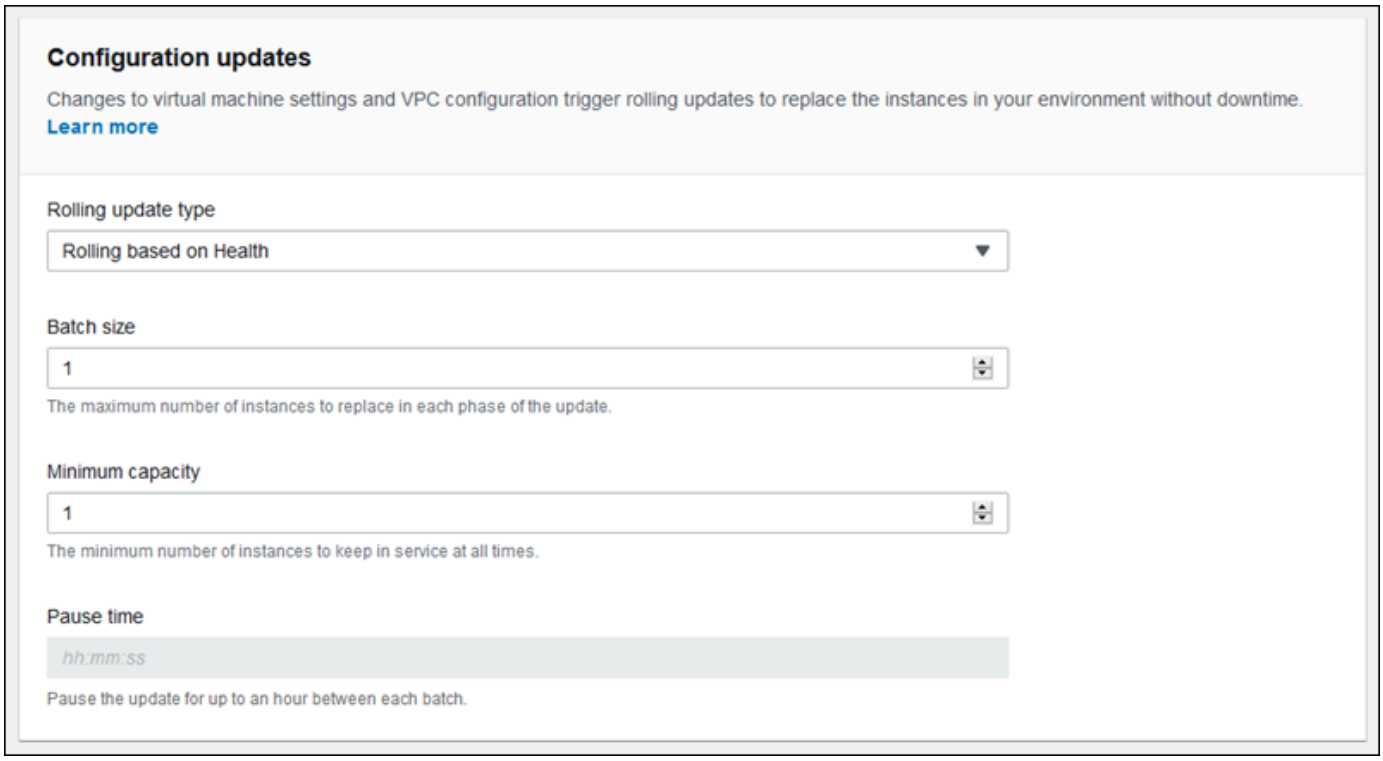
oder die Software auf den Instances ändern, angewendet. Einstellungen der Bereitstellungs-Zustandsprüfung (Ignore health check (Zustandsprüfung ignorieren)), Healthy threshold (Zulässige Schwelle) (Command timeout (Befehls-Timeout)) gelten auch für zustandsbasierte fortlaufende Updates und unveränderliche Updates während der Bereitstellungsphase.

Konfigurieren von fortlaufendes Updates

Sie können fortlaufende Updates in der Elastic Beanstalk-Konsole aktivieren und konfigurieren.

So aktivieren Sie fortlaufende Updates

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Rolling updates and deployments (Fortlaufende Updates und Bereitstellungen) die Option Edit (Bearbeiten).
5. Wählen Sie im Bereich Configuration updates (Konfigurationsaktualisierungen) für Rolling update type (Typ der fortlaufenden Aktualisierung) und wählen Sie eine der Optionen unter Rolling (Fortlaufend) aus.



6. Wählen Sie die Einstellungen Batch size (Stapelgröße), Minimum capacity (Mindestkapazität) und Pause time (Pausendauer).
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Im Bereich Configuration updates (Konfigurationsaktualisierungen) der Seite Updates and Deployments (Updates und Bereitstellungen) sind die folgenden Optionen für fortlaufende Updates verfügbar:

- Rolling update type (Art des fortlaufenden Updates) – Elastic Beanstalk wartet, nachdem das Aktualisieren eines Instance-Stapels abgeschlossen wurde, bevor mit dem nächsten Stapel fortgefahren wird, damit diese Instances das Bootstrapping abschließen und mit dem Bereitstellen von Datenverkehr beginnen können. Wählen Sie aus den folgenden Optionen aus:
  - Rolling based on Health (Fortlaufend basierend auf dem Zustand) – Warten Sie, bis Instances im aktuellen Stapel stabil sind, bevor Sie Instances in Betrieb nehmen und den nächsten Stapel starten.
  - Rolling based on Time (Fortlaufend basierend auf Zeit) – Geben Sie eine Dauer ein, die zwischen dem Starten neuer Instances und deren Inbetriebnahme gewartet wird, bevor der nächste Stapel gestartet wird.
  - Immutable (Unveränderlich) – Wenden Sie die Konfigurationsänderungen auf eine neue Gruppe von Instances an, indem Sie ein unveränderliches Update durchführen.
- Batch size (Stapelgröße) – Die Anzahl der Instances, die in jedem Stapel ersetzt werden sollen, zwischen **1** und **10000**. Standardmäßig ist dieser Wert ein Drittel der Mindestgröße der Auto Scaling-Gruppe, auf eine Ganzzahl aufgerundet.
- Minimum capacity (Mindestkapazität) – Die minimale Anzahl der Instances, die weiterhin ausgeführt werden, während andere Instances aktualisiert werden, zwischen **0** und **9999**). Der Standardwert ist entweder die Mindestgröße der Auto Scaling-Gruppe oder eine Instance weniger als die maximale Größe der Auto Scaling-Gruppe – je nachdem, welche Zahl die niedrigere ist.
- Pause time (Pausendauer) (nur zeitbasiert) – Die Zeit, die gewartet wird, nachdem ein Stapel aktualisiert wurde, bevor mit dem nächsten Stapel fortgefahren wird, sodass Ihre Anwendung mit dem Empfang von Datenverkehr beginnen kann. Zwischen 0 Sekunden und einer Stunde.

Der aws:autoscaling:updatepolicy:rollingupdate-Namespace

Sie können auch die Konfigurationsoptionen im `aws:autoscaling:updatepolicy:rollingupdate`-Namespace zum Konfigurieren fortlaufender Bereitstellungen verwenden.

Verwenden Sie die `RollingUpdateEnabled`-Option, um fortlaufende Updates zu aktualisieren, und `RollingUpdateType`, um den Update-Typ auszuwählen. Folgende Werte werden für `RollingUpdateType` unterstützt:

- `Health` – Warten Sie, bis Instances im aktuellen Stapel stabil sind, bevor Sie Instances in Betrieb nehmen und den nächsten Stapel starten.
- `Time` – Geben Sie eine Dauer ein, die zwischen dem Starten neuer Instances und deren Inbetriebnahme gewartet wird, bevor der nächste Stapel gestartet wird.
- `Immutable` – Wenden Sie die Konfigurationsänderungen auf eine neue Gruppe von Instances an, indem Sie ein unveränderliches Update durchführen.

Wenn Sie fortlaufende Updates aktivieren, legen Sie die Optionen `MaxBatchSize` und `MinInstancesInService` fest, um die Größe der einzelnen Stapel zu definieren. Für zeitbasierte und zustandsbasierte fortlaufende Updates können Sie auch `PauseTime` bzw. `Timeout` konfigurieren.

Um beispielsweise bis zu fünf Instances gleichzeitig zu starten, während mindestens zwei Instances in Betrieb bleiben, und um fünf Minuten und 30 Sekunden zwischen Stapeln zu warten, geben Sie die folgenden Optionen und Werte an.

Example.ebextensions/timebased.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateEnabled: true
    MaxBatchSize: 5
    MinInstancesInService: 2
    RollingUpdateType: Time
    PauseTime: PT5M30S
```

Zum Aktivieren zustandsbasierter fortlaufender Updates mit einem 45-Minuten-Timeout für jeden Stapel geben Sie die folgenden Optionen und Werte an.

Example.ebextensions/healthbased.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateEnabled: true
    MaxBatchSize: 5
    MinInstancesInService: 2
    RollingUpdateType: Health
    Timeout: PT45M
```

`Timeout`und `PauseTime` Werte müssen im Format ISO8601 duration: angegeben werdenPT#H#M#S, wobei jedes # für die Anzahl der Stunden, Minuten oder Sekunden steht.

Die EB-CLI und Elastic-Beanstalk-Konsole wenden die für die vorhergehenden Optionen empfohlenen Werte an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter Empfohlene Werte.

Unveränderliche Umgebungsaktualisierungen

Unveränderliche Umgebungsaktualisierungen stellen eine Alternative zu fortlaufenden Updates dar. Unveränderliche Umgebungsaktualisierungen sorgen dafür, dass Konfigurationsänderungen, bei denen ein Ersetzen von Instances erforderlich ist, effizient und sicher angewendet werden. Wenn ein unveränderliches Umgebungs-Update fehlschlägt, erfordert der Rollbackvorgang nur das Beenden einer Auto Scaling-Gruppe. Ein fehlgeschlagenes fortlaufendes Update erfordert hingegen die Durchführung eines zusätzlichen fortlaufenden Updates, um die Änderungen rückgängig zu machen.

Um ein unveränderliches Umgebungs-Update durchzuführen, erstellt Elastic Beanstalk eine zweite, temporäre Auto Scaling-Gruppe hinter dem Load Balancer Ihrer Umgebung für die neuen Instances. Zunächst startet Elastic Beanstalk eine einzelne Instance mit der neuen Konfiguration in der neuen Gruppe. Diese Instance stellt Datenverkehr zusammen mit allen Instances in der ursprünglichen Auto Scaling-Gruppe bereit, auf denen die vorherige Konfiguration ausgeführt wird.

Wenn die erste Instance die Zustandsprüfungen besteht, startet Elastic Beanstalk zusätzliche Instances mit der neuen Konfiguration, die der Anzahl der Instances, die in der ursprünglichen Auto Scaling-Gruppe ausgeführt werden, entsprechen. Wenn alle neuen Instances Zustandsprüfungen bestehen, überträgt Elastic Beanstalk sie an die ursprüngliche Auto Scaling-Gruppe und beendet die temporäre Auto Scaling-Gruppe und alte Instances.

> **Note**
>
> Während eines unveränderlichen Umgebungs-Update verdoppelt sich die Kapazität Ihrer Umgebung für eine kurze Zeit, wenn die Instances in der neuen Auto Scaling-Gruppe Anforderungen verarbeiten und bevor die Instances der ursprünglichen Auto Scaling-Gruppe beendet werden. Wenn Ihre Umgebung viele Instances hat oder Sie ein niedriges On-Demand-Instance-Kontingent haben, stellen Sie sicher, dass Sie genügend Kapazität haben, um ein unveränderliches Umgebungs-Update durchzuführen. Wenn Sie in der Nähe der Kontingentgrenze liegen, sollten Sie stattdessen fortlaufende Updates verwenden.

Unveränderliche Updates erfordern erweiterte Zustandsberichte zum Auswerten des Zustands Ihrer Umgebung während des Updates. Erweiterte Zustandsberichte kombinieren Standard-Load Balancer-Zustandsprüfungen mit Instance-Überwachung, um sicherzustellen, dass die Instances, auf denen die neue Konfiguration ausgeführt wird, Anforderungen erfolgreich verarbeitet.

Sie können auch unveränderliche Updates für die Bereitstellung neuer Versionen Ihrer Anwendung verwenden, als Alternative zu fortlaufenden Bereitstellungen. Wenn Sie Elastic Beanstalk so konfigurieren, dass unveränderliche Updates für die Anwendungsbereitstellung verwendet werden, werden alle Instances in Ihrer Umgebung jedes Mal ersetzt, wenn Sie eine neue Version Ihrer Anwendung bereitstellen. Wenn eine unveränderliche Anwendungsbereitstellung fehlschlägt, kehrt Elastic Beanstalk die Änderungen sofort zurück, indem die neue Auto Scaling-Gruppe beendet wird. Dies kann eine teilweise Flottenbereitstellungen verhindern, die auftreten kann, wenn eine fortlaufende Bereitstellung fehlschlägt, nachdem einige Stapel bereits abgeschlossen wurden.

> **Warning**
>
> Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle gesammelten Amazon EC2 Burst-Guthaben verloren. Dies geschieht in den folgenden Fällen:
>
> - Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
> - Unveränderliche Updates
> - Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Wenn ein unveränderliches Update fehlschlägt, laden die neuen Instances Bundle-Protokolle auf Amazon S3 hoch, bevor Elastic Beanstalk sie beendet. Elastic Beanstalk hinterlässt eine Stunde lang

Protokolle eines fehlerhaften unveränderlichen Updates in Amazon S3 bevor sie gelöscht werden (anstatt der standardmäßigen 15 Minuten für Bundle- und Tail-Protokolle).

> **Note**
>
> Wenn Sie unveränderliche Updates für die Bereitstellung von Anwendungsversionen verwenden aber nicht für die Konfiguration, erhalten Sie möglicherweise eine Fehlermeldung, wenn Sie versuchen, eine Anwendungsversion bereitzustellen, die Konfigurationsänderungen enthält, die in der Regel ein fortlaufendes Update auslösen würden (beispielsweise Konfigurationen, die den Instance-Typ ändern). Um dies zu vermeiden, machen Sie die Konfigurationsänderung in einem separaten Update oder konfigurieren Sie unveränderliche Updates für Bereitstellungs- und Konfigurationsänderungen.

Es ist nicht möglich, ein unveränderliches Update in Verbindung mit Konfigurationsänderungen durchzuführen. Beispiel: Sie können keine -Einstellungen ändern, die eine Instance-Ersetzung erfordern, während auch andere Einstellungen aktualisiert werden, oder eine unveränderliche Bereitstellung mit Konfigurationsdateien durchführen, die Konfigurationseinstellungen oder zusätzliche Ressourcen in Ihrem Quellcode ändern. Wenn Sie versuchen, Ressourceneinstellungen zu ändern (z. B. Load Balancer-Einstellungen) und gleichzeitig ein unveränderliches Update durchführen, gibt Elastic Beanstalk einen Fehler zurück.

Wenn Ihre Ressourcenkonfigurationsänderungen nicht abhängig von Ihrer Quellcodeänderung oder Instance-Konfiguration sind, führen Sie sie in zwei Updates hinzu. Wenn sie abhängig sind, führen Sie stattdessen eine blaue/grüne Bereitstellung durch.

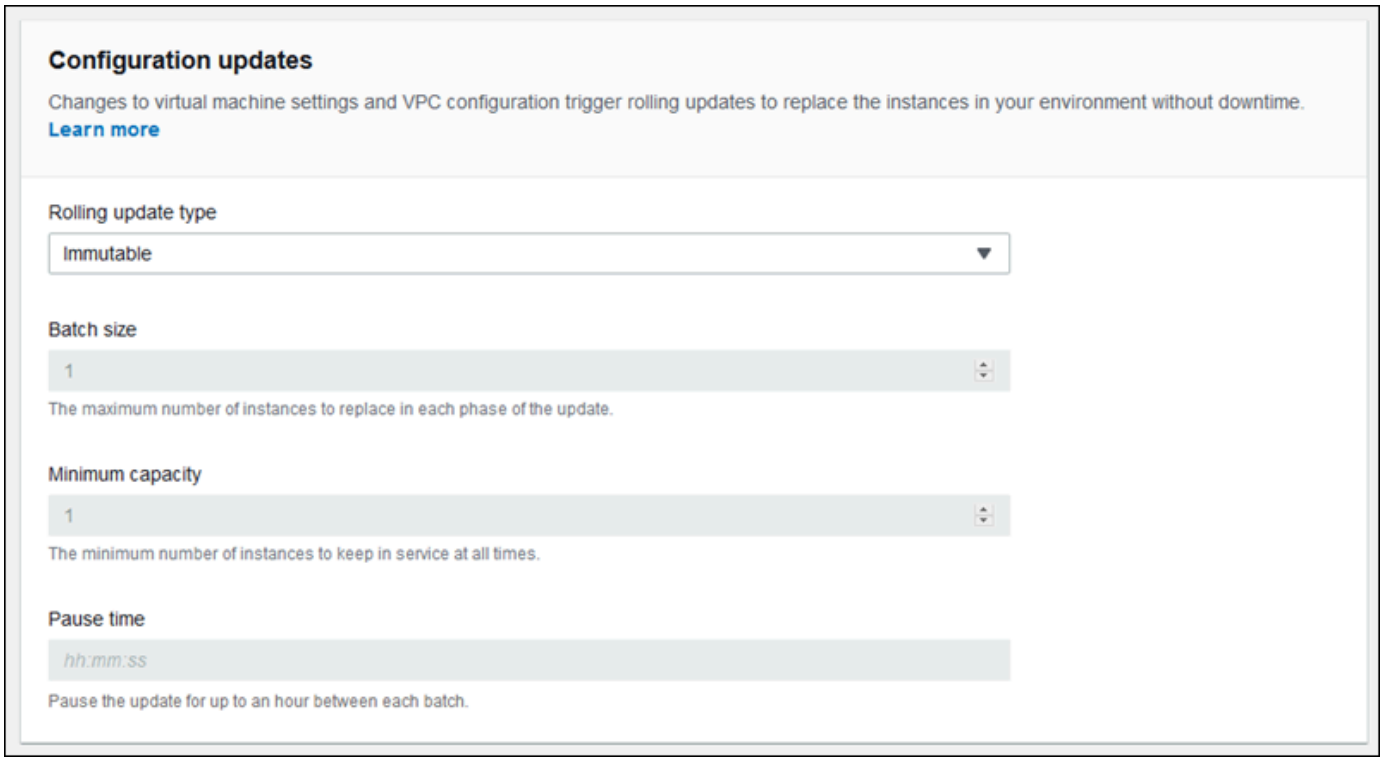
Konfigurieren unveränderlicher Updates

Sie können unveränderliche Updates in der Elastic Beanstalk-Konsole aktivieren und konfigurieren.

So aktivieren Sie unveränderliche Updates (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

4. Wählen Sie in der Konfigurationskategorie Rolling updates and deployments (Fortlaufende Updates und Bereitstellungen) die Option Edit (Bearbeiten).
5. Legen Sie im Abschnitt Configuration Updates Rolling update type auf Immutable (Unveränderlich) fest.



6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Der aws:autoscaling:updatepolicy:rollingupdate-Namespace

Sie können auch die Optionen im `aws:autoscaling:updatepolicy:rollingupdate`-Namespace zum Konfigurieren unveränderlicher Updates verwenden. Die folgende Beispiel-Konfigurationsdatei ermöglicht unveränderliche Updates für Konfigurationsänderungen.

Example.ebextensions/immutable-updates.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Immutable
```

Die folgende Beispiel-Konfigurationsdatei ermöglicht unveränderliche Updates für Konfigurationsänderungen und Bereitstellungen.

Example.ebextensions/immutable-all.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Immutable
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Immutable
```

Die EB-CLI und Elastic-Beanstalk-Konsole wenden die für die vorhergehenden Optionen empfohlenen Werte an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter Empfohlene Werte.

Aktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung

Elastic Beanstalk veröffentlicht regelmäßig neue Plattformversionen, um alle Linux-basierte und Windows Server-basierte Plattformen zu aktualisieren. Neue Plattformversionen bieten Aktualisierungen der vorhandenen Softwarekomponenten und Unterstützung für neue Features und Konfigurationsoptionen. Weitere Informationen zu Plattformen und Plattformversionen finden Sie unter Elastic Beanstalk-Plattformen – Glossar.

Sie können mit der Elastic-Beanstalk-Konsole oder der EB-CLI die Plattformversion der Umgebung aktualisieren. Abhängig von der Plattformversion, die Sie aktualisieren möchten, empfiehlt Elastic Beanstalk eine von zwei Methoden zum Ausführen von Plattformaktualisierungen.

- Methode 1 – Aktualisieren der Plattformversion Ihrer Umgebung. Wir empfehlen diese Methode, wenn Sie auf die neueste Plattformversion in einer Plattformvariante aktualisieren – ohne Änderung an Laufzeit, Webserver, Anwendungsserver, Betriebssystem oder Hauptplattformversion. Dies ist die häufigste und routinemäßige Plattformaktualisierung.
- Methode 2 — Führen Sie eine Bereitstellung durch Blue/Green. Wir empfehlen diese Methode, wenn Sie auf eine Plattformversion in einem anderen Plattformzweig aktualisieren – mit einer anderen Laufzeit, einem anderen Webserver, einem anderen Anwendungsserver oder einem anderen Betriebssystem oder auf eine andere Hauptplattformversion. Dies ist ein guter Ansatz, wenn Sie die Vorteile der neuen Laufzeitfunktionen oder die neueste Elastic-Beanstalk-Funktionalität nutzen möchten oder wenn Sie von einer veralteten oder ausgemusterten Plattformvariante weg wechseln möchten.

Für die Migration von einer älteren Plattformversion ist eine blue/green Bereitstellung erforderlich, da diese Plattformversionen nicht mit den derzeit unterstützten Versionen kompatibel sind.

Die Migration einer Linux-Anwendung auf Amazon Linux 2 erfordert eine blue/green Bereitstellung, da Amazon Linux 2-Plattformversionen nicht mit früheren Amazon Linux AMI-Plattformversionen kompatibel sind.

Weitere Unterstützung bei der Auswahl der optimalen Methode zur Plattformaktualisierung , erweitern Sie den Abschnitt für die Plattform Ihrer Umgebung.

Docker

Verwenden Sie Methode 1 zum Ausführen von Plattformaktualisierungen.

Multicontainer-Docker

Verwenden Sie Methode 1 zum Ausführen von Plattformaktualisierungen.

Vorkonfigurierte Docker-Plattform

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Plattform migrieren, z. B. von Go 1.4 (Docker) auf Go 1.11 oder von Python 3.4 (Docker) auf Python 3.6, verwenden Sie Methode 2.
- Wenn Sie Ihre Anwendung für eine andere Docker-Container-Version migrieren, z. B. von Glassfish 4.1 (Docker) auf Glassfish 5.0 (Docker), verwenden Sie Methode 2.
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Container-Version oder Hauptversion aktualisieren, verwenden Sie Methode 1.

Go

Verwenden Sie Methode 1 zum Ausführen von Plattformaktualisierungen.

Java SE

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Java-Laufzeitversion migrieren, z. B. von Java 7 auf Java 8, verwenden Sie Methode 2.

- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeitversion aktualisieren, verwenden Sie Methode 1.

Java mit Tomcat

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Java-Laufzeitversion oder Tomcat-Anwendungsserver-Version aktualisieren, z. B. von Java 7 mit Tomcat 7 auf Java 8 mit Tomcat 8.5, verwenden Sie Methode 2.
- Wenn Sie Ihre Anwendung auf größeren Java mit Tomcat-Plattformversionen (v1.x.x, v2.x.x, und v3.x.x) migrieren, verwenden Sie Methode 2.
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeitversion, Anwendungsserverversion oder Hauptversion aktualisieren, verwenden Sie Methode 1.

.NET on Windows Server mit IIS

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Windows-Betriebssystemversion migrieren, z. B. von Windows Server 2008 R2 auf Windows Server 2016, verwenden Sie Methode 2.
- Wenn Sie Ihre Anwendung auf großen Windows Server-Plattformversionen migrieren, finden Sie weitere Informationen unter Migration von früheren Hauptversionen der Windows Server-Plattform. Verwenden Sie Methode 2.
- Wenn Ihre Anwendung derzeit auf einer Windows Server-Plattform V2.x.x ausgeführt wird und Sie auf eine aktuelle Plattformversion aktualisieren, verwenden Sie Methode 1.

> **Note**
>
> Windows Server-Plattformversionen vor v2 werden nicht semantisch versioniert. Sie können nur jeweils die neueste Version einer jeden dieser Windows Server-Plattformhauptversionen starten. Ein Rollback nach einem Upgrade ist nicht möglich.

Node.js

Verwenden Sie Methode 2 zum Ausführen von Plattformaktualisierungen.

PHP

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere PHP-Laufzeitversion migrieren, z. B. von PHP 5.6 auf PHP 7.2, verwenden Sie Methode 2.
- Wenn Sie Ihre Anwendung auf große PHP-Plattformversionen (v1.x.x und v2.x.x) migrieren, verwenden Sie Methode 2.
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeit- oder -version aktualisieren, verwenden Sie Methode 1.

Python

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Python-Laufzeitversion migrieren, z. B. von Python 2.7 auf Python 3.6, verwenden Sie Methode 2.
- Wenn Sie Ihre Anwendung auf große Python-Plattformversionen (v1.x.x und v2.x.x) migrieren, verwenden Sie Methode 2.
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeit- oder -version aktualisieren, verwenden Sie Methode 1.

Ruby

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Ruby-Laufzeitversion oder Anwendungsserverversion migrieren, z. B. von Ruby 2.3 mit Puma auf Ruby 2.6 mit Puma migrieren, verwenden Sie Methode 2.
- Wenn Sie Ihre Anwendung auf große Ruby-Plattformversionen (v1.x.x und v2.x.x) migrieren, verwenden Sie Methode 2.
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeitversion, Anwendungsserverversion oder Hauptversion aktualisieren, verwenden Sie Methode 1.

Methode 1 – Aktualisieren der Plattformversion Ihrer Umgebung

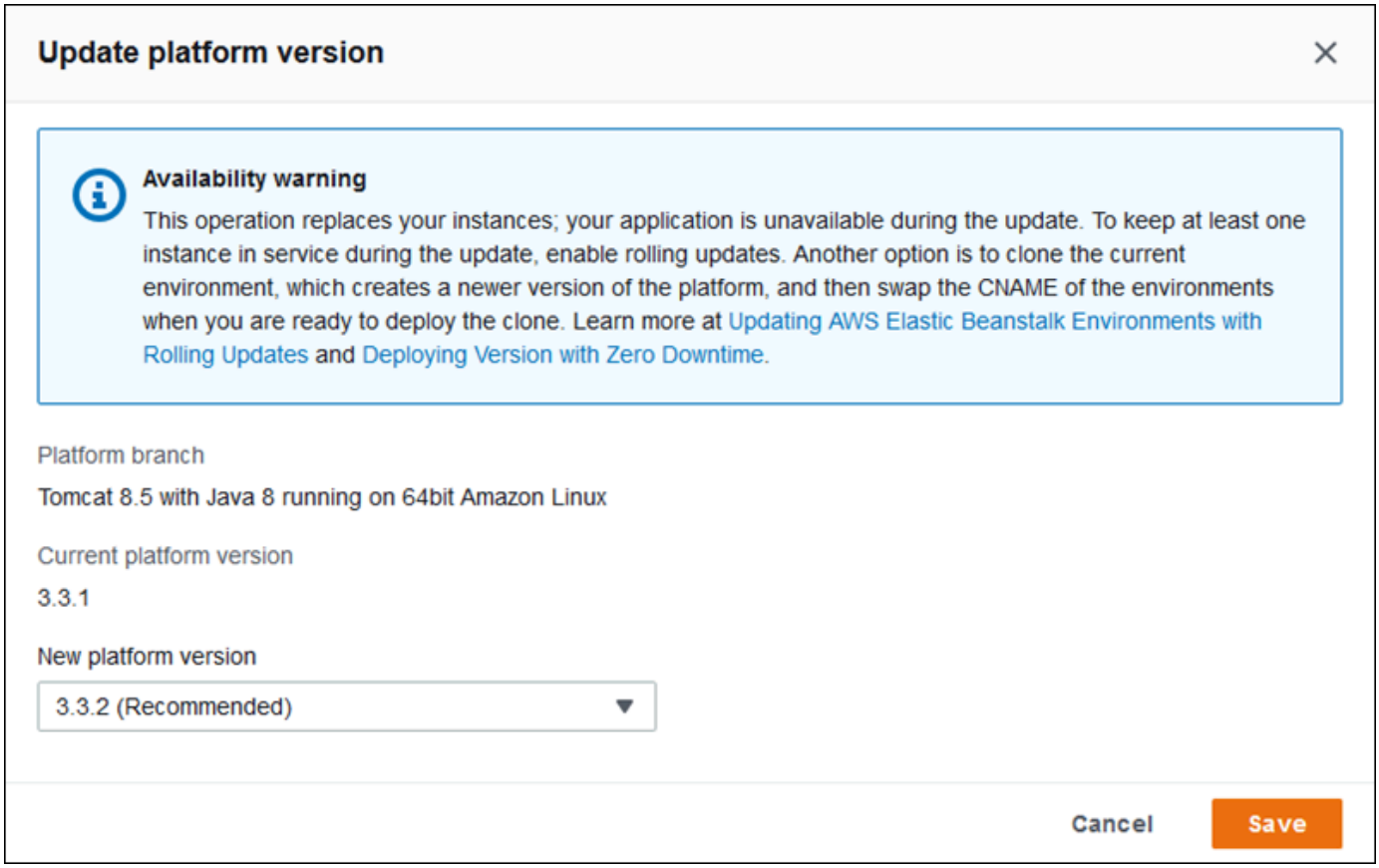
Verwenden Sie diese Methode zum Aktualisieren auf die aktuelle Plattformvariante Ihrer Umgebung. Falls Sie zuvor eine Umgebung mit einer älteren Plattformversion erstellt oder Ihre Umgebung von einer älteren Version aktualisiert haben, können Sie mit dieser Methode eine vorherige Plattformversion wiederherstellen, sofern sich diese in derselben Plattformvariante befindet.

So aktualisieren Sie die Plattformversion Ihrer Umgebung:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite unter Platform (Plattform) die Option Change (Ändern) aus.



4. Wählen Sie im Dialogfeld Update platform version (Plattformversion aktualisieren) eine Plattformversion aus. Die neueste (empfohlene) Plattformversion in der Variante wird automatisch ausgewählt. Sie können auf jede Version aktualisieren, die Sie in der Vergangenheit verwendet haben.



5. Wählen Sie Save aus.

Lassen Sie die Plattformaktualisierungen von Elastic Beanstalk verwalten, um den Vorgang zu vereinfachen. Sie können die Umgebung so konfigurieren, dass während eines definierbaren wöchentlichen Wartungsfensters automatisch Neben- und Patch-Versionsaktualisierungen durchgeführt werden. Elastic Beanstalk führt verwaltete Aktualisierungen ohne Ausfallzeit oder Kapazitätseinbußen aus. Falls Instances, auf denen die Anwendung mit der neuen Version ausgeführt wird, die Zustandsprüfungen nicht bestehen, wird die Aktualisierung unverzüglich abgebrochen. Details hierzu finden Sie unter Verwaltete Plattformaktualisierungen.

Methode 2 — Führen Sie eine Bereitstellung durch Blue/Green

Verwenden Sie diese Methode, um auf eine andere Plattformvariante zu aktualisieren – mit anderer Laufzeit, anderem Webserver, Anwendungsserver oder Betriebssystem oder einer anderen Hauptplattformversion. Dies ist in der Regel erforderlich, wenn Sie die Vorteile der neuen Laufzeitfunktionen oder der aktuellen Elastic-Beanstalk-Funktionalität nutzen möchten. Es ist auch erforderlich, wenn Sie von einer veralteten oder ausgemusterten Plattformvariante weg migrieren.

Wenn Sie auf größere Plattformversionen oder auf Plattformversionen mit wichtigen Komponentenaktualisierungen aktualisieren möchten, ist die Wahrscheinlichkeit größer, dass Ihre Anwendung oder einige ihrer Aspekte, auf der neuen Plattformversion nicht wie erwartet funktionieren und möglicherweise Änderungen erfordern.

Aktualisieren Sie Ihren lokalen Entwicklungscomputer vor dem Ausführen der Migration auf neuere Laufzeitversionen und andere Komponenten der Plattform, die Sie migrieren wollen. Stellen Sie sicher, dass Ihre Anwendung weiterhin wie erwartet funktioniert und nehmen Sie bei Bedarf Codekorrekturen und -änderungen vor. Verwenden Sie dann die folgende bewährte Methode zur sicheren Migration Ihrer Umgebung auf die neue Plattformversion.

So migrieren Sie Ihre Umgebung auf eine Plattformversion mit wichtigen Aktualisierungen

1. Erstellen Sie eine neue Umgebung unter Verwendung der neuen Plattformzielversion und stellen Sie für sie Ihren Anwendungscode bereit. Die neue Umgebung sollte sich in der Elastic-Beanstalk-Anwendung befinden, die die Umgebung enthält, in die Sie migrieren. Beenden Sie die vorhandene Umgebung noch nicht.
2. Verwenden Sie die neue Umgebung, um Ihre Anwendung zu migrieren. Insbesondere:

   - Suchen und beheben Sie Probleme mit der Anwendungskompatibilität, die Sie während der Entwicklungsphase nicht erkennen konnten.
   - Stellen Sie sicher, dass alle Anpassungen, die Ihre Anwendung mithilfe von Konfigurationsdateien vornimmt, in der neuen Umgebung korrekt funktionieren. Hierbei kann es sich z. B. um Optionseinstellungen, zusätzliche installierte Pakete, benutzerdefinierte Sicherheitsrichtlinien und Skripts oder auf Umgebungs-Instances installierte Konfigurationsdateien handeln.
   - Wenn Ihre Anwendung ein benutzerdefiniertes Amazon Machine Image (AMI) verwendet, erstellen Sie ein neues benutzerdefiniertes AMI basierend auf dem AMI der neuen Plattformversion. Weitere Informationen hierzu finden Sie unter Verwenden eines benutzerdefinierten Amazon Machine Images (AMI) in Ihrer Elastic Beanstalk Beanstalk-Umgebung. Dies ist insbesondere erforderlich, wenn Ihre Anwendung die Windows Server-Plattform mit einem benutzerdefinierten AMI nutzt und Sie eine Migration auf eine Windows Server V2-Plattformversion vornehmen. Weitere Informationen über diesen Fall finden Sie auch unter Migration von früheren Hauptversionen der Windows Server-Plattform.

   Führen Sie so lange Tests durch und stellen Sie Ihre Korrekturen bereit, bis Sie mit der Anwendung in der neuen Umgebung zufrieden sind.

3. Machen Sie die neue Umgebung zu Ihrer Produktionsumgebung, indem Sie ihren CNAME mit dem CNAME der vorhandenen Produktionsumgebung vertauschen. Details hierzu finden Sie unter Blau/Grün-Bereitstellungen mit Elastic Beanstalk.
4. Wenn Sie mit dem Status Ihrer neuen Umgebung in der Produktion zufrieden sind, schließen Sie die alte Umgebung. Details hierzu finden Sie unter Elastic Beanstalk-Umgebung terminieren.

Verwaltete Plattformaktualisierungen

AWS Elastic Beanstalk veröffentlicht regelmäßig Plattform-Updates, um Korrekturen, Softwareupdates und neue Funktionen bereitzustellen. Mit verwalteten Plattformaktualisierungen können Sie die Umgebung so konfigurieren, dass während eines geplanten Wartungsfensters automatisch eine Aktualisierung auf die neueste Version einer Plattform durchgeführt wird. Die Anwendung bleibt während der Aktualisierung aktiv, und zwar ohne Kapazitätseinbußen. Verwaltete Aktualisierungen stehen sowohl für Umgebungen mit einer Instance als auch mit Load Balancing zur Verfügung.

> **Note**
>
> Diese Funktion steht bei Windows Server-Plattformversionen vor Version 2 (v2) nicht zur Verfügung.

Sie können die Umgebung zur automatischen Ausführung von Patch-Versionsaktualisierungen oder sowohl Patch- als auch Nebenversionsaktualisierungen konfigurieren. Verwaltete Plattformaktualisierungen unterstützen keine Aktualisierungen über Plattformvarianten hinweg (Aktualisierungen für verschiedene Hauptversionen von Plattformkomponenten wie beispielsweise Betriebssystem, Laufzeit oder Elastic-Beanstalk-Komponenten), da diese zu nicht abwärtskompatiblen Änderungen führen können.

Sie können Elastic Beanstalk auch so konfigurieren, dass alle Instances der Umgebung während des Wartungsfensters ersetzt werden, auch wenn keine Plattformaktualisierung verfügbar ist. Das Ersetzen aller Instances der Umgebung ist sinnvoll, wenn die Anwendung über einen langen Zeitraum ausgeführt wird und Fehler oder Speicherprobleme auftreten.

In Umgebungen, die am 25. November 2019 oder später mit der Elastic-Beanstalk-Konsole erstellt wurden, werden verwaltete Updates standardmäßig aktiviert (sofern möglich). Verwaltete Updates erfordern die Aktivierung einer verbesserten Integrität . Die erweiterte Integrität ist standardmäßig aktiviert, wenn Sie eine der Konfigurationsvoreinstellungen auswählen, und deaktiviert, wenn Sie

Custom configuration (Benutzerdefinierte Konfiguration) auswählen. Die Konsole kann verwaltete Updates nicht für ältere Plattformversionen aktivieren, die den erweiterten Status nicht unterstützen, oder wenn die erweiterte Integrität deaktiviert ist. Wenn die Konsole verwaltete Updates für eine neue Umgebung aktiviert, wird das Weekly update window (Wöchentliche Aktualisierungsfenster) auf einen zufälligen Zeitpunkt an einem zufälligen Wochentag eingestellt. Der Wert für Update level (Aktualisierungsstufe) ist auf Minor und patch (Nebenstufe und Patch) eingestellt und Instance replacement (Instance-Ersetzung) ist deaktiviert. Sie können verwaltete Updates vor dem letzten Schritt zur Umgebungserstellung deaktivieren oder neu konfigurieren.

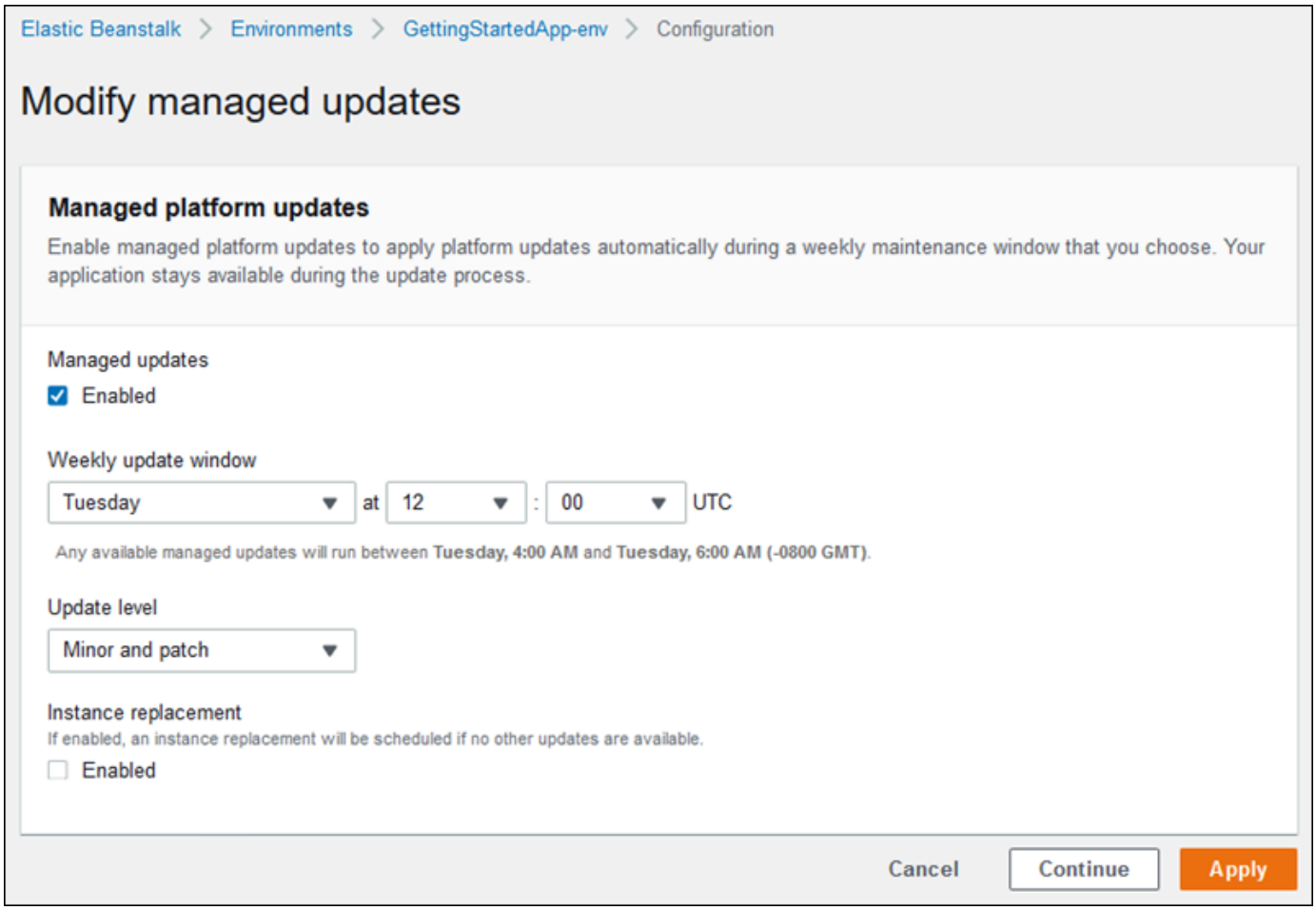
Für eine vorhandene Umgebung können Sie verwaltete Plattformaktualisierungen jederzeit mit der Elastic-Beanstalk-Konsole konfigurieren.

> **Important**
>
> Ein hohe Zahl von Beanstalk-Umgebungen in einem AWS -Konto können während verwalteter Updates zu Drosselungsproblemen führen. Hohe Zahl ist ein relativer Begriff, und die Anzahl hängt davon ab, wie genau Sie die verwalteten Updates für Ihre Umgebungen planen. Bei über 200 Umgebungen in einem Konto, die eng aufeinander abgestimmt sind, kann es zu Drosselungsproblemen kommen, obwohl auch schon eine geringere Anzahl von Umgebungen mit Problemen einhergehen kann.
> Um die Ressourcenbelastung für verwaltete Updates auszugleichen, empfehlen wir, die geplanten Wartungsfenster für die Umgebungen eines Kontos zu verteilen.
> Sie sollten auch eine Strategie mit mehreren Konten in Erwägung ziehen. Weitere Informationen finden Sie unter Organisieren Ihrer AWS Umgebung mithilfe mehrerer Konten auf der AWS Whitepapers & Guides-Website.

So konfigurieren Sie verwaltete Plattformaktualisierungen

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Kategorie Managed Updates (Verwaltete Updates) die Option Edit (Bearbeiten) aus.
5. Deaktivieren oder aktivieren Sie Managed updates (Verwaltete Updates).

6. Wenn verwaltete Updates aktiviert sind, wählen Sie ein Wartungsfenster aus und danach eine Einstellung unter Update level (Aktualisierungsstufe) aus.
7. (Optional) Wählen Sie Instance replacement (Instance-Ersetzung) aus, um den wöchentlichen Instance-Ersatz zu aktivieren.



8. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Verwaltete Plattformaktualisierungen erfordern erweiterte Zustandsberichte, um zu bestimmen, ob die Anwendung stabil genug ist und die Plattformaktualisierung als erfolgreich eingestuft werden kann. Detaillierte Anweisungen finden Sie unter Aktivieren der erweiterten Elastic-Beanstalk-Integritätsberichte.

Abschnitte

- Zur Durchführung von verwalteten Plattformaktualisierungen erforderliche Berechtigungen
- Wartungsfenster für verwaltete Aktualisierung

- Neben- und Patch-Versionsaktualisierungen
- Unveränderliche Umgebungsaktualisierungen
- Verwalten von verwalteten Aktualisierungen
- Namespaces mit der Option für verwaltete Aktionen

Zur Durchführung von verwalteten Plattformaktualisierungen erforderliche Berechtigungen

Elastic Beanstalk benötigt die Berechtigung, um in Ihrem Namen eine Plattformaktualisierung zu initiieren. Um diese Berechtigungen zu erhalten, übernimmt Elastic Beanstalk die Servicerolle für verwaltete Aktualisierungen. Wenn Sie die standardmäßige Servicerolle für Ihre Umgebung verwenden, verwendet die Elastic-Beanstalk-Konsole diese auch als Servicerolle für verwaltete Aktualisierungen. Die Konsole weist der Servicerolle die `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`-verwaltete Richtlinie zu. Diese Richtlinie verfügt über alle Berechtigungen, die Elastic Beanstalk zum Ausführen verwalteter Plattformaktualisierungen benötigt.

Details zu weiteren Möglichkeiten für die Einrichtung einer Servicerolle für verwaltete Updates finden Sie unter the section called “Servicerollen”.

> **Note**
>
> Wenn Sie Konfigurationsdateien verwenden, um die Umgebung zu erweitern und zusätzliche Ressourcen einzufügen, müssen Sie der Servicerolle für verwaltete Updates in Ihrer Umgebung möglicherweise zusätzliche Berechtigungen hinzufügen. In der Regel ist das Hinzufügen von Berechtigungen erforderlich, wenn namentlich in anderen Abschnitten oder Dateien auf diese Ressourcen verwiesen wird.

Wenn eine Aktualisierung nicht erfolgreich ist, können Sie den Grund auf der Seite Managed Updates (Verwaltete Updates) finden.

Wartungsfenster für verwaltete Aktualisierung

Wenn eine neue Version der Plattform Ihrer Umgebung AWS veröffentlicht wird, plant Elastic Beanstalk im nächsten wöchentlichen Wartungsfenster ein Update der verwalteten Plattform. Ein

Wartungsfenster überspannt einen Zeitraum von zwei Stunden. Elastic Beanstalk startet während des Wartungsfensters eine zeitgesteuerte Aktualisierung. Die Aktualisierung ist möglicherweise erst nach dem Ende des Fensters abgeschlossen.

> **Note**
>
> In den meisten Fällen plant Elastic Beanstalk Ihre verwaltete Aktualisierung für das nächste wöchentliche Wartungsfenster ein. Das System berücksichtigt die verschiedenen Aspekte der Sicherheit der Aktualisierung und der Serviceverfügbarkeit beim Planen verwalteter Updates. In seltenen Fällen kann eine Aktualisierung nicht für das nächste anstehende Wartungsfenster geplant werden. In diesem Fall versucht es das System während des nächsten Wartungsfensters erneut. Zur manuellen Anwendung der verwalteten Aktualisierung wählen Sie Apply now (Jetzt anwenden) wie unter Verwalten von verwalteten Aktualisierungen auf dieser Seite erläutert.

Neben- und Patch-Versionsaktualisierungen

Sie können verwaltete Plattformaktualisierungen so einstellen, dass nur Patch-Versionsaktualisierungen oder sowohl Neben- als auch Patch-Versionsaktualisierungen angewendet werden. Patch-Versionsaktualisierungen sorgen für Fehlerbehebungen sowie Leistungsverbesserungen und können geringfügige Konfigurationsänderungen der Software, Skripts und Konfigurationsoptionen für Instances umfassen. Nebenversionsaktualisierungen liefern die Unterstützung für neue Elastic-Beanstalk-Funktionen. Mit verwalteten Plattformaktualisierungen können Sie keine Hauptversionsaktualisierungen ausführen, da mit diesen möglicherweise abwärts inkompatible Änderungen eingeführt werden.

Bei der Plattformversionsnummer steht die zweite Ziffer für die Nebenversionsaktualisierung und die dritte Ziffer bezeichnet die Patch-Version. Beispielsweise gibt die Plattformversion 2.0.7 die Nebenversion 0 und die Patch-Version 7 an.

Unveränderliche Umgebungsaktualisierungen

Verwaltete Plattformaktualisierungen führen unveränderliche Umgebungsaktualisierungen aus, um die Umgebung auf eine neue Plattformversion zu aktualisieren. Bei unveränderlichen Aktualisierungen wird die Umgebung ohne Instance-Ausfallzeit und Umgebungsänderungen aktualisiert, bevor bestätigt ist, dass die Instances mit der neuen Version die Zustandsprüfungen bestehen.

Elastic Beanstalk stellt bei einer unveränderlichen Aktualisierung die neue Plattformversion auf allen zu dem Zeitpunkt ausgeführten Instances bereit. Die neuen Instances verarbeiten die Anforderungen gemeinsam mit den alten Instances. Sofern die neuen Instances alle Zustandsprüfungen bestehen, beendet Elastic Beanstalk die alten Instances, sodass nur Instances mit der neuen Version ausgeführt werden.

Verwaltete Plattformaktualisierungen führen immer unveränderliche Aktualisierungen aus, auch wenn Sie diese außerhalb des Wartungsfensters anwenden. Wenn Sie die Plattformversion im Dashboard ändern, übernimmt Elastic Beanstalk die ausgewählte Richtlinie für Konfigurationsaktualisierungen.

> **Warning**
>
> Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle angesammelten Amazon-EC2-Steigerungsguthaben verloren. Dies geschieht in den folgenden Fällen:
>
> - Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
> - Unveränderliche Updates
> - Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Verwalten von verwalteten Aktualisierungen

Die Elastic-Beanstalk-Konsole zeigt detaillierte Informationen zu verwalteten Updates auf der Seite Managed Updates overview (Übersicht über verwaltete Aktualisierungen) an.

So zeigen Sie Informationen zu verwalteten Aktualisierungen an (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Managed Updates (Verwaltete Updates).

Im Abschnitt Managed Updates overview (Übersicht über die verwalteten Aktualisierungen) finden Sie Informationen zu geplanten und ausstehenden verwalteten Aktualisierungen. Unter History (Verlauf) werden erfolgreiche Aktualisierungen und Fehlversuche aufgeführt.

Sie können eine geplante Aktualisierung auch sofort ausführen (anstatt auf das nächste Wartungsfenster zu warten).

So übernehmen Sie eine verwaltete Plattformaktualisierung sofort (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Managed Updates (Verwaltete Updates).
4. Wählen Sie Apply now (Jetzt anwenden) aus.
5. Überprüfen Sie die Aktualisierungsdetails und wählen Sie dann Apply (Übernehmen).

Bei einer verwalteten Plattformaktualisierung außerhalb des Wartungsfensters führt Elastic Beanstalk eine unveränderliche Aktualisierung aus. Wenn Sie die Umgebungsplattform im Dashboard oder mit einem anderen Client aktualisieren, verwendet Elastic Beanstalk den für Konfigurationsänderungen ausgewählten Aktualisierungstyp.

Falls keine verwaltete Aktualisierung geplant ist, wird die Umgebung vermutlich bereits mit der neuesten Version ausgeführt. Es gibt auch andere Gründe dafür, dass keine Aktualisierung geplant ist:

- Eine Nebenversionsaktualisierung ist verfügbar, aber die Umgebung ist nur für die automatische Ausführung von Patch-Versionsaktualisierungen konfiguriert.
- Die Umgebung wurde seit der Veröffentlichung der Aktualisierung nicht geprüft. Elastic Beanstalk prüft in der Regel einmal pro Stunde, ob eine Aktualisierung vorhanden ist.
- Eine Aktualisierung steht aus oder wird gerade ausgeführt.

Mit dem Beginn des Wartungsfensters oder wenn Sie auf Apply now (Jetzt anwenden) klicken, wechseln geplante Aktualisierungen vor der Ausführung in den Status "Pending (Anstehend)".

Namespaces mit der Option für verwaltete Aktionen

Mit den Konfigurationsoptionen in den Namespaces `aws:elasticbeanstalk:managedactions` und `aws:elasticbeanstalk:managedactions:platformupdate` können Sie verwaltete Plattformaktualisierungen aktivieren und konfigurieren.

Mithilfe der Option `ManagedActionsEnabled` werden verwaltete Plattformaktualisierungen aktiviert. Legen Sie für diese Option den Wert `true` fest, um verwaltete Plattformaktualisierungen zu aktivieren, und konfigurieren Sie mit den anderen Optionen das Aktualisierungsverhalten.

Verwenden Sie diese `PreferredStartTime` Option, um den Beginn des wöchentlichen Wartungsfensters im*day*:*hour*: *minute* -Format zu konfigurieren.

Legen Sie `UpdateLevel` auf `minor` oder `patch` fest, damit sowohl Neben- als auch Patch-Versionsaktualisierungen bzw. nur Patch-Versionsaktualisierungen ausgeführt werden.

Wenn verwaltete Plattformaktualisierungen aktiviert sind, können Sie den Instance-Ersatz aktivieren. Wählen Sie dazu für die Option `InstanceRefreshEnabled` den Wert `true` aus. Ist diese Einstellung aktiviert, führt Elastic Beanstalk einmal pro Woche eine unveränderliche Aktualisierung Ihrer Umgebung aus, und zwar unabhängig davon, ob eine neue Plattformversion zur Verfügung steht.

Im folgenden Beispiel einer Konfigurationsdatei werden verwaltete Plattformaktualisierungen für Patch-Versionsaktualisierungen in einem Wartungsfenster, das jeden Dienstag um 9:00 Uhr koordinierter Weltzeit (UTC) beginnt, aktiviert.

Example.ebeextensions/ .config managed-platform-update

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: patch
    InstanceRefreshEnabled: true
```

Migration Ihrer Anwendung von einer Legacy-Plattformversion

Wenn Sie eine Elastic-Beanstalk-Anwendung bereitgestellt haben, die eine Legacy-Plattformversion verwendet, sollten Sie Ihre Anwendung in eine neue Umgebung mit einer nicht veralteten Plattformversion migrieren, sodass Sie Zugriff auf neue Funktionen erhalten. Wenn Sie sich nicht sicher sind, ob Sie Ihre Anwendung mit einer alten Plattformversion ausführen, können Sie dies in der Elastic-Beanstalk-Konsole überprüfen. Detaillierte Anweisungen finden Sie unter So prüfen Sie, ob Sie eine ältere Plattformversion verwenden.

Welche neuen Funktionen fehlen in älteren Plattformversionen?

Ältere Plattformen unterstützen die folgenden Funktionen nicht:

- Konfigurationsdateien, wie im Thema Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`) beschrieben
- ELB-Zustandsprüfungen, wie im Thema Grundlegende Zustandsberichte beschrieben
- Instance-Profile, wie im Thema Elastic Beanstalk Instance-Profile verwalten beschrieben
- VPCs, wie im Verwenden von Elastic Beanstalk mit Amazon VPC Thema beschrieben
- Datenschichten, wie im Thema Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung beschrieben
- Worker-Schichten, wie im Thema Elastic-Beanstalk-Worker-Umgebungen beschrieben
- Umgebungen mit einer Instance, wie im Thema Typen von Umgebungen beschrieben
- Tags, wie im Thema Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren beschrieben
- Fortlaufende Updates, wie im Thema Fortlaufende Updates der Elastic Beanstalk-Umgebungskonfiguration beschrieben

Warum sind einige Plattformversionen als Legacy gekennzeichnet?

Von einigen älteren Plattformversionen werden die neuesten Elastic-Beanstalk-Funktionen nicht unterstützt. Diese Versionen sind als auf der Umgebungsübersichtsseite in der Elastic-Beanstalk-Konsole mit (legacy) ((veraltet)) gekennzeichnet.

So prüfen Sie, ob Sie eine ältere Plattformversion verwenden

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Zeigen Sie auf der Umgebungsübersichtsseite den Namen der Platform (Plattform) an.

   Ihre Anwendung verwendet eine ältere Plattformversion, wenn neben dem Plattformnamen (legacy) (veraltet) angezeigt wird.

So migrieren Sie Ihre Anwendung

1. Stellen Sie Ihre Anwendung in einer neuen Umgebung bereit. Anweisungen finden Sie unter Erstellen einer Elastic Beanstalk-Umgebung.
2. Wenn Sie eine Amazon-RDS-DB-Instance haben, aktualisieren Sie die Sicherheitsgruppe Ihrer Datenbank, um den Zugriff auf Ihre EC2-Sicherheitsgruppe für Ihre neue Umgebung zuzulassen. Anweisungen, wie Sie den Namen Ihrer EC2-Sicherheitsgruppe mithilfe der AWS Management Console ermitteln können, finden Sie unter. EC2 Sicherheitsgruppen Weitere Informationen zum Konfigurieren Ihrer EC2-Sicherheitsgruppe finden Sie im Abschnitt "Autorisieren von Netzwerkzugriff auf eine Amazon-EC2-Sicherheitsgruppe" unter Arbeiten mit DB-Sicherheitsgruppen im Amazon-Relational-Database-Service-Benutzerhandbuch.
3. Tauschen Sie Ihre Umgebungs-URL aus. Anweisungen finden Sie unter Blau/Grün-Bereitstellungen mit Elastic Beanstalk.
4. Beenden Sie Ihre alte Umgebung. Anweisungen finden Sie unter Elastic Beanstalk-Umgebung terminieren.

> **Note**
>
> Wenn Sie AWS Identity and Access Management (IAM) verwenden, müssen Sie Ihre Richtlinien so aktualisieren, dass sie Amazon RDS CloudFormation einschließen (falls zutreffend). Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit AWS Identity and Access Management.

Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2

In diesem Abschnitt wird beschrieben, wie Sie die Anwendung mit einem der folgenden Migrationspfade migrieren.

- Migrieren Sie von einem Amazon-Linux-2- zu einem Amazon-Linux-2023-Plattformzweig.
- Migrieren Sie von einem Amazon Linux AMI (AL1) -Plattformzweig zu einem Amazon Linux 2023 (empfohlen) oder einem Amazon Linux 2-Plattformzweig.

Themen

- Migration von Amazon Linux 2 zu Amazon Linux 2023
- Migration von Amazon Linux AMI (AL1) zu AL2 oder AL2 023

Migration von Amazon Linux 2 zu Amazon Linux 2023

Dieses Thema bietet eine Anleitung zum Migrieren Ihrer Anwendung von einem Amazon-Linux-2-Plattformzweig zu einem Amazon-Linux-2023-Plattformzweig.

Unterschiede und Kompatibilität

Zwischen den Plattformen Elastic Beanstalk AL2 und 023 AL2

Es besteht ein hohes Maß an Kompatibilität zwischen den Plattformen Elastic Beanstalk Amazon Linux 2 und Amazon Linux 2023. Es gibt jedoch einige Unterschiede zu beachten:

- Instance Metadata Service Version 1 (IMDSv1) — Die IMDSv1 Option „Deaktivieren“ ist auf 023-Plattformen standardmäßig aktiviert. `true` AL2 Die Standardeinstellung ist `false` auf Plattformen. AL2
- pkg-repo-Instanztool — Das Pkg-Repo Tool ist nicht für Umgebungen verfügbar, die auf AL2 023-Plattformen ausgeführt werden. Sie können Paket- und Betriebssystemupdates jedoch weiterhin manuell auf eine 023-Instanz anwenden. AL2 Weitere Informationen finden Sie unter Verwaltung von Paketen und Betriebssystem-Updates im Benutzerhandbuch für Amazon Linux 2023.
- HTTPd Apache-Konfiguration — Die `httpd.conf` Apache-Datei für AL2 023-Plattformen enthält einige Konfigurationseinstellungen, die sich von denen für folgende unterscheiden: AL2
  - Verweigern Sie standardmäßig den Zugriff auf das gesamte Dateisystem des Servers. Diese Einstellungen werden unter Serverdateien standardmäßig schützen auf der Seite mit den Sicherheitstipps der Apache-Website beschrieben.
  - Verweigert den Zugriff auf die Einrichtung von `.htaccess` in allen Verzeichnissen, mit Ausnahme der speziell aktivierten. Diese Einstellung wird unter Schützen von Systemeinstellungen auf der Seite mit den Sicherheitstipps der Apache-Website beschrieben. Auf der Seite Server-Tutorial für Apache HTTP: .htaccess-Dateien heißt es, dass diese Einstellung zur Verbesserung der Leistung beitragen kann.
  - Verweigern Sie den Zugriff auf Dateien mit einem Namensmuster `.ht*`. Diese Einstellung verhindert, dass Webclients alle `.htaccess`- und `.htpasswd`-Dateien anzeigen.

  Sie können jede der oben genannten Konfigurationseinstellungen für die Umgebung ändern. Weitere Informationen finden Sie unter Konfigurieren von Apache HTTPD.

- Unterstützung für mehrzeilige Umgebungsvariablen — AL2 023 Plattformen unterstützen mehrzeilige Werte für Umgebungsvariablen und geheime Schlüssel in Systemd-Dienstkonfigurationen. Amazon Linux 2-Plattformen unterstützen keine mehrzeiligen Umgebungsvariablenwerte. Diese Erweiterung ermöglicht es Ihnen, mehrzeilige Geheimnisse und Konfigurationswerte auf AL2 023-Plattformen zu verwenden. Weitere Hinweise zur Verwendung von Umgebungsvariablen und Geheimnissen finden Sie unter. the section called “Mehrzeilige Werte”
- CloudWatch benutzerdefinierte Protokollweiterleitung — Der veraltete CloudWatch Logs-Agent (`awslogs`Paket) ist auf AL2 023-Plattformen nicht verfügbar. Wenn Sie über benutzerdefinierte Protokollweiterleitungskonfigurationen verfügen, die den veralteten `awslogs` Agenten installieren und verwenden, müssen Sie Ihre Konfigurationsdateien aktualisieren, um den Unified CloudWatch Agent bei der Migration von Amazon Linux 2 auf 023 zu verwenden. AL2 Weitere Informationen finden Sie unter the section called “Benutzerdefiniertes Streaming von Protokolldateien”.

Plattformspezifische Unterschiede

Zusätzlich zu den grundlegenden Betriebssystemunterschieden gibt es plattformspezifische Unterschiede zwischen den Laufzeitplattformen Amazon Linux 2 und AL2 023:

- .NET-Plattformverzweigung — Die Branching-Strategie für die .NET-Plattform unterscheidet sich zwischen Amazon Linux 2 und AL2 023. Auf Amazon Linux 2 unterhält die .NET Core-Plattform ein rotierendes Fenster mit den.NET-Hauptversionen innerhalb eines einzigen Plattformzweigs. Bei AL2 023 ist jeder Plattformzweig an eine spezifische.NET-Hauptversion gebunden (z. B.. NET 9, .NET 10).

  Wenn Sie Framework-abhängige Anwendungen (Anwendungen, die auf der installierten .NET-Laufzeit der Plattform basieren) bereitstellen, müssen Sie einen Plattformzweig auswählen, der der .NET-Zielversion Ihrer Anwendung entspricht. Wenn Sie eigenständige Anwendungen (Anwendungen, die ihre eigene .NET-Laufzeit bündeln) bereitstellen, können Sie unabhängig von der.NET-Version Ihrer Anwendung einen beliebigen AL2 023-.NET-Plattformzweig verwenden, da Ihre Anwendung nicht von der installierten Laufzeit der Plattform abhängig ist. Weitere Informationen finden Sie unter the section called “Anwendungsbündelung”.
- Versionsauswahl für Node.js — Die Plattform Node.js auf Amazon Linux 2 unterstützt die Angabe einer Node.js -Version in der `package.json` Datei Ihrer Anwendung. Die Node.js -Plattform auf AL2 Version 023 unterstützt diese Funktion nicht. Sie müssen die Standardversion von Node.js verwenden, die vom Plattformzweig bereitgestellt wird. Weitere Informationen

zur Versionsverwaltung von Node.js finden Sie unterthe section called “Konfigurieren von Abhängigkeiten”.

- Ruby Puma-Serverversion — Die Ruby-Plattform auf Amazon Linux 2 ignoriert die in der `Gemfile.lock` Datei Ihrer Anwendung angegebene Puma-Version und verwendet die Puma-Standardversion der Plattform. Die Ruby-Plattform auf AL2 0.23 berücksichtigt die in angegebene Puma-Version, falls vorhanden. `Gemfile.lock` Wenn keine Version angegeben ist, installiert die Plattform die Puma-Standardversion der Plattform.
- Verfügbarkeit von PHP-Paketen — Einige Pakete, die auf Amazon Linux 2 PHP-Plattformen verfügbar sind, sind auf AL2 023 PHP-Plattformen nicht verfügbar:
  - MySQL-Client-Pakete — Die Client-Pakete `mysql` und die `mysql-devel` Befehlszeilen-Client-Pakete sind auf AL2 023 PHP-Plattformen nicht installiert. Wenn für Ihre Anwendung eine MySQL-Datenbankverbindung erforderlich ist, verwenden Sie PHP `mysqli` oder `pdo_mysql` Erweiterungen, die auf beiden Plattformen verfügbar sind.
  - Compass- und Ruby-Tools — Die `ruby-devel` `rubygems` Pakete für die Compass CSS-Framework-Unterstützung sind auf AL2 023 PHP-Plattformen nicht installiert. Compass ist veraltet. Erwägen Sie die Verwendung moderner CSS-Vorverarbeitungstools als Alternative.
- Tools zur Go-Versionskontrolle — Das Bazaar-Versionskontrollsystem (`bzr`) ist auf AL2 023 Go-Plattformen nicht verfügbar. Bazaar ist veraltet und nicht im 023-Paket-Repository enthalten. AL2 Verwenden Sie stattdessen Git, Mercurial oder Subversion für die Versionskontrolle, die alle auf AL2 023 Go-Plattformen verfügbar sind.

Zwischen den Amazon-Linux-Betriebssystemen

Weitere Informationen zu den Unterschieden zwischen den Betriebssystemen Amazon Linux 2 und Amazon Linux 2023 finden Sie unter Vergleich von Amazon Linux 2 und Amazon Linux 2023 im Benutzerhandbuch von Amazon Linux 2023.

Weitere Informationen zu Amazon Linux 2023 finden Sie unter Was ist Amazon Linux 2023? im Benutzerhandbuch von Amazon Linux 2023.

Allgemeiner Migrationsprozess

Wenn Sie bereit sind, mit der Produktion zu beginnen, benötigt Elastic Beanstalk ein blue/green Deployment, um das Upgrade durchzuführen. Im Folgenden finden Sie die allgemeinen Best-Practice-Schritte, die wir für die Migration mit einem blue/green Bereitstellungsverfahren empfehlen.

Vorbereitung der Tests für die Migration

Lesen Sie sich die Informationen im vorherigen Abschnitt Unterschiede und Kompatibilität durch, bevor Sie Ihre Anwendung bereitstellen und mit dem Testen beginnen. Lesen Sie außerdem das darin genannte Kapitel Vergleich von Amazon Linux 2 und Amazon Linux 2023 im Benutzerhandbuch von Amazon Linux 2023. Notieren Sie sich die spezifischen Informationen, die (möglicherweise) für Ihre Anwendung und Konfiguration gelten.

Übergeordnete Migrationsschritte

1. Erstellen Sie eine neue Umgebung, die auf einem AL2 023-Plattformzweig basiert.
2. Stellen Sie Ihre Anwendung in der AL2 023-Zielumgebung bereit.

   Ihre vorhandene Produktionsumgebung bleibt aktiv und unberührt, während Sie Tests durchführen und Anpassungen an der neuen Umgebung vornehmen.
3. Testen Sie die Anwendung gründlich in der neuen Umgebung.
4. Wenn Ihre AL2 023-Zielumgebung bereit für die Produktion ist, tauschen Sie die beiden Umgebungen CNAMEs aus, um den Datenverkehr in die neue AL2 023-Umgebung umzuleiten.

Detailliertere Migrationsschritte und bewährte Methoden

Ein detaillierteres blue/green Bereitstellungsverfahren finden Sie unter. Blau/Grün-Bereitstellungen mit Elastic Beanstalk

Spezifischere Anleitungen und detaillierte bewährte Schritte finden Sie unter Blau/Grün-Methode.

Weitere Referenzen zur Planung der Migration

Die folgenden Referenzen können zusätzliche Informationen zur Planung der Migration bieten.

- Von Elastic Beanstalk unterstützte Plattformen in AWS Elastic Beanstalk -Plattformen
- Bisherige eingestellte Plattformzweige
- the section called “Linux-Plattformen”
- Häufig gestellte Fragen zur Einstellung von Plattformen

Migration von Amazon Linux AMI (AL1) zu AL2 oder AL2 023

Wenn Ihre Elastic-Beanstalk-Anwendung auf einer Amazon-Linux-AMI-Plattform basiert, erfahren Sie in diesem Abschnitt, wie Sie die Umgebungen der Anwendung zu Amazon Linux 2 oder Amazon

Linux 2023 migrieren. Plattformzweige der vorherigen Generation, die auf Amazon Linux AMI basieren, wurden inzwischen eingestellt.

Es wird dringend empfohlen, zu Amazon Linux 2023 zu migrieren, da es aktueller ist als Amazon Linux 2. Der Support für das Betriebssystem Amazon Linux 2 wird eher eingestellt als der Support für Amazon Linux 2023. Dementsprechend profitieren Sie bei einer Migration zu Amazon Linux 2023 von einem längeren Supportzeitraum.

Zu beachten ist, dass zwischen den Plattformen Elastic Beanstalk Amazon Linux 2 und Amazon Linux 2023 ein hohes Maß an Kompatibilität besteht. In einigen Bereichen gibt es jedoch Unterschiede: die Standardoption Version 1 (IMDSv1) des Instance Metadata Service, Unterstützung für das Instanztool pkg-repo und einige Apache-Konfigurationen. HTTPd Weitere Informationen finden Sie unter Amazon Linux 2023.

Unterschiede und Kompatibilität

Es kann nicht garantiert werden, dass die auf AL2 023/ AL2 basierenden Plattformzweige abwärtskompatibel mit Ihrer bestehenden Anwendung sind. Ebenfalls zu beachten: Selbst wenn der Anwendungscode in der neuen Plattformversion erfolgreich bereitgestellt wird, kann er sich aufgrund von Betriebssystem- und Laufzeitunterschieden möglicherweise anders verhalten.

Obwohl Amazon Linux AMI und AL2 023/ AL2 denselben Linux-Kernel verwenden, unterscheiden sie sich in den folgenden Aspekten: ihrem Initialisierungssystem, den `libc` Versionen, der Compiler-Toolchain und verschiedenen Paketen. Weitere Informationen finden Sie unter Amazon Linux FAQs 2.

Beim Elastic-Beanstalk-Service wurden auch plattformspezifische Versionen von Laufzeit, Build-Tools und anderen Abhängigkeiten aktualisiert.

Daher empfehlen wir Ihnen, sich Zeit zu nehmen, Ihre Anwendung gründlich in einer Entwicklungsumgebung zu testen und notwendige Anpassungen vorzunehmen.

Allgemeiner Migrationsprozess

Wenn Sie bereit sind, mit der Produktion zu beginnen, benötigt Elastic Beanstalk ein blue/green Deployment, um das Upgrade durchzuführen. Im Folgenden finden Sie die allgemeinen Best-Practice-Schritte, die wir für die Migration mit einem blue/green Bereitstellungsverfahren empfehlen.

Vorbereitung der Tests für die Migration

Bevor Sie die Anwendung bereitstellen und mit dem Testen beginnen, sollten Sie sich die Informationen im Abschnitt Überlegungen für alle Linux-Plattformen weiter unten in diesem Thema durchlesen. Lesen Sie außerdem die für Ihre Plattform geltenden Informationen im folgenden Abschnitt Plattformspezifische Überlegungen. Notieren Sie sich die spezifischen Informationen, die (möglicherweise) für Ihre Anwendung und Konfiguration gelten.

Übergeordnete Migrationsschritte

1. Erstellen Sie eine neue Umgebung, die auf einem AL2 oder AL2 023-Plattformzweig basiert. Wir empfehlen Ihnen, auf einen AL2 023-Plattformzweig zu migrieren.
2. Stellen Sie Ihre Anwendung in der AL2 023/-Zielumgebung AL2 bereit.

   Ihre vorhandene Produktionsumgebung bleibt aktiv und unberührt, während Sie Tests durchführen und Anpassungen an der neuen Umgebung vornehmen.
3. Testen Sie die Anwendung gründlich in der neuen Umgebung.
4. Wenn Ihre AL2 AL2 023/-Zielumgebung für die Produktion bereit ist, tauschen Sie die beiden Umgebungen CNAMEs aus, um den Datenverkehr in die neue Umgebung umzuleiten.

Detailliertere Migrationsschritte und bewährte Methoden

Ein detaillierteres blue/green Bereitstellungsverfahren finden Sie unter. Blau/Grün-Bereitstellungen mit Elastic Beanstalk

Spezifischere Anleitungen und detaillierte bewährte Schritte finden Sie unter Blau/Grün-Methode.

Weitere Referenzen zur Planung der Migration

Die folgenden Referenzen können zusätzliche Informationen zur Planung der Migration bieten.

- Vergleich von Amazon Linux 2 und Amazon Linux 2023 im Benutzerhandbuch von Amazon Linux 2023.
- Was ist Amazon Linux 2023? im Benutzerhandbuch von Amazon Linux 2023
- Von Elastic Beanstalk unterstützte Plattformen in AWS Elastic Beanstalk -Plattformen
- Bisherige eingestellte Plattformzweige
- the section called “Linux-Plattformen”
- Häufig gestellte Fragen zur Einstellung von Plattformen

Überlegungen für alle Linux-Plattformen

In der folgenden Tabelle werden Überlegungen erläutert, die Sie bei der Planung einer Anwendungsmigration auf AL2 AL2 023/ berücksichtigen sollten. Diese Überlegungen gelten für alle Elastic-Beanstalk-Linux-Plattformen, unabhängig von bestimmten Programmiersprachen oder Anwendungsservern.

Area	Änderungen und Informationen
Konfigurationsdateien	Auf AL2 023/ AL2 -Plattformen können Sie Konfigurationsdateien wie bisher verwenden, und alle Abschnitte funktionieren auf die gleiche Weise. Bestimmte Einstellungen funktionieren jedoch möglicherweise nicht so wie bei früheren Amazon-Linux-AMI-Plattformen. Beispiel: • Einige Softwarepakete, die Sie mithilfe einer Konfigurationsdatei installieren, sind unter AL2 023/ möglicherweise nicht verfügbarAL2, oder ihre Namen haben sich möglicherweise geändert. • Einige plattformspezifische Konfigurationsoptionen wurden von ihren plattform spezifischen Namespaces in andere plattformunabhängige Namespaces verschoben. • Proxy-Konfigurationsdateien, die im Verzeichnis `.ebextensions/nginx` bereitgestellt werden, sollten in das Plattform-Hooks-Verzeichnis `.platform/nginx` verschoben werden. Details hierzu finden Sie unter Reverse-Proxy-Konfiguration. Wir raten zur Verwendung von Plattform-Hooks für die Ausführung von benutzerdefiniertem Code auf Ihren Umgebungs-Instances. Sie können in `.ebextensions` -Konfigurationsdateien weiterhin Befehle und Containerbefehle verwenden, aber sie sind nicht mehr so einfach in der Verwendung. Beispielsweise kann das Schreiben von Befehlsskripten in einer YAML-Datei umständlich und schwer zu testen sein. Sie müssen weiterhin `.ebextensions` -Konfigurationsdateien für jedes Skript verwenden, das einen Verweis auf eine AWS CloudFormation -Ressource benötigt.
Plattform-Hooks	AL2 Plattformen haben eine neue Möglichkeit eingeführt, die Plattform Ihrer Umgebung zu erweitern, indem den Hook-Verzeichnissen auf den Instanzen der Umgebung ausführbare Dateien hinzugefügt werden. Bei früheren Linux-Pla

Area	Änderungen und Informationen
	ttformversionen haben Sie möglicherweise benutzerdefinierte Plattform-Hooks verwendet. Diese Hooks wurden nicht für verwaltete Plattformen entwickelt und wurden nicht unterstützt, konnten aber in einigen Fällen auf nützliche Weise funktionieren. Bei AL2 Plattformversionen AL2 023/ funktionieren benutzerdefinierte Plattform-Hooks nicht. Sie sollten alle Hooks auf die neuen Plattform-Hooks migrieren. Weitere Informationen finden Sie unter Plattform-Hooks.

Area	Änderungen und Informationen
Unterstützte Proxy-Server	AL2023/ AL2 Plattformversionen unterstützen dieselben Reverse-Proxy-Server wie jede Plattform, die in ihren Amazon Linux AMI-Plattformversionen unterstützt wird. Alle AL2 023/AL2; -Plattformversionen verwenden Nginx als Standard-Reverse-Proxy-Server, mit Ausnahme der ECS- und Docker-Plattformen. Die Plattformen Tomcat, Node.js, PHP und Python unterstützen ebenfalls Apache HTTPD als Alternative. Alle Plattformen ermöglichen eine einheitliche Konfiguration des Proxy-Servers wie in diesem Abschnitt beschrieben. Die Konfiguration des Proxy-Servers ist jedoch ein wenig anders als auf dem Amazon-Linux-AMI. Dies sind die Unterschiede für alle Plattformen: • Standard ist nginx — Der Standard-Proxyserver auf allen AL2 023/ AL2 Plattformversionen ist nginx. Bei Amazon-Linux-AMI-Plattformversionen von Tomcat, PHP und Python war der Standard-Proxy-Server Apache HTTPD. • Konsistenter Namespace — Alle AL2 023/ AL2 Plattformversionen verwenden den Namespace zur Konfiguration des Proxyservers. `aws:elasticbeanstalk:environment:proxy` Bei Amazon-Linux-AMI-Plattformversionen war dies eine Entscheidung pro Plattform und Node.js verwendete einen anderen Namespace. • Speicherort der Konfigurationsdatei — Sie sollten die Proxykonfigurationsdateien in den `.platform/httpd` Verzeichnissen `.platform/nginx` und auf allen AL2 023/ Plattformversionen platzieren. AL2 Auf Amazon-Linux-AMI-Plattformversionen waren diese Speicherorte `.ebextensions/nginx` bzw. `.ebextensions/httpd` . Informationen zu plattformspezifischen Proxy-Konfigurationsänderungen finden Sie unter the section called “Plattformspezifische Überlegungen”. Informationen zur Proxykonfiguration auf AL2 023/ AL2 -Plattformen finden Sie unter. Reverse-Proxy-Konfiguration

Area	Änderungen und Informationen
Proxy-Konfigurationsänderungen	Einige Änderungen an der Proxy-Konfiguration gelten einheitlich für alle Plattformen, zusätzlich zu den plattformspezifischen Änderungen der Proxy-Konfiguration. Es ist wichtig, sich auf beide zu beziehen, damit Ihre Umgebungen genau konfiguriert werden können. • Alle Plattformen — siehe Reverse-Proxy-Konfiguration • Plattformspezifisch— siehe the section called “Plattformspezifische Überlegungen”.
Instance-Profil	AL2023/ AL2 Plattformen erfordern die Konfiguration eines Instanzprofils. Die Umgebung wird ohne Profil möglicherweise zunächst erfolgreich erstellt, kurz nach der Erstellung können in der Umgebung aber Fehler auftreten, wenn Aktionen, die ein Instance-Profil erfordern, fehlschlagen. Details hierzu finden Sie unter the section called “Instance-Profile”.
Erweiterter Zustand	AL2023/ AL2 Plattformversionen ermöglichen standardmäßig eine erweiterte Systemintegrität. Dies ist eine Änderung, wenn Sie nicht die Elastic-Beanstalk-Konsole zum Erstellen von Umgebungen verwenden. Die Konsole ermöglicht standardmäßig möglichst eine erweiterte Integrität, unabhängig von der Plattformversion. Details hierzu finden Sie unter the section called “Erweiterte Zustandsberichte und -überwachung”.
Benutzerdefiniertes AMI	Wenn Ihre Umgebung ein benutzerdefiniertes AMI verwendet, erstellen Sie mithilfe einer Elastic Beanstalk AL2 AL2 023/ -Plattform ein neues AMI auf Basis von 023/ AL2 für Ihre neue Umgebung. AL2
Benutzerdefinierte Plattformen	Die Managed AMIs of AL2 023/ AL2 Plattformversionen unterstützen keine benutzerdefinierten Plattformen.

Plattformspezifische Überlegungen

In diesem Abschnitt werden Migrationsüberlegungen behandelt, die für bestimmte Elastic-Beanstalk-Linux-Plattformen spezifisch sind.

Docker

Die Docker-Plattform-Branch-Familie, die auf Amazon Linux AMI (AL1) basiert, umfasst drei Plattformzweige. Für jede empfiehlt sich ein anderer Migrationspfad.

AL1 Zweig der Plattform	Migrationspfad zu AL2 023/ AL2
Von Amazon ECS verwaltet er Multi-Container-Docker, der auf Amazon Linux AMI ausgeführt wird () AL1	ECS-basierte Docker AL2 0.23/ Plattformzweige AL2 Die ECS-basierten Docker AL2 023/ AL2 Plattformzweige bieten einen unkompliz ierten Migrationspfad für Umgebungen, die auf der Docker-Plattform mit mehreren Containern ausgeführt werden. AL1 • Wie der vorherige AL1Multi-Container-Docker-Zweig verwenden auch die AL2 023/ AL2 platform Branches Amazon ECS, um die Bereitstellung mehrerer Docker-Container in einem Amazon ECS-Cluster in einer Elastic Beanstalk Beanstalk-Umgebung zu koordinieren. • Die AL2 Plattform-Branches AL2 023/ unterstützen alle Funktionen des vorherige n Multi-Container-Docker-Zweigs. AL1 • Die AL2 Plattformzweige AL2 023/ unterstützen auch dieselbe v2-Datei. `Dockerrun.aws.json` Weitere Informationen zur Migration Ihrer Anwendungen, die auf dem Multicontainer Docker Amazon Linux-Plattformzweig ausgeführt werden, zu einem Amazon ECS, das auf AL2 023/ AL2 platform branch ausgeführt wird, finden Sie unter. ???
Docker läuft auf Amazon Linux AMI () AL1 Vorkonfig urierter Docker (Glassfis	Docker läuft auf der Zweigstelle 023/ AL2 AL2 Wir empfehlen Ihnen, Ihre Anwendungen, die auf Umgebungen laufen, die auf vorkonfiguriertem Docker (Glassfish 5.0) oder Docker auf Amazon Linux AMI (AL1) basieren, in Umgebungen zu migrieren, die auf den Plattformzweigen Docker Running on Amazon Linux 2 oder Docker Running on 023 basieren. AL2 Wenn Ihre Umgebung auf dem Plattformzweig mit vorkonfiguriertem Docker (Glassfish 5.0) basiert, siehe the section called “Tutorial — GlassFish auf Docker: Pfad zu Amazon Linux 2023”.

<table>
<tr><td>AL1 Zweig der Plattform</td><td>Migrationspfad zu AL2 023/ AL2</td></tr>
<tr><td>h 5.0) mit Amazon Linux AMI () AL1</td><td>In der folgenden Tabelle sind Migrationsinformationen aufgeführt, die für den Plattformzweig Docker Running on 023/ spezifisch sind. AL2 AL2
<table>
<tr><th>Area</th><th>Änderungen und Informationen</th></tr>
<tr><td>Speicher</td><td>Elastic Beanstalk konfiguriert Docker so, dass Speichertreiber zum Speichern von Docker-Images und Containerdaten verwendet werden. Auf Amazon-Linux-AMI verwendete Elastic Beanstalk den Device Mapper storage driver (Device Mapper-Speichertreiber). Um die Leistung zu verbessern, stellte Elastic Beanstalk ein zusätzlic hes Amazon-EBS-Volume bereit. Auf Versionen der AL2 Docker-Ve rsion AL2 023/ verwendet Elastic Beanstalk den OverlayFS-Speicher treiber und erzielt so eine noch bessere Leistung, ohne dass ein separates Volume mehr benötigt wird.

Wenn Sie mit Amazon-Linux-AMI die Option <code>BlockDevi ceMappings</code> des Namespace <code>aws:autoscaling:la unchconfiguration</code> verwendet haben, um benutzerdefinierte Speicher-Volume zu einer Docker-Umgebung hinzuzufügen, haben wir Ihnen empfohlen, auch das Amazon-EBS-Volume <code>/dev/xvdc z</code> hinzuzufügen, das durch Elastic Beanstalk bereitgestellt wurde. Elastic Beanstalk stellt dieses Volume nicht mehr bereit, daher sollten Sie es aus Ihren Konfigurationsdateien entfernen. Details hierzu finden Sie unter the section called “Docker-Konfiguration auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)”.</td></tr>
<tr><td>Private Repositor y-Authent ifizierung</td><td>Wenn Sie eine von Docker generierte Authentifizierungsdatei bereitstellen, um eine Verbindung zu einem privaten Repositor y herzustellen, müssen Sie sie nicht mehr in das ältere Format konvertieren, das für die Versionen der Amazon Linux AMI-Docker-Plattform erforderlich war. AL2023/ Die Versionen der AL2 Docker-Pl attform unterstützen das neue Format. Details hierzu finden Sie unter the section called “Authentifizierung mit Image-Repositorys”.</td></tr>
</table></td></tr>
</table>

AL1 Zweig der Plattform	Migrationspfad zu AL2 023/ AL2

Area	Änderungen und Informationen
Proxy-Server	AL2023/ Versionen der AL2 Docker-Plattform unterstützen keine eigenständigen Container, die nicht hinter einem Proxyserver laufen. Bei Amazon-Linux-AMI-Docker-Plattformversionen war dies früher durch den Wert none der Option `ProxyServer` im Namespace `aws:elasticbeanstalk:environment:proxy` möglich.

Go

In der folgenden Tabelle sind Migrationsinformationen für die AL2 023/ AL2 Plattformversionen auf der Go-Plattform aufgeführt.

Area	Änderungen und Informationen
Übergeben eines Portwertes	Auf AL2 023/ AL2 -Plattformen übergibt Elastic Beanstalk keinen Portwert über die Umgebungsvariable an Ihren Anwendungsprozess. `PORT` Sie können dieses Verhalten für Ihren Prozess simulieren, indem Sie eine `PORT`-Umgebungseigensch aft selbst konfigurieren. Wenn Sie jedoch mehrere Prozesse haben und darauf zählen, dass Elastic Beanstalk inkrementelle Portwerte an Ihre Prozesse übergibt (5000, 5100, 5200 usw.), sollten Sie Ihre Implementierung ändern. Weitere Informati onen finden Sie unter Reverse-Proxy-Konfiguration.

Amazon Corretto

In der folgenden Tabelle werden Migrationsinformationen für die Corretto-Plattformzweige in der Java SE-Plattform aufgelistet.

Area	Änderungen und Informationen
Corretto gegen OpenJDK	Zur Implementierung der Java Platform, Standard Edition (Java SE), AL2 023/ AL2 Platform Branches verwenden Amazon Corretto, eine AWS Distribution des Open Java Development Kit (OpenJDK). Frühere Elastic-Beanstalk-Java-SE-Plattformz weige verwenden die in Amazon-Linux-AMI enthaltenen OpenJDK-Pakete.
Build-Tools	AL2023/ AL2 Plattformen verfügen über neuere Versionen der Build-Tools:,, und. `gradle` `maven` `ant`
JAR-Datei behandlun g	Wenn Ihr Quellpaket (ZIP-Datei) auf AL2 023/ AL2 -Plattformen eine einzige JAR-Datei und keine anderen Dateien enthält, benennt Elastic Beanstalk die JAR-Datei nicht mehr in um. `application.jar` Die Umbenennung erfolgt nur, wenn Sie eine JAR-Datei alleine und nicht innerhalb einer ZIP-Datei übermitteln.
Übergeben eines Portwertes	Auf AL2 023/ AL2 -Plattformen übergibt Elastic Beanstalk keinen Portwert über die Umgebungsvariable an Ihren Anwendungsprozess. `PORT` Sie können dieses Verhalten für Ihren Prozess simulieren, indem Sie eine `PORT`-Umgebungseigensch aft selbst konfigurieren. Wenn Sie jedoch mehrere Prozesse haben und darauf zählen, dass Elastic Beanstalk inkrementelle Portwerte an Ihre Prozesse übergibt (5000, 5100, 5200 usw.), sollten Sie Ihre Implementierung ändern. Weitere Informati onen finden Sie unter Reverse-Proxy-Konfiguration.
Java 7	Elastic Beanstalk unterstützt keinen AL2 AL2 023/Java 7-Plattformzweig. Wenn Sie eine Java 7-Anwendung haben, migrieren Sie sie auf Corretto 8 oder Corretto 11.

Tomcat

In der folgenden Tabelle sind Migrationsinformationen für die AL2 023/ AL2 Plattformversionen auf der Tomcat-Plattform aufgeführt.

Area	Änderungen und Informationen
Konfigura tionsopti onen	Auf AL2 Plattformversionen AL2 023/ unterstützt Elastic Beanstalk nur eine Teilmenge der Konfigurationsoptionen und Optionswerte im Namespace. `aws:elasticbeanstalk:environment:proxy` Im Folgenden finden Sie Informationen zur Migration für die einzelnen Optionen.

<table>
<tr><th>Area</th><th>Änderungen und Informationen</th></tr>
<tr><td></td><td>
<table>
<tr><th>Option</th><th>Informationen zur Migration</th></tr>
<tr><td><code>GzipCompression</code></td><td>Wird auf Plattformversionen 023/ nicht unterstützt. AL2 AL2</td></tr>
<tr><td><code>ProxyServer</code></td><td>AL2023/ Die AL2 Tomcat-Plattformversionen unterstützen sowohl die Nginx- als auch die Apache HTTPD-Proxyserver der Version 2.4. Apache Version 2.2 wird jedoch nicht unterstützt.

Bei Amazon-Linux-AMI-Plattformversionen war der Standard-Proxy Apache 2.4. Wenn Sie die Standard-Proxyeinstellung verwendet und benutzerdefinierte Proxy-Konfigurationsdateien hinzugefügt haben, sollte Ihre Proxykonfiguration auch auf 023/ funktionieren. AL2 AL2 Wenn Sie jedoch den Optionswert <code>apache/2.2</code> verwendet haben, müssen Sie nun Ihre Proxy-Konfiguration auf Apache Version 2.4 migrieren.</td></tr>
</table>

Die <code>XX:MaxPermSize</code> Option im <code>aws:elasticbeanstalk:container:tomcat:jvmoptions</code> Namespace wird auf Plattformversionen AL2 AL2 023/ nicht unterstützt. Die JVM-Einstellung zur Änderung der Größe der permanenten Generierung gilt nur für Java 7 und frühere Versionen und gilt daher nicht für 023/ Plattformversionen. AL2 AL2
</td></tr>
<tr><td>Anwendungspfad</td><td>Auf AL2 023/ AL2 -Plattformen lautet der Pfad zum Verzeichnis der Anwendung auf EC2 Amazon-Instances Ihrer Umgebung. <code>/var/app/current</code> Auf Amazon-Linux-AMI-Plattformen lautete er <code>/var/lib/tomcat8/webapps</code> .</td></tr>
</table>

Node.js

In der folgenden Tabelle sind Migrationsinformationen für die AL2 023/ AL2 Plattformversionen auf der Plattform Node.js aufgeführt.

<table>
<tr><th>Area</th><th>Änderungen und Informationen</th></tr>
<tr><td>Installierte Node.js-Versionen</td><td>Auf AL2 023/ AL2 -Plattformen verwaltet Elastic Beanstalk mehrere Plattform zweige von Node.js und installiert nur die neueste Version der Hauptversion von Node.js, die dem Plattformzweig auf jeder Plattformversion entspricht. Beispiels weise wird für jede Plattformversion im Node.js 12-Plattformzweig standardmäßig nur Node.js 12.x.y installiert. Im Fall von Amazon-Linux-AMI-Plattformversionen sind auf jeder Plattformversion die verschiedenen Versionen der verschiedenen Node.js-Versionen installiert und es wird nur ein einziger Plattformzweig gewartet.

Wählen Sie den Node.js-Plattformzweig für die Node.js-Hauptversion aus, die Ihre Anwendung benötigt.</td></tr>
<tr><td>Apache HTTPD-Pro tokolldat einamen</td><td>Wenn Sie auf AL2 023/ AL2 -Plattformen den Apache HTTPD-Proxyserver verwenden, lauten die Namen der HTTPD-Protokolldateien <code>access_log</code> und, was mit allen anderen Plattformen übereinstimmt<code>error_log</code> , die Apache HTTPD unterstützen. Auf Amazon-Linux-AMI-Plattformversionen hießen diese Protokoll dateien <code>access.log</code> bzw. <code>error.log</code> .

Weitere Details zu Namen und Speicherorten von Protokolldateien für alle Plattform en finden Sie unter the section called “So richtet Elastic Beanstalk Logs ein CloudWatch”.</td></tr>
<tr><td>Konfigura tionsopti onen</td><td>Auf AL2 023/ AL2 -Plattformen unterstützt Elastic Beanstalk die Konfigurationsopti onen im Namespace nicht. <code>aws:elasticbeanstalk:container:nodejs</code> Für einige Optionen gibt es Alternativen. Im Folgenden finden Sie Informationen zur Migration für die einzelnen Optionen.

<table>
<tr><th>Option</th><th>Informationen zur Migration</th></tr>
<tr><td><code>NodeComm</code> nd</td><td>Verwenden Sie ein <code>Procfile</code> oder das Schlüsselwort <code>scripts</code> in einer <code>package.json</code> -Datei, um das Startskript anzugeben.</td></tr>
<tr><td><code>NodeVers</code> on</td><td>Verwenden Sie in einer <code>package.json</code> -Datei das Schlüsselwort <code>engines</code>, um die Node.js-Version anzugeben. Beachten Sie, dass Sie jeweils nur eine Node.js-Version angeben können, die Ihrem Plattformzweig entspricht. Wenn Sie beispielsweise den Plattform zweig Node.js 12 verwenden, können Sie nur eine 12.x.y Node.js-</td></tr>
</table></td></tr>
</table>

Area	Änderungen und Informationen

Option	Informationen zur Migration
	Version angeben. Details hierzu finden Sie unter the section called “Angeben von Node.js-Abhängigkeiten mit einer package.json-Datei”.
`GzipCompression`	Wird auf Plattformversionen 023/ nicht unterstützt. AL2 AL2

`ProxyServer`	Bei den Plattformversionen AL2 023/ AL2 Node.js wurde diese Option in den Namespace verschoben. `aws:elasticbeanstalk:environment:proxy` Sie können zwischen `nginx` (Standardeinstellung) und `apache` wählen.
	
	AL2Die Plattformversionen 023/ AL2 Node.js unterstützen keine eigenständigen Anwendungen, die nicht hinter einem Proxyserver laufen. Bei Amazon-Linux-AMI-Node.js-Plattformversionen war dies früher durch den Wert none der Option `ProxyServer` im Namespace `aws:elasticbeanstalk:container:nodejs` möglich. Wenn in Ihrer Umgebung eine eigenständige Anwendung ausgeführt wird, aktualisieren Sie den Code so, dass dem Port zugehört wird, an den der Proxy-Server (nginx oder Apache) den Datenverkehr weiterleitet.
	
	`var port = process.env.PORT || 5000;`
	
	`app.listen(port, function() {`
	`  console.log('Server running at http://127.0.0.1:%s',`
	` port);`
	`});`

PHP

In der folgenden Tabelle sind Migrationsinformationen für die AL2 023/ AL2 Plattformversionen auf der PHP-Plattform aufgeführt.

Area	Änderungen und Informationen
PHP-Dateiverarbeitung	Auf AL2 023/ AL2 -Plattformen werden PHP-Dateien mit PHP-FPM (einem CGI-Prozessmanager) verarbeitet. Auf Amazon-Linux-AMI-Plattformen wird mod_php (ein Apache-Modul) verwendet.
Proxy-Server	AL2023/ AL2 PHP-Plattformversionen unterstützen sowohl den Nginx- als auch den Apache HTTPD-Proxyserver. Standardmäßig wird nginx verwendet. Amazon-Linux-AMI-PHP-Plattformversionen unterstützten nur Apache HTTPD. Wenn Sie benutzerdefinierte Apache-Konfigurationsdateien hinzugefügt haben, können Sie die Option `ProxyServer` im Namespace `aws:elasticbeanstalk:environment:proxy` auf `apache` festlegen.

Python

In der folgenden Tabelle sind Migrationsinformationen für die AL2 023/ AL2 Plattformversionen auf der Python-Plattform aufgeführt.

Area	Änderungen und Informationen
WSGI-Server	Auf AL2 023/ AL2 -Plattformen ist Gunicorn der Standard-WSGI-Server. Standardmäßig überwacht Gunicorn Port 8000. Der Port kann sich von dem unterscheiden, den Ihre Anwendung auf der Amazon-Linux-AMI-Plattform verwendet hat. Wenn Sie die `WSGIPath`-Option des `aws:elasticbeanstalk:container:python` -Namespace festlegen, ersetzen Sie den Wert durch die Syntax von Gunicorn. Details hierzu finden Sie unter the section called “Namespaces der Python-Konfiguration”. Alternativ können Sie einen `Procfile` verwenden, um den WSGI-Server anzugeben und zu konfigurieren. Details hierzu finden Sie unter the section called “Procfile”.
Anwendungspfad	Auf AL2 023/ AL2 -Plattformen lautet der Pfad zum Verzeichnis der Anwendung auf EC2 Amazon-Instances Ihrer Umgebung. `/var/app/current` Auf Amazon-Linux-AMI-Plattformen lautete er `/opt/python/current/app` .

Area	Änderungen und Informationen
Proxy-Server	AL2023/ Die Versionen der AL2 Python-Plattform unterstützen sowohl den Nginx- als auch den Apache HTTPD-Proxyserver. Standardmäßig wird nginx verwendet. Amazon-Linux-AMI-Python-Plattformversionen unterstützten nur Apache HTTPD. Wenn Sie benutzerdefinierte Apache-Konfigurationsdateien hinzugefügt haben, können Sie die Option `ProxyServer` im Namespace `aws:elasticbeanstalk:environment:proxy` auf `apache` festlegen.

Ruby

In der folgenden Tabelle sind Migrationsinformationen für die AL2 023/ Plattformversionen auf der Ruby-Plattform AL2 aufgeführt.

Area	Änderungen und Informationen
Installierte Ruby-Versionen	Auf AL2 023/ AL2 -Plattformen installiert Elastic Beanstalk nur die neueste Version einer einzigen Ruby-Version, die dem Plattformzweig entspricht, auf jeder Plattformversion. Beispielsweise ist auf den einzelnen Plattformversionen im Ruby 2.6-Plattformzweig nur Ruby 2.6.x installiert. Auf Amazon-Linux-AMI-Plattformversionen sind die neuesten Versionen mehrerer Ruby-Versionen installiert, z. B. 2.4.x, 2.5.x und 2.6.x. Wenn Ihre Anwendung eine Ruby-Version verwendet, die nicht dem von Ihnen verwendeten Plattformzweig entspricht, sollten Sie zu einem Plattformzweig wechseln, auf dem die richtige Ruby-Version für Ihre Anwendung installiert ist.
Anwendungsserver	Auf AL2 023/ AL2 -Plattformen installiert Elastic Beanstalk nur den Puma-Anwendungsserver auf allen Ruby-Plattformversionen. Sie können eine `Procfile`-Datei verwenden, um einen anderen Anwendungsserver zu starten, und eine, `Gemfile`-Datei, um ihn zu installieren. Auf der Amazon-Linux-AMI-Plattform werden zwei Varianten von Plattformzweigen für jede Ruby-Version unterstützt – Variante mit dem Puma-Anwendungsserver und eine Variante mit dem Passenger-Anwendungsserver. Wenn Ihre Anwendung Passenger verwendet, können Sie Ihre Ruby-Umgebung für die Installation und Verwendung von Passenger konfigurieren.

Area	Änderungen und Informationen
	Weitere Informationen und Beispiele finden Sie unter the section called “Die Ruby-Plattform”.

Häufig gestellte Fragen zur Einstellung von Plattformen

> **Note**
>
> Elastic Beanstalk hat am 18. Juli 2022 alle Plattformbranchen, die auf Amazon Linux AMI (AL1) basieren, eingestellt.

Diese FAQ enthalten Antworten auf folgende Themen:

- Support-Richtlinie für die Elastic Beanstalk-Plattform
- Bisherige eingestellte Plattformzweige
- Von Elastic Beanstalk unterstützte Plattformen in AWS Elastic Beanstalk -Plattformen
- Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2
- Amazon Linux FAQs 2.

1. Was bedeutet es, wenn ein Plattformzweig eingestellt wird?

Nach dem angekündigten Einstellungsdatum eines Plattformzweigs können Sie keine neue Umgebung mehr erstellen, die auf dem veralteten Plattformzweig basiert, es sei denn, Sie haben bereits eine aktive Umgebung, die auf diesem Plattformzweig basiert. Weitere Informationen finden Sie unter FAQ #11. Elastic Beanstalk wird für diese Versionszweige der Plattform keine neuen Wartungs-Updates mehr bereitstellen. Von der Verwendung eines veralteten Plattformzweigs in Produktionsumgebungen wird abgeraten. Weitere Informationen finden Sie unter FAQ #5.

2. Warum wurden die Filialen der AL1 basierten Plattformen AWS eingestellt?

Elastic Beanstalk stellt Plattformzweige ein, wenn Plattformkomponenten veraltet sind oder deren Anbieter sie selbst ausmustern. In diesem Fall hat das Amazon Linux AMI (AL1) den Standardsupport zum 31. Dezember 2020 eingestellt. Elastic Beanstalk bot zwar bis 2022 weiterhin AL1 basierte Plattformen an, aber seitdem haben wir auch Plattformen veröffentlicht AL2 , die über die neuesten

Funktionen verfügen. AL2023 Damit Kunden auch in Zukunft von den neuesten Sicherheits- und Funktionen profitieren können, ist es für Kunden von entscheidender Bedeutung, AL2023 auf unsere AL2 oder deren Plattformen zu migrieren.

3. Welche Plattformverzweigungen werden außer Betrieb genommen?

Eine Liste der Plattformkomponenten und Plattformverzweigungen, die außer Betrieb genommen wurden, finden Sie unter Bisherige eingestellte Plattformzweige.

4. Welche Plattformen werden derzeit unterstützt?

Siehe Von Elastic Beanstalk unterstützte Plattformen in AWS Elastic Beanstalk -Plattformen.

5. Wird Elastic Beanstalk Komponenten meiner Umgebung nach der Einstellung entfernen oder beenden?

Unsere Richtlinie für stillgelegte Plattformfilialen verbietet weder den Zugriff auf Umgebungen noch das Löschen von Ressourcen. Eine Umgebung, die auf einem veralten Plattformzweig basiert, kann jedoch in eine unvorhersehbare Situation geraten, da Elastic Beanstalk für eingestellte Plattformzweige keine Sicherheitsaktualisierungen, keinen technischen Support und keine Hotfixes mehr bereitstellen kann, wenn der jeweilige Anbieter seine Komponente als End of Life (EOL) einstuft. Beispielsweise kann eine nachteilige und kritische Sicherheitslücke in einer Umgebung auftreten, die in einem eingestellten Plattformzweig ausgeführt wird. Es könnte auch passieren, dass eine EB-API-Aktion nicht mehr für die Umgebung funktioniert, wenn sie irgendwann nicht mehr mit dem Elastic-Beanstalk-Service kompatibel ist. Die Gefahr hierfür steigt, je länger eine Umgebung, die auf einem eingestellten Plattformzweig basiert, aktiv bleibt.

Falls bei der Ausführung Ihrer Anwendung auf einer veralteten Plattform Probleme auftreten und Sie sie nicht auf eine unterstützte Plattform migrieren können, müssen Sie andere Alternativen in Betracht ziehen. Eine mögliche Umgehung ist das Einkapseln der Anwendung in ein Docker-Image zum Ausführen als Docker-Container. Dies würde es einem Kunden ermöglichen, jede unserer Docker-Lösungen wie unsere Elastic Beanstalk AL2023 AL2 Beanstalk/Docker-Plattformen oder andere Docker-basierte Dienste wie Amazon ECS oder Amazon EKS zu nutzen. Zu den Alternativen, die nicht zu Docker gehören, gehört unser AWS CodeDeploy Service, der eine vollständige Anpassung der von Ihnen gewünschten Laufzeiten ermöglicht.

6. Kann ich beantragen, dass die Einstellung später stattfindet?

Nein. Nach dem Stichtag werden bestehende Umgebungen jedoch weiterhin funktionieren. Elastic Beanstalk wird allerdings keine Wartungs- und Sicherheitsupdates für die Plattform mehr

bereitstellen. Daher ist es wichtig, zu einer basierten Plattform zu migrieren AL2 oder AL2023 ob Sie noch Anwendungen auf einer AL1 basierten Plattform ausführen. Weitere Informationen zu Risiken und Problemumgehungen finden Sie unter FAQ #5.

7. Was sind die Problemumgehungen, wenn ich meine AL2 oder die AL2023 Migration nicht rechtzeitig abschließen kann?

Kunden können die Umgebung zwar weiterhin ausführen, wir empfehlen jedoch dringend, alle Elastic-Beanstalk-Umgebungen auf eine unterstützte Plattformversion zu migrieren. Dadurch wird das Risiko minimiert und Sie können von den wichtigen Verbesserungen bei Sicherheit, Leistung und Funktionen profitieren, die die neueren Versionen mit sich bringen. Weitere Informationen zu Risiken und Problemumgehungen finden Sie unter FAQ #5.

8. Was ist der empfohlene Prozess für die Migration zu unseren Plattformen AL2 ? AL2023

Umfassende Anweisungen AL1 zur AL2023 AL2 /-Migration finden Sie unterMigrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2. In diesem Thema wird erklärt, dass Elastic Beanstalk ein blue/green Deployment benötigt, um das Upgrade durchzuführen.

9. Welche Auswirkungen hat es, wenn ich eine Umgebung auf einer eingestellten Plattform habe?

Eine Umgebung, die auf einem eingestellten Plattformzweig basiert, kann in eine unvorhersehbare Situation geraten, da Elastic Beanstalk für eingestellte Plattformzweige keine Sicherheitsaktualisierungen, keinen technischen Support und keine Hotfixes mehr bereitstellen kann, wenn der jeweilige Anbieter seine Komponente als End of Life (EOL) einstuft. Beispielsweise kann eine nachteilige und kritische Sicherheitslücke in einer Umgebung auftreten, die in einem eingestellten Plattformzweig ausgeführt wird. Es könnte auch passieren, dass eine EB-API-Aktion nicht mehr für die Umgebung funktioniert, wenn sie irgendwann nicht mehr mit dem Elastic-Beanstalk-Service kompatibel ist. Die Gefahr dafür steigt, je länger ein Umfeld auf einem eingestellten Plattformzweig aktiv bleibt. Weitere Informationen finden Sie unter FAQ #5.

10. Was passiert 90 Tage nach dem Stichtag für die Einstellung?

Unsere Richtlinie für stillgelegte Plattform-Branches verhindert weder den Zugriff auf Umgebungen noch das Löschen von Ressourcen. Beachten Sie jedoch, dass eine Umgebung, die auf einem eingestellten Plattformzweig basiert, in eine unvorhersehbare Situation geraten kann, da Elastic Beanstalk für eingestellte Plattformzweige keine Sicherheitsaktualisierungen, keinen technischen

Support und keine Hotfixes mehr bereitstellen kann, wenn der jeweilige Anbieter seine Komponente als End of Life (EOL) einstuft. Beispielsweise kann eine nachteilige und kritische Sicherheitslücke in einer Umgebung auftreten, die in einem eingestellten Plattformzweig ausgeführt wird. Es könnte auch passieren, dass eine EB-API-Aktion nicht mehr für die Umgebung funktioniert, wenn sie irgendwann nicht mehr mit dem Elastic-Beanstalk-Service kompatibel ist. Die Gefahr dafür steigt, je länger ein Umfeld auf einem eingestellten Plattformzweig aktiv bleibt. Weitere Informationen finden Sie unter FAQ #5.

11. Kann ich eine neue Umgebung erstellen, die auf einer eingestellten Plattform basiert?

Sie können eine neue Umgebung basierend auf einem eingestellten Plattformzweig erstellen, wenn Sie diesen Plattformzweig bereits verwendet haben, um eine vorhandene Umgebung mit demselben Konto und in derselben Region zu erstellen. Der stillgelegte Plattformzweig wird in der Elastic Beanstalk Beanstalk-Konsole nicht verfügbar sein. Für Kunden mit Umgebungen, die auf einem eingestellten Plattformzweig basieren, wird er jedoch über EB CLI, EB API und AWS CLI verfügbar sein. Auch Bestandskunden können die Konsolen Clone environment (Umgebung klonen) und Rebuild environment (Umgebung neu erstellen) nutzen. Beachten Sie jedoch, dass eine Umgebung, die auf einem eingestellten Plattformzweig basiert, in eine unvorhersehbare Situation kommen kann. Weitere Informationen finden Sie unter FAQ #5.

12. Wenn ich eine bestehende Umgebung habe, die auf einem stillgelegten Plattform-Branch läuft, bis wann kann ich dann eine neue Umgebung auf der Grundlage des stillgelegten Plattform-Branches erstellen? Kann ich das über die Konsole, die CLI oder die API tun?

Sie können die Umgebung nach dem Datum der Außerbetriebnahme erstellen. Beachten Sie jedoch, dass ein eingestellter Plattformzweig in eine unvorhersehbare Situation kommen kann. Je später eine solche Umgebung geschaffen wird oder aktiv ist, desto höher ist das Risiko, dass in der Umgebung unerwartete Probleme auftreten. Weitere Informationen zum Erstellen einer Entwicklungsumgebung finden Sie unter FAQ #11.

13. Kann ich meine auf einer eingestellten Plattform basierende Umgebung klonen oder neu erstellen?

Ja. Dazu können Sie die Konsolen Clone environment (Umgebung klonen) und Rebuild environment (Umgebung neu erstellen) nutzen. Sie können auch die EB-CLI, die EB-API und verwenden AWS CLI. Weitere Informationen zum Erstellen einer Entwicklungsumgebung finden Sie unter FAQ #11.

Wir empfehlen Ihnen aber dringend, alle Ihre Elastic-Beanstalk-Umgebungen auf eine unterstützte Plattformversion zu migrieren. Dadurch wird das Risiko minimiert und Sie können von den wichtigen Verbesserungen bei Sicherheit, Leistung und Funktionen profitieren, die die neueren Versionen mit sich bringen. Weitere Informationen zu Risiken und Problemumgehungen finden Sie unter FAQ #5.

14. Was würde nach dem Auslaufdatum mit den AWS Ressourcen meiner Elastic Beanstalk Beanstalk-Umgebung passieren, die auf einem stillgelegten Plattformzweig basiert? Wenn beispielsweise die laufende EC2-Instance beendet wird, könnte Elastic Beanstalk dann eine neue AL1-basierte EC2-Instance starten, um die Kapazität aufrechtzuerhalten?

Die Ressourcen der Umgebung würden aktiv bleiben und weiterhin funktionieren. Elastic Beanstalk wird zwar automatisch für AL1-EC2-Instances in der Umgebung skalieren. Aber Elastic Beanstalk wird keine neuen Updates für die Plattformwartung für die Umgebung bereitstellen, was dazu führen kann, dass die Umgebung im Laufe der Zeit in eine unvorhersehbare Situation gelangt. Weitere Informationen finden Sie unter FAQ #5.

15. Was sind die Hauptunterschiede zwischen den Betriebssystemen AL2023/AL2 und Amazon Linux AMI (AL1)? Wie sind die Elastic Beanstalk- und AL2023 AL2 Plattformbranchen betroffen?

Obwohl Amazon Linux AMI und AL2023/AL2 denselben Linux-Kernel verwenden, unterscheiden sie sich in ihrem Initialisierungssystem, ihren `libc` Versionen, der Compiler-Toolchain und verschiedenen Paketen. Weitere Informationen finden Sie unter Amazon Linux FAQs 2.

Beim Elastic-Beanstalk-Service wurden auch plattformspezifische Versionen von Laufzeit, Build-Tools und anderen Abhängigkeiten aktualisiert. Es kann nicht garantiert werden, dass die Plattformzweige auf der AL2 Basis von AL2023/abwärtskompatibel mit Ihrer bestehenden Anwendung sind. Selbst wenn Ihr Anwendungscode auf der neuen Plattformversion erfolgreich bereitgestellt wird, kann er sich aufgrund von Betriebssystem- und Laufzeitunterschieden möglicherweise anders verhalten oder abweichen. Eine Übersicht und Beschreibung der Konfigurationen und Anpassungen, die Sie überprüfen und testen müssen, finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Abbrechen von Umgebungskonfigurations-Updates und Anwendungsbereitstellungen

Sie können angefangene Updates abbrechen, die durch Änderungen der Umgebungskonfiguration ausgelöst wurden. Sie können auch die laufende Bereitstellung einer neuen Anwendungsversion abbrechen. Sie können beispielsweise eine Aktualisierung abbrechen, wenn Sie sich dazu entscheiden, die bestehende Umgebungskonfiguration weiter zu verwenden, anstatt neue Einstellungen für die Umgebungskonfiguration anzuwenden. Oder Sie stellen möglicherweise fest, dass bei Ihrer Anwendungsversion Probleme vorliegen, die dazu führen, dass sie nicht gestartet oder ordnungsgemäß ausgeführt werden. Durch das Abbrechen einer Umgebungsaktualisierung oder Anwendungsversion können verhindern, dass Sie warten müssen, bis das Update oder der Bereitstellungsprozess durchgeführt wurde, bevor Sie erneut versuchen, die Umgebung oder Anwendungsversion zu aktualisieren.

> **Note**
>
> Während der Bereinigungsphase, in der alte Ressourcen entfernt werden, die nicht mehr benötigt werden, können Sie das Update nicht mehr abbrechen, nachdem der letzte Stapel an Instances aktualisiert wurde.

Elastic Beanstalk führt das Rollback genau wie das letzte erfolgreiche Update durch. Wenn Sie beispielsweise zeitbasierte fortlaufende Updates in Ihrer Umgebung aktiviert haben, wartet Elastic Beanstalk die angegebene Pausenzeit zwischen dem Zurücksetzen der Änderungen auf einem Stapel von Instances, bevor die Änderungen auf dem nächsten Stapel zurückgesetzt werden. Wenn Sie vor Kurzem fortlaufende Updates aktiviert haben, die letzte erfolgreiche Aktualisierung Ihrer Einstellungen der Umgebungskonfiguration jedoch ohne fortlaufende Updates durchgeführt wurde, führt Elastic Beanstalk das Rollback auf allen Instances gleichzeitig durch.

Sie können Elastic Beanstalk nicht daran hindern, zur vorherigen Umgebungskonfiguration zurückzukehren, sobald mit dem Abbrechen des Updates begonnen wurde. Der Rollbackvorgang wird fortgesetzt, bis alle Instances der Umgebung wieder die vorherige Umgebungskonfiguration aufweisen oder bis im Rollback Fehler auftreten. Bei Bereitstellungen von Anwendungsversionen wird durch das Abbrechen der Bereitstellung diese nur gestoppt. Einige Instances haben die neue Anwendungsversion und andere führen weiterhin die vorhandene Anwendungsversion aus. Sie können die gleiche oder eine andere Anwendungsversion zu einem späteren Zeitpunkt bereitstellen.

Weitere Informationen zu fortlaufenden Aktualisierungen finden Sie unter Fortlaufende Updates der Elastic Beanstalk-Umgebungskonfiguration. Weitere Informationen zu Bereitstellungen der Anwendungsversion in Stapeln finden Sie unter Bereitstellungsrichtlinien und -einstellungen.

So brechen Sie ein Update ab

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite Actions (Aktionen) und dann Abort current operation (Laufende Operation abbrechen).

Elastic Beanstalk-Umgebungen wiederherstellen

Ihre AWS Elastic Beanstalk Umgebung kann unbrauchbar werden, wenn Sie die Elastic Beanstalk Beanstalk-Funktionalität nicht verwenden, um die der Umgebung zugrunde liegenden Ressourcen zu ändern oder zu beenden. AWS Wenn dies der Fall ist, können Sie die Umgebung erneut erstellen, um zu versuchen, sie wieder in einen betriebsbereiten Zustand zu versetzen. Durch die Neuerstellung einer Umgebung werden alle ihre Ressourcen beendet und durch neue Ressourcen mit derselben Konfiguration ersetzt.

Sie können beendete Umgebungen auch innerhalb von sechs Wochen (42 Tage) nach ihrer Beendigung erneut erstellen. Bei der Neuerstellung versucht Elastic Beanstalk, eine neue Umgebung mit dem gleichen Namen, der gleichen ID und Konfiguration zu erstellen.

Neuerstellung einer laufenden Umgebung

Sie können eine Umgebung über die Elastic Beanstalk-Konsole oder mit Hilfe der `RebuildEnvironment` API neu erstellen.

> **Warning**
>
> Wenn Ihre Umgebung über eine gekoppelte Datenbank verfügt, wird diese im Rahmen der Neuerstellung gelöscht, und die neue Datenbank in der neu erstellten Umgebung enthält die vorherigen Daten nicht. Wenn Sie die Datenbank behalten oder einen Snapshot erstellen möchten, stellen Sie sicher, dass die Richtlinie zum Löschen der Datenbank ordnungsgemäß

konfiguriert ist, damit sie nach dem Neuaufbau die gewünschten Ergebnisse erzielt. Weitere Informationen finden Sie unter Datenbank-Lebenszyklus.

So erstellen Sie eine laufende Umgebung erneut (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Rebuild environment (Umgebung neu erstellen).
4. Wählen Sie Rebuild (Neu erstellen).

Um eine laufende Umgebung mit der Elastic Beanstalk Beanstalk-API neu aufzubauen, verwenden Sie die `RebuildEnvironment`Aktion mit dem AWS CLI oder dem AWS SDK.

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Neuerstellung einer beendeten Umgebung

Sie können eine Umgebung erneut erstellen oder eine beendete Umgebung wiederherstellen, indem Sie die Elastic Beanstalk-Konsole, die EB-CLI oder die `RebuildEnvironment`-API verwenden.

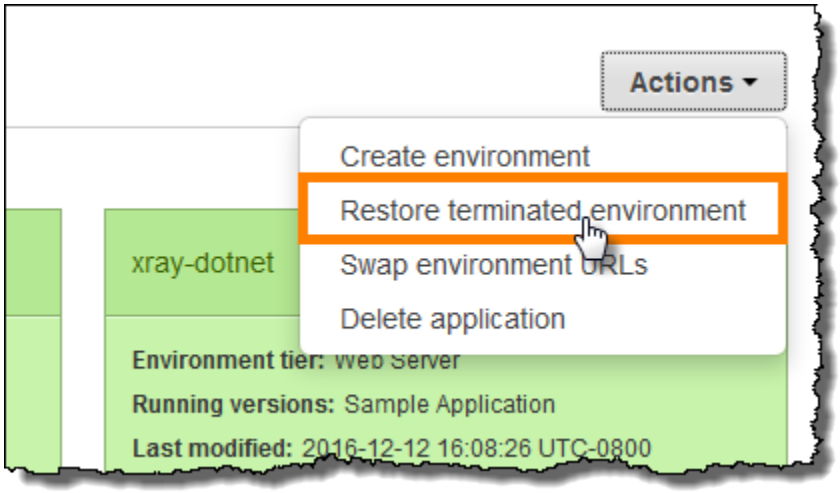
> **Note**
>
> Sofern Sie nicht Ihren eigenen benutzerdefinierten Domänennamen für Ihre beendete Umgebung verwenden, verwendet die Umgebung eine Subdomäne von elasticbeanstalk.com. Diese Unterdomänen werden innerhalb einer Elastic Beanstalk-Region gemeinsam genutzt. Daher können sie von jeder Umgebung verwendet werden, die von einem beliebigen Kunden in derselben Region erstellt wurden. Wenn Ihre Umgebung beendet wurde, kann eine andere Umgebung ihre Subdomäne verwenden. In diesem Fall würde die Neuerstellung fehlschlagen.
> Sie können dieses Problem vermeiden, indem Sie eine benutzerdefinierte Domäne verwenden. Details dazu finden Sie unter Der Domainname Ihrer Elastic Beanstalk-Umgebung.

Kürzlich beendete Umgebungen werden bis zu einer Stunde lang in der Anwendungsübersicht angezeigt. Während dieser Zeit können Sie Ereignisse für die Umgebung in deren Dashboard anzeigen und die Restore environment (Umgebung wiederherstellen)-Action (Aktion) zur Neuerstellung verwenden.

Zur Neuerstellung einer Umgebung, die nicht mehr sichtbar ist, verwenden Sie die Option Restore terminated environment (Beendete Umgebung wiederherstellen) auf der Anwendungsseite.

So erstellen Sie eine beendete Umgebung erneut (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Klicken Sie auf Actions und dann auf Restore terminated environment (Beendete Umgebung wiederherstellen).



4. Wählen Sie eine beendete Umgebung.
5. Wählen Sie Restore (Wiederherstellen) aus.



Bei der Neuerstellung versucht Elastic Beanstalk, eine neue Umgebung mit dem gleichen Namen, der gleichen ID und Konfiguration zu erstellen. Wenn eine Umgebung mit dem gleichen Namen oder der gleichen URL vorhanden ist, wenn Sie versuchen, eine Neuerstellung durchzuführen, schlägt diese fehl. Das Löschen der Anwendungsversion, die für die Umgebung bereitgestellt wurde, verursacht ebenfalls einen Fehler bei der Neuerstellung.

Wenn Sie die EB CLI für die Verwaltung Ihrer Umgebung verwenden, können Sie über den eb restore-Befehl eine beendete Umgebung erneut erstellen.

```
$ eb restore e-vdnftxubwq
```

Weitere Informationen finden Sie unter eb restore.

Um eine beendete Umgebung mit der Elastic Beanstalk Beanstalk-API neu aufzubauen, verwenden Sie die `RebuildEnvironment`Aktion mit dem AWS CLI oder dem AWS SDK.

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Typen von Umgebungen

In AWS Elastic Beanstalk können Sie eine skalierbare Umgebung mit Lastenausgleich oder eine Einzelinstanzumgebung erstellen. Der benötigte Umgebungstyp hängt von der bereitgestellten Anwendung ab. Beispielsweise können Sie eine Anwendung in einer Single-Instance-Umgebung entwickeln und testen, um Kosten zu sparen. Wenn die Anwendung für die Produktion bereit ist, können Sie ein Upgrade auf eine skalierbare Umgebung mit Lastausgleich ausführen.

> **Note**
>
> Die Worker-Umgebungsebene einer Webanwendung, die Hintergrundaufgaben ausführt, verfügt nicht über einen Load Balancer. Allerdings kann in einer Worker-Umgebung effektiv skaliert werden, sodass bei entsprechendem Workload weitere Instances für die Datenverarbeitung der Amazon SQS-Warteschlange zur Auto Scaling-Gruppe hinzufügt werden.

Skalierbare Umgebung mit Lastausgleich

Eine skalierbare Umgebung mit Lastenausgleich verwendet die Services Elastic Load Balancing und Amazon EC2 Auto Scaling, um die EC2 Amazon-Instances bereitzustellen, die für Ihre bereitgestellte Anwendung erforderlich sind. Amazon EC2 Auto Scaling startet automatisch zusätzliche Instances, um der zunehmenden Belastung Ihrer Anwendung Rechnung zu tragen. Wenn die Auslastung Ihrer Anwendung abnimmt, stoppt Amazon EC2 Auto Scaling die Instances, lässt aber immer die angegebene Mindestanzahl von Instances laufen. Wenn die Anwendung Skalierbarkeit mit einer Option für die Ausführung in mehreren Availability Zones erfordert, sollten Sie eine skalierbare Umgebung mit Lastausgleich verwenden. Falls Sie nicht sicher sind, welcher Umgebungstyp der richtige ist, können Sie einfach einen auswählen und diesen später bei Bedarf ändern.

Single-Instance-Umgebung

Eine Single-Instance-Umgebung enthält eine EC2 Amazon-Instance mit einer Elastic IP-Adresse. Eine Single-Instance-Umgebung besitzt keinen Load Balancer. Dies kann helfen, die Kosten im Vergleich zu einer skalierbaren Umgebung mit Lastausgleich zu senken. Obwohl eine Einzelinstanz-Umgebung den Amazon EC2 Auto Scaling-Service verwendet, sind die Einstellungen für die Mindestanzahl von Instances, die maximale Anzahl von Instances und die gewünschte Kapazität alle auf 1 gesetzt. Folglich werden keine neuen Instances gestartet, mit denen die Datenverarbeitung der Anwendung auch bei einer erhöhten Workload gewährleistet werden könnte.

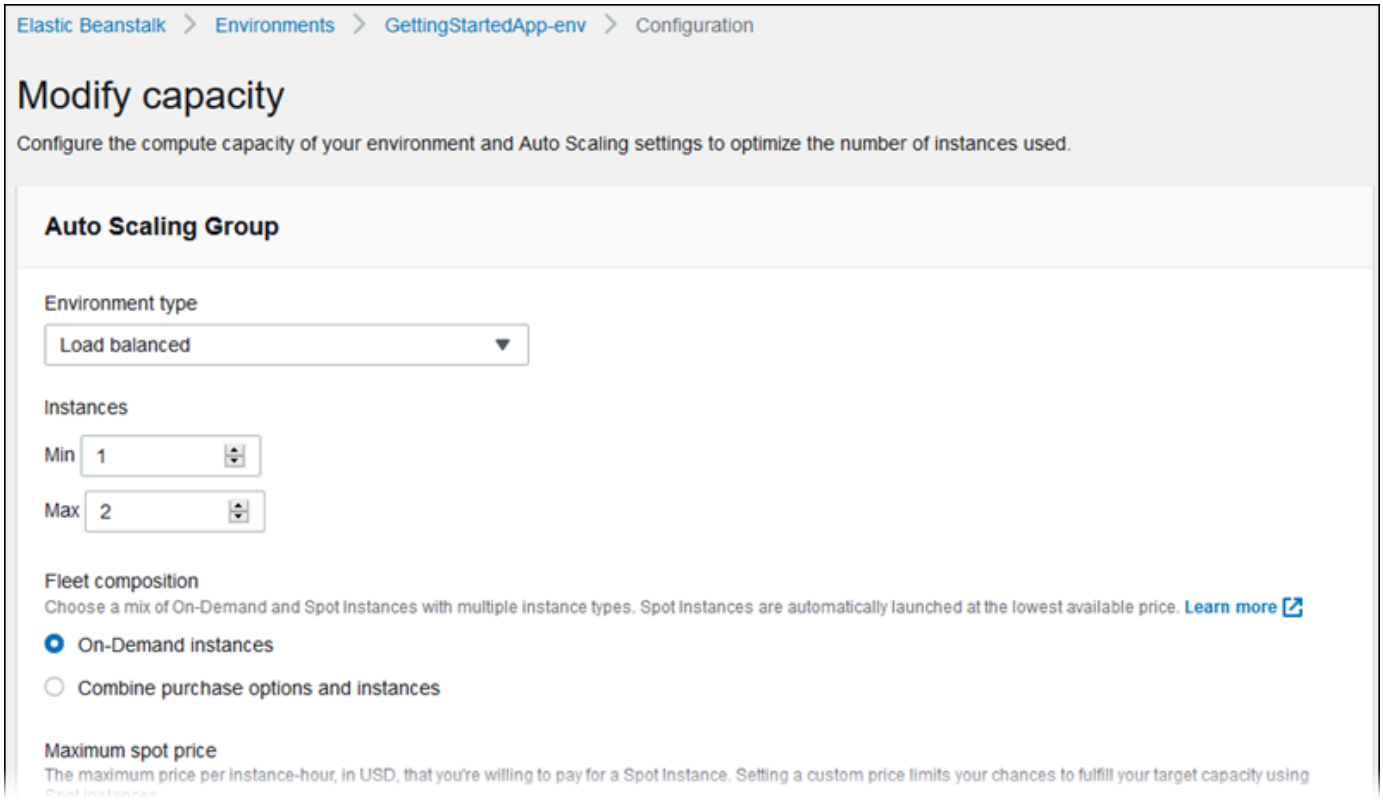
Verwenden Sie eine Umgebung mit einer Instance, wenn Sie davon ausgehen, dass die Produktionsanwendung mit geringem Workload arbeitet, oder wenn Sie eine Remote-Entwicklung ausführen. Falls Sie nicht sicher sind, welcher Umgebungstyp der richtige ist, können Sie einfach einen auswählen und diesen später ändern. Weitere Informationen finden Sie unter Ändern des Umgebungstyps.

Ändern des Umgebungstyps

Sie können den Typ Ihrer Umgebung in eine Single-Instance-Umgebung oder eine skalierbare Umgebung mit Lastausgleich ändern, indem Sie die Konfiguration Ihrer Umgebung bearbeiten. In einigen Fällen ist es möglicherweise erforderlich, von einem Umgebungstyp zum anderen zu wechseln. Das kann beispielsweise der Fall sein, wenn Sie eine Anwendung in einer Umgebung mit einer Instance entwickelt und getestet haben, um Kosten zu sparen. Wenn die Anwendung für die Produktion bereit ist, können Sie den Umgebungstyp in eine skalierbare Umgebung mit Lastausgleich ändern, damit sie entsprechend den Anforderungen Ihrer Kunden skaliert werden kann.

So ändern Sie den Umgebungstyp

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Kategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Wählen Sie aus der Liste Environment Type (Umgebungstyp) den gewünschten Umgebungstyp aus.



6. Wählen Sie Speichern.

   Es kann mehrere Minuten dauern, bis die Umgebung aktualisiert ist, während Elastic Beanstalk Ressourcen bereitstellt AWS .

Wenn sich Ihre Umgebung in einer VPC befindet, wählen Sie Subnetze aus, in denen Elastic Load Balancing und EC2 Amazon-Instances platziert werden sollen. Jede Availability Zone, in der die Anwendung ausgeführt wird, muss über beides verfügen. Details dazu finden Sie unter Verwenden von Elastic Beanstalk mit Amazon VPC.

Elastic-Beanstalk-Worker-Umgebungen

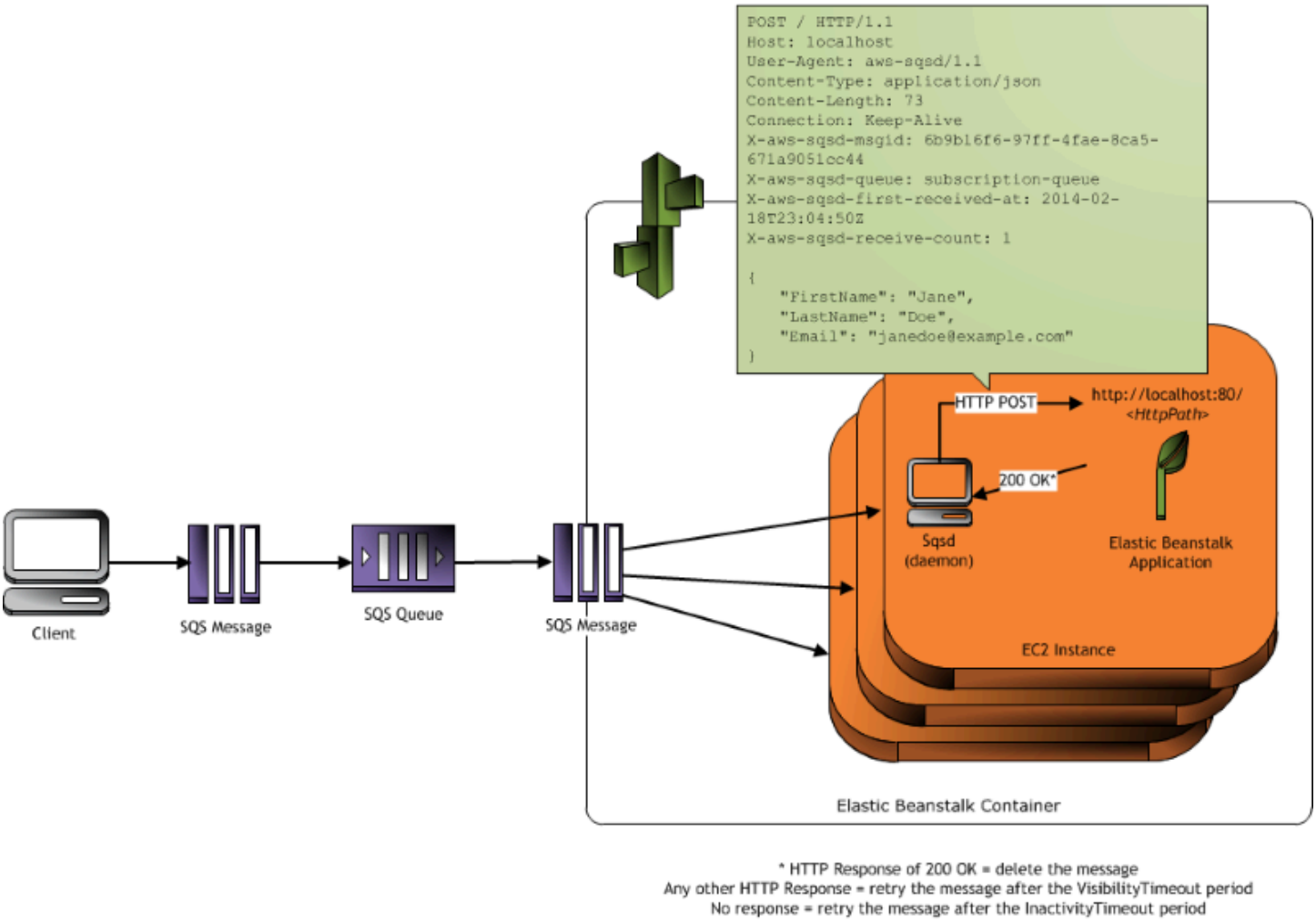
Wenn Ihre AWS Elastic Beanstalk Anwendung Operationen oder Workflows ausführt, deren Ausführung viel Zeit in Anspruch nimmt, können Sie diese Aufgaben auf eine spezielle Arbeitsumgebung auslagern. Das Frontend der Webanwendung von einem Prozess zu entkoppeln, der blockierende Vorgänge ausführt, ist ein gängiger Weg, um die Reaktionsfähigkeit der Anwendung auch bei Auslastung sicherzustellen.

Zu einer zeitaufwendigen Aufgabe zählen alle Vorgänge, die eine erhebliche Verlängerung der Anforderungsverarbeitung verursachen, also z. B. Bild- oder Videobearbeitung, Versenden von E-Mails oder Generieren eines ZIP-Archivs. Diese Vorgänge nehmen nur ein oder zwei Sekunden in Anspruch, jedoch ist eine Verzögerung von wenigen Sekunden ein hoher Wert für eine Webanforderung, die ansonsten in weniger als 500 ms verarbeitet wird.

Eine Möglichkeit besteht darin, einen Worker-Prozess lokal zu erstellen, erfolgreiche Antworten zurückzugeben und die Aufgabe asynchron zu verarbeiten. Das funktioniert, wenn die Instance alle gesendeten Aufgaben verarbeiten kann. Bei hoher Auslastung kann die Instance jedoch durch die Hintergrundaufgaben überlastet werden und daher nicht mehr auf Anforderungen mit höherer Priorität reagieren. Wenn einzelne Benutzer mehrere Aufgaben generieren können, entspricht die gestiegene Auslastung möglicherweise nicht einer höheren Benutzeranzahl, sodass keine effiziente Skalierung der Webserverebene mehr möglich ist.

Um zu vermeiden, dass Aufgaben mit langer Laufzeit lokal ausgeführt werden, können Sie das AWS SDK für Ihre Programmiersprache verwenden, um sie an eine Amazon Simple Queue Service (Amazon SQS) -Warteschlange zu senden und den Prozess, der sie ausführt, auf einer separaten Gruppe von Instances auszuführen. Die Worker-Instances verarbeiten Elemente aus der Warteschlange nur, wenn genügend Kapazität für deren Ausführung vorhanden ist. Auf diese Weise wird eine Überlastung verhindert.

Elastic Beanstalk Worker-Umgebungen vereinfachen diesen Prozess mithilfe der Amazon SQS-Warteschlangenverwaltung und der Ausführung eines Daemon-Prozesses auf jeder Instance, der Warteschlangenelemente liest. Wenn der Daemon ein Element aus der Warteschlange abruft, sendet er eine lokale HTTP POST-Anforderung auf Port 80 mit dem Textinhalt der Warteschlangennachricht an `http://localhost/`. Dann muss die Anwendung nur noch die zeitaufwendige Aufgabe als Reaktion auf die POST-Anforderung ausführen. Sie können den Daemon konfigurieren, um die POST-Anforderung an einen anderen Pfad zu senden, einen anderen MIME-Typ als Anwendung/JSON zu verwenden, die Verbindung zu einer vorhandenen Warteschlange herzustellen oder Verbindungen (maximale gleichzeitige Anforderungen), Timeouts und Wiederholungen einzurichten.



Mit regelmäßigen Aufgaben können Sie den Worker-Daemon so konfigurieren, dass er Nachrichten auf Basis eines Cron-Plans in die Warteschlange stellt. Jede periodische Aufgabe kann die POST-Anforderung an einen anderen Pfad senden. Aktivieren Sie periodische Aufgaben, indem Sie eine YAML-Datei, die den Plan und den Pfad für die einzelnen Aufgaben definiert, in den Quellcode einbinden.

> **Note**
>
> Die .NET-Plattform auf Windows Server unterstützt keine Worker-Umgebungen.

Sections

- SQS-Daemon in der Worker-Umgebung
- Warteschlangen mit unzustellbaren Briefen

- Regelmäßige Aufgaben
- Verwenden Sie Amazon CloudWatch für die automatische Skalierung in den Stufen der Mitarbeiterumgebung
- Konfigurieren von Worker-Umgebungen

SQS-Daemon in der Worker-Umgebung

In Worker-Umgebungen wird ein von Elastic Beanstalk bereitgestellter Daemon-Prozess ausgeführt. Dieser Daemon wird regelmäßig mit neuen Funktionen und Fehlerbehebungen aktualisiert. Für die neueste Daemon-Version aktualisieren Sie auf die neueste Plattformversion.

Wenn die Anwendung in der Worker-Umgebung als Antwort `200 OK` zurückgibt und damit den Empfang und die erfolgreiche Verarbeitung der Anforderung bestätigt, sendet der Daemon den Aufruf `DeleteMessage` an die Amazon SQS-Warteschlange, damit die Nachricht aus der Warteschlange gelöscht wird. Gibt die Anwendung eine andere Antwort als `200 OK` zurück, stellt Elastic Beanstalk die Nachricht nach Ablauf des konfigurierten `ErrorVisibilityTimeout`-Zeitraums wieder in die Warteschlange. Falls es gar keine Antwort gibt, platziert Elastic Beanstalk die Nachricht nach Verstreichen des konfigurierten `InactivityTimeout`-Zeitraums wieder in die Warteschlange, damit sie anderweitig verarbeitet werden kann.

> **Note**
>
> Die Eigenschaften von Amazon SQS SQS-Warteschlangen (Nachrichtenreihenfolge, at-least-once Versand und Nachrichtenabtastung) können sich darauf auswirken, wie Sie eine Webanwendung für eine Arbeitsumgebung entwerfen. Weitere Informationen finden Sie unter Eigenschaften verteilter Warteschlangen im Amazon Simple Queue Service-Entwicklerhandbuch.

Amazon SQS löscht automatisch Nachrichten, die sich länger als der konfigurierte `RetentionPeriod`-Zeitraum in der Warteschlange befinden.

Der Daemon legt folgende HTTP-Header fest.

HTTP-Header	
Name	Wert

HTTP-Header	

`User-Agent`	`aws-sqsd`
	
	`aws-sqsd/1.1 1`

`X-Aws-Sqsd-Msgid`	SQS-Nachrichten-ID, mit der Nachrichtenfluten (ungewöhnlich hohe Anzahl neuer Nachrichten) erkannt werden.
`X-Aws-Sqsd-Queue`	Gibt den Namen der SQS-Warteschlange an.
`X-Aws-Sqsd-First-Received-At`	Gibt den Zeitpunkt, an dem die Meldung zum ersten Mal empfangen wurde, in koordinierter Weltzeit (UTC) im Format ISO 8601 an.
`X-Aws-Sqsd-Receive-Count`	Gibt die Anzahl der empfangenen SQS-Nachr ichten an.
`X-Aws-Sqsd-Attr-` *`message-attribute-name`*	Benutzerdefinierte Attribute, die der gerade verarbeiteten Nachricht zugeordnet sind. Der tatsächliche Name des Nachrichtenattribu ts wird mit `message-attribute-name` angegeben. Alle Nachrichtenattribute im Zeichenfolgen- oder Ziffernformat werden zum Header hinzugefügt. Binary-Attribute werden verworfen und nicht in den Header aufgenomm en.
`Content-Type`	Gibt die Mime-Typkonfiguration an, standardm äßig `application/json` .

Warteschlangen mit unzustellbaren Briefen

Elastic Beanstalk-WorkerUmgebungen unterstützen Amazon Simple Queue Service (Amazon SQS) Dead-Letter-Warteschlangen. Andere (Quell-) Warteschlangen können Nachrichten, die aus irgendeinem Grund nicht erfolgreich verarbeitet werden konnten, an die Warteschlange für unzustellbare Nachrichten senden. Eine solche Warteschlange für unzustellbare Nachrichten bietet

den Vorteil, dass nicht erfolgreich verarbeitete Nachrichten ausgesondert und isoliert werden können. Anschließend können Sie alle Nachrichten analysieren, die an die Warteschlange für unzustellbare Nachrichten gesendet wurden, und ermitteln, warum die Verarbeitung nicht erfolgreich war.

Für eine Worker-Umgebung wird standardmäßig eine Warteschlange für unzustellbare Nachrichten aktiviert, wenn Sie bei der Erstellung der Worker-Umgebungsebene eine automatisch generierte Amazon SQS-Warteschlange angeben. Bei Auswahl einer vorhandenen SQS-Warteschlange für die Worker-Umgebung müssen Sie mithilfe von SQS eine Warteschlange für unzustellbare Nachrichten separat konfigurieren. Weitere Informationen zur Verwendung von SQS zum Konfigurieren einer Warteschlange für unzustellbare Nachrichten finden Sie unter Using Amazon SQS Dead Letter Queues.

Warteschlangen für unzustellbare Nachrichten können nicht deaktiviert werden. Nicht zustellbare Nachrichten werden letztlich immer an eine Warteschlange für unzustellbare Nachrichten gesendet. Sie können diese Funktion jedoch effektiv deaktivieren, indem Sie die Option `MaxRetries` auf den maximal zulässigen Wert von 100 festlegen.

Wenn keine Warteschlange für unzustellbare Nachrichten für die Amazon SQS-Warteschlange Ihrer Arbeitsumgebung konfiguriert ist, werden Nachrichten in der Warteschlange von Amazon SQS aufbewahrt, bis der Aufbewahrungszeitraum abgelaufen ist. Weitere Informationen zum Konfigurieren des Aufbewahrungszeitraums finden Sie unter the section called “Konfigurieren von Worker-Umgebungen”.

> **Note**
>
> Die Elastic Beanstalk `MaxRetries`-Option entspricht der SQS `MaxReceiveCount`-Option. Sofern in der Worker-Umgebung keine automatisch generierte SQS-Warteschlange verwendet wird, lässt sich die Warteschlange für unzustellbare Nachrichten effektiv mit der SQS-Option `MaxReceiveCount` deaktivieren. Weitere Informationen finden Sie unter Using Amazon SQS Dead Letter Queues.

Weitere Informationen zum Lebenszyklus einer SQS-Nachricht finden Sie unter Message Lifecycle.

Regelmäßige Aufgaben

Sie können periodische Aufgaben in einer `cron.yaml`-Datei im Quell-Bundle definieren, damit in regelmäßigen Abständen automatisch Aufträge zur Warteschlange der Worker-Umgebung hinzugefügt werden.

Die folgende `cron.yaml`-Datei erstellt beispielsweise zwei periodische Aufgaben. Die erste läuft alle 12 Stunden und die zweite läuft täglich um 23:00 Uhr UTC.

Example cron.yaml

```
version: 1
cron:
 - name: "backup-job"
   url: "/backup"
   schedule: "0 */12 * * *"
 - name: "audit"
   url: "/audit"
   schedule: "0 23 * * *"
```

Die Eingabe für **`name`** muss für jede Aufgabe eindeutig sein. Die URL ist der Pfad, an den die POST-Anforderung gesendet wird, um den Auftrag auszulösen. Die Planung erfolgt über einen CRON-Ausdruck, mit dem der Ausführungszeitpunkt der Aufgabe festgelegt wird.

Wenn eine Aufgabe ausgeführt wird, sendet der Daemon eine Nachricht mit einem Header, in dem der auszuführende Auftrag beschrieben wird, an die SQS-Warteschlange der Umgebung. Jede Instance der Umgebung kann die Nachricht annehmen und den Auftrag verarbeiten.

> **Note**
>
> Bei Konfiguration der Worker-Umgebung mit einer vorhandenen SQS-Warteschlange müssen Sie eine Amazon SQS-FIFO-Warteschlange wählen, regelmäßige Aufgaben werden nicht unterstützt.

Elastic Beanstalk bestimmt anhand der Leader-Wahl, welche Instance der Worker-Umgebung die periodische Aufgabe in die Warteschlange stellt. Jede Instance versucht, durch Schreiben in eine Amazon DynamoDB-Tabelle zum Leader zu werden. Die erste erfolgreiche Instance fungiert als Leader und muss weiterhin in die Tabelle schreiben, um den Leader-Status zu behalten. Wenn der Leader ausfällt, übernimmt rasch eine andere Instance.

Bei regelmäßigen Aufgaben legt der Worker-Daemon die folgenden zusätzlichen Header fest.

HTTP-Header

Name	Wert
`X-Aws-Sqsd-Taskname`	Bei regelmäßigen Aufgaben ist dies der Name der auszuführenden Aufgabe.
`X-Aws-Sqsd-Scheduled-At`	Gibt den Zeitpunkt an, für den die regelmäßige Aufgabe geplant war.
`X-Aws-Sqsd-Sender-Id`	AWS Kontonummer des Absenders der Nachricht

Verwenden Sie Amazon CloudWatch für die automatische Skalierung in den Stufen der Mitarbeiterumgebung

Gemeinsam CloudWatch überwachen Amazon EC2 Auto Scaling und die CPU-Auslastung der laufenden Instances in der Worker-Umgebung. Mit der Konfiguration des Auto Scaling-Limits für die CPU-Kapazität geben Sie an, wie viele Instances von der Auto Scaling-Gruppe zur Verarbeitung des Nachrichtendurchsatzes in der Amazon SQS-Warteschlange ausgeführt werden. Jede EC2-Instance veröffentlicht ihre CPU-Nutzungsmetriken an. CloudWatch Amazon EC2 Auto Scaling verwendet CloudWatch die durchschnittliche CPU-Auslastung aller Instances in der Worker-Umgebung.
Sie konfigurieren den oberen und den unteren Grenzwert sowie die Anzahl der Instances, die entsprechend der CPU-Kapazität hinzugefügt oder beendet werden sollen. Wenn von der Amazon EC2 Auto Scaling-Funktion erkannt wird, dass der angegebene obere Grenzwert für die CPU-Kapazität erreicht ist, erstellt Elastic Beanstalk neue Instances in der Worker-Umgebung. Sinkt die CPU-Auslastung wieder unter den Grenzwert, werden die Instances gelöscht.

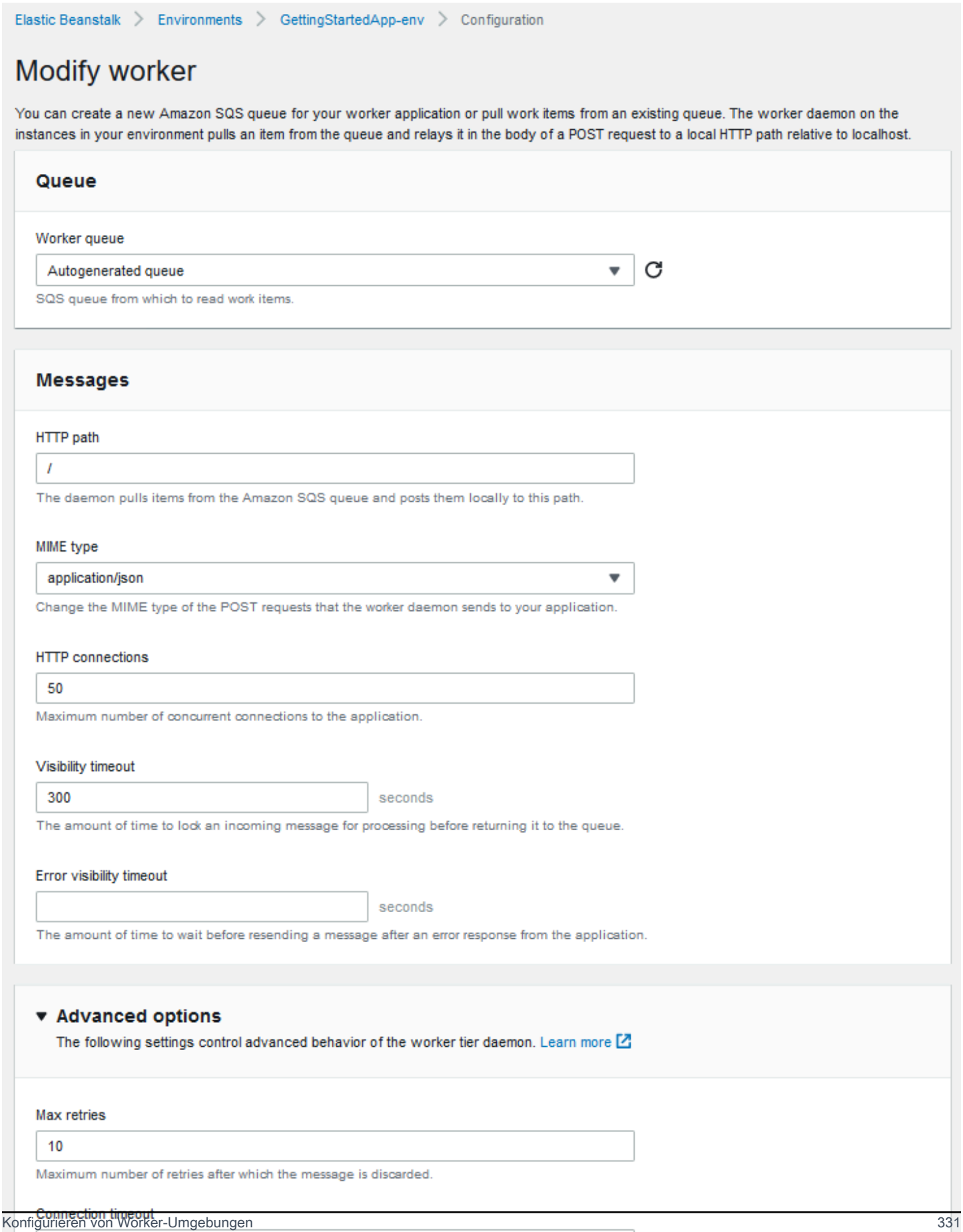
> **Note**
>
> Sofern beim Beenden einer Instance noch nicht verarbeitete Nachrichten vorhanden sind, werden diese an die Warteschlange zurückgegeben und können von einem anderen Daemon auf einer ausgeführten Instance verarbeitet werden.

Sie können bei Bedarf auch andere CloudWatch Alarme einrichten, indem Sie die Elastic Beanstalk Beanstalk-Konsole, CLI oder die Optionsdatei verwenden. Weitere Informationen finden Sie unter

Elastic Beanstalk mit Amazon verwenden CloudWatch und Erstellen einer Auto Scaling-Gruppe mit Richtlinien für die schrittweise Skalierung.

Konfigurieren von Worker-Umgebungen

Die Verwaltung der Worker-Umgebungskonfiguration erfolgt über die Bearbeitung der Kategorie Worker auf der Seite Configuration (Konfiguration) in der Environment Management Console.


Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration
Modify worker
You can create a new Amazon SQS queue for your worker application or pull work items from an existing queue. The worker daemon on the instances in your environment pulls an item from the queue and relays it in the body of a POST request to a local HTTP path relative to localhost.
Queue
Worker queue
Autogenerated queue
SQS queue from which to read work items.
Messages
HTTP path
/
The daemon pulls items from the Amazon SQS queue and posts them locally to this path.
MIME type
application/json
Change the MIME type of the POST requests that the worker daemon sends to your application.
HTTP connections
50
Maximum number of concurrent connections to the application.
Visibility timeout
300
seconds
The amount of time to lock an incoming message for processing before returning it to the queue.
Error visibility timeout
seconds
The amount of time to wait before resending a message after an error response from the application.
Advanced options
The following settings control advanced behavior of the worker tier daemon. Learn more
Max retries
10
Maximum number of retries after which the message is discarded.
Connection timeout
5
Inactivity timeout

> **Note**
>
> Sie können die URL-Pfad für die Veröffentlichung von Worker-Warteschlangen-Nachrichten konfigurieren, nicht aber den IP-Port. Elastic Beanstalk sendet Worker-Warteschlangennachrichten immer über Port 80. Die Anwendung der Worker-Umgebung oder ihr Proxy müssen Port 80 überwachen.

So konfigurieren Sie den Worker-Daemon

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Worker die Option Edit (Bearbeiten).

Die Konfigurationsseite Modify worker (Worker ändern) verfügt über die folgenden Optionen.

Im Abschnitt Queue (Warteschlange):

- Worker queue – Geben Sie die Amazon SQS-Warteschlange an, die vom Daemon gelesen wird. Sofern verfügbar, können Sie eine vorhandene Warteschlange auswählen. Wenn Sie Autogenerated queue (Automatisch generierte Warteschlange) auswählen, erstellt Elastic Beanstalk eine neue Amazon SQS-Warteschlange und eine entsprechende Worker-Warteschlangen-URL.

  > **Note**
  >
  > Wenn Sie Autogenerated queue (Automatisch generierte Warteschlange) auswählen, erstellt Elastic Beanstalk eine Amazon SQS-Standard-Warteschlange. Wenn Sie eine vorhandene Warteschlange auswählen, können Sie entweder eine Standard- oder eine FIFO--Amazon SQS-Warteschlange angeben. Beachten Sie, dass bei Angabe einer FIFO-Warteschlange regelmäßige Aufgaben nicht unterstützt werden.

- Worker queue URL (URL für Worker-Warteschlange) – Wenn Sie eine vorhandene Worker queue (Worker-Warteschlange) auswählen, wird in dieser Einstellung die URL für diese Amazon SQS-Warteschlange angezeigt.

Im Abschnitt Messages (Nachrichten):

- HTTP path – Geben Sie den relativen Pfad zur Anwendung an, die Daten von der Amazon SQS-Warteschlange empfängt. Die Daten werden in den Nachrichtentext einer HTTP POST-Nachricht eingebunden. Der Standardwert ist `/`.
- MIME type (MIME-Typ) – Geben Sie den von der HTTP POST-Nachricht verwendeten MIME-Typ an. Der Standardwert ist `application/json`. Allerdings sind alle Werte gültig, da Sie einen eigenen MIME-Typ erstellen und dann angeben können.
- HTTP connections (HTTP-Verbindungen) – Geben Sie die maximale Anzahl gleichzeitiger Verbindungen an, die vom Daemon zu jeder Anwendung auf einer Amazon EC2-Instance hergestellt werden kann. Der Standardwert ist **`50`**. Sie können **`1`** bis **`100`** eingeben.
- Visibility timeout – Geben Sie die Zeitspanne, die eine eingehende Nachricht aus der Amazon SQS-Warteschlange zur Verarbeitung gesperrt ist, in Sekunden an. Nach Ablauf des konfigurierten Zeitraums wird die Nachricht in der Warteschlange wieder sichtbar und kann von einem anderen Daemon gelesen werden. Wählen Sie einen Wert aus, der über der erwarteten Verarbeitungszeit der Nachricht durch die Anwendung liegt, bis max. **`43200`** Sekunden.
- Error visibility timeout – Geben Sie die zu verstreichende Zeitspanne, bevor Elastic Beanstalk eine Nachricht nach einem fehlgeschlagenen Verarbeitungsversuch mit einem expliziten Fehler an die Amazon SQS-Warteschlange zurückgibt, in Sekunden an. Sie können **`0`** bis **`43200`** Sekunden eingeben.

Führen Sie im Abschnitt Advanced options (Erweiterte Optionen) Folgendes durch:

- Max retries – Geben Sie die maximale Anzahl der Wiederholversuche für Elastic Beanstalk vor, um die Nachricht an die Amazon SQS-Warteschlange zu senden, bevor die Nachricht in die Warteschlange für unzustellbare Nachrichten verschoben wird. Der Standardwert ist **`10`**. Sie können **`1`** bis **`100`** eingeben.

> **Note**
>
> Die Option Max retries (Maximale Wiederholungen) gilt nur für Amazon-SQS-Warteschlangen, die mit einer Warteschlange für unzustellbare Nachrichten konfiguriert

sind. Für alle Amazon-SQS-Warteschlangen, die nicht mit einer Warteschlange für unzustellbare Nachrichten konfiguriert sind, werden Nachrichten in der Warteschlange von Amazon SQS aufbewahrt und verarbeitet, bis der durch die Option Retention period (Aufbewahrungszeitraum) angegebene Zeitraum abläuft.

- Connection timeout (Zeitbeschränkung für Verbindung) – Geben Sie die Zeitspanne, die auf erfolgreiche Verbindungserstellungen zu einer Anwendung gewartet wird, in Sekunden an. Der Standardwert ist **5**. Sie können **1** bis **60** Sekunden eingeben.
- Inactivity timeout (Zeitbeschränkung für Inaktivität) – Geben Sie die Zeitspanne, die bei einer bestehenden Verbindung zu einer Anwendung auf eine Antwort gewartet wird, in Sekunden an. Der Standardwert ist **180**. Sie können **1** bis **36000** Sekunden eingeben.
- Retention period (Aufbewahrungszeitraum) – Geben Sie die Zeitspanne, die eine Nachricht gültig ist und aktiv verarbeitet wird, in Sekunden an. Der Standardwert ist **345600**. Sie können **60** bis **1209600** Sekunden eingeben.

Bei Verwendung einer vorhandenen Amazon SQS-Warteschlange kann zwischen den Einstellungen, die Sie bei der Worker-Umgebungserstellung konfiguriert haben, und den direkt von Ihnen in Amazon SQS konfigurierten Einstellungen ein Konflikt entstehen. Wenn beispielsweise der für eine Worker-Umgebung konfigurierte `RetentionPeriod`-Wert über dem in Amazon SQS festgelegten `MessageRetentionPeriod`-Wert ist, wird die Nachricht bei einer Überschreitung des `MessageRetentionPeriod`-Werts von Amazon SQS gelöscht.

Umgekehrt gilt dasselbe: Wenn der in den Worker-Umgebungseinstellungen konfigurierte `RetentionPeriod`-Wert niedriger ist als der in Amazon SQS definierte `MessageRetentionPeriod`-Wert, wird die Nachricht vom Daemon gelöscht, bevor Amazon SQS den Löschvorgang ausführt. Der für den Daemon in den Worker-Umgebungseinstellungen konfigurierte Wert setzt den in Amazon SQS eingestellten `VisibilityTimeout`-Wert für `VisibilityTimeout` außer Kraft. Stellen Sie sicher, dass Nachrichten angemessen gelöscht werden, indem Sie Ihre Elastic Beanstalk-Einstellungen mit Ihren Amazon SQS-Einstellungen vergleichen.

Verknüpfungen zwischen Elastic Beanstalk-Umgebungen erstellen

Mit zunehmender Anwendungsgröße und -komplexität kann es sinnvoll sein, sie in Komponenten mit unterschiedlichen Entwicklungs- und Betriebslebenszyklen zu untergliedern. Durch die Ausführung von kleineren Services, die über eine klar definierte Schnittstelle miteinander interagieren, können

Teams unabhängig voneinander arbeiten und die Bereitstellungen repräsentieren ein geringeres Risiko. Mit AWS Elastic Beanstalk können Sie die Umgebungen verknüpfen, um Informationen zwischen Komponenten freizugeben, die voneinander abhängig sind.

> **Note**
>
> Derzeit unterstützt Elastic Beanstalk Umgebungslinks für alle Plattformen (außer Multicontainer Docker).

Mithilfe von Umgebungslinks können Sie die Verbindungen zwischen den Umgebungen der Anwendungskomponenten als namentliche Referenzen angeben. Wenn Sie eine Umgebung mit einer Linkdefinition erstellen, legt Elastic Beanstalk eine Umgebungsvariable mit dem gleichen Namen des Links fest. Der Variablenwert ist der Endpunkt, über den Sie die Verbindung zur anderen Komponente herstellen; dabei kann es sich um eine Webserver- oder Worker-Umgebung handeln.

Angenommen, Ihre Anwendung besteht aus einem Frontend zur Erfassung von E-Mail-Adressen und einem Worker, der Begrüßungs-E-Mails an die vom Frontend gesammelten E-Mail-Adressen sendet. Nun können Sie in Ihrem Frontend einen Link zum Worker einrichten, damit das Frontend automatisch den Endpunkt (Warteschlangen-URL) des Workers erkennt.

Definieren Sie Links zu anderen Umgebungen in einem Umgebungsmanifest. Dies ist eine Datei im YAML-Format mit dem Namen `env.yaml` im Stamm der Anwendungsquelle. In folgendem Manifest wird ein Link zu einer Umgebung mit dem Namen "worker" definiert:

`~/workspace/my-app/frontend/env.yaml`

```
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentLinks:
  "WORKERQUEUE": "worker"
```

Wenn Sie eine Umgebung mit einer Anwendungsversion erstellen, in der obiges Umgebungsmanifest enthalten ist, sucht Elastic Beanstalk nach einer Umgebung mit dem Namen `worker`, die zu dieser Anwendung gehört. Wenn diese Umgebung vorhanden ist, erstellt Elastic Beanstalk eine Umgebungseigenschaft namens `WORKERQUEUE`. Der Wert von `WORKERQUEUE` ist die URL der Amazon SQS-Warteschlange. Die Frontend-Anwendung kann diese Eigenschaft auf dieselbe Weise lesen wie eine Umgebungsvariable. Details dazu finden Sie unter Umgebungsmanifest (`env.yaml`).

Um Umgebungslinks zu verwenden, fügen Sie Ihrer Anwendungsquelle ein Umgebungsmanifest hinzu und laden Sie es mit der EB-CLI AWS CLI oder einem SDK hoch. Wenn Sie das AWS CLI oder ein SDK verwenden, setzen Sie das `process` Flag, wenn Sie Folgendes aufrufen`CreateApplicationVersion`:

```
$ aws elasticbeanstalk create-application-version --process --application-name
 my-app --version-label frontend-v1 --source-bundle S3Bucket="amzn-s3-demo-
bucket",S3Key="front-v1.zip"
```

Mit dieser Option wird Elastic Beanstalk angewiesen, das Umgebungsmanifest und die Konfigurationsdateien im Quell-Bundle zu validieren, wenn Sie die Anwendungsversion erstellen. In der EB CLI wird das Flag automatisch gesetzt, sobald sich ein Umgebungsmanifest im Projektverzeichnis befindet.

Erstellen Sie die Umgebungen ganz regulär mit einem beliebigen Client. Wenn Umgebungen beendet werden müssen, beenden Sie zuerst die Umgebung mit dem Link. Falls eine Umgebung von einer anderen Umgebung verlinkt ist, verhindert Elastic Beanstalk, dass die verlinkte Umgebung beendet wird. Diese Schutzmaßnahme setzen Sie mit dem Flag `ForceTerminate` außer Kraft. In der AWS CLI ist dieser Parameter als `--force-terminate` verfügbar:

```
$ aws elasticbeanstalk terminate-environment --force-terminate --environment-name
 worker
```

Wiederherstellung Ihrer Elastic Beanstalk Beanstalk-Umgebung aus einem ungültigen Zustand

Dieses Thema enthält einige Hintergrundinformationen und Ressourcen, die erklären, wie Sie Fehler in einer Elastic Beanstalk Beanstalk-Umgebung in einem ungültigen Zustand beheben können.

Behebung des Fehlers

Standardvorgänge in einer Umgebung mit einem ungültigen Status werden nicht erfolgreich abgeschlossen. Der fehlgeschlagene Vorgang gibt einen Fehler zurück, der den folgenden Text enthält:

```
The stack stack_id associated with environment environment-ID is in stack-status state.
```

Informationen zur Fehlerbehebung und Behebung dieses Fehlers finden Sie im Knowledge Center-Artikel Warum befindet sich meine Elastic Beanstalk Beanstalk-Umgebung im ungültigen Zustand? .

> **Note**
>
> Vor dem 16. Dezember 2024 gab der fehlgeschlagene Vorgang stattdessen den folgenden Fehler zurück: `Environment is in an invalid state for this operation. Must be ready.` In diesem Fall mussten Sie sich an den AWS Support wenden, um den Umgebungsstatus zurückzusetzen, nachdem Sie die Korrekturmaßnahmen abgeschlossen hatten.
> Heute müssen Sie die Stack-Probleme immer noch anhand der Anweisungen im Knowledge Center-Artikel lösen, auf den verwiesen wird. Sobald Sie die Korrekturmaßnahmen jedoch erfolgreich abgeschlossen haben, aktualisiert Elastic Beanstalk den Status der Umgebung automatisch von ungültig auf verfügbar, sodass Sie die Standardoperationen in Ihrer Umgebung ohne weitere Verzögerung wieder aufnehmen können.

Warum tritt der Fehler auf

Wenn Sie eine Anwendung in Elastic Beanstalk bereitstellen, erstellt der Service einen zugrunde liegenden AWS CloudFormation Stack. Elastic Beanstalk ruft den CloudFormation Service auf, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren.

Wenn Elastic Beanstalk einen Vorgang in einer Umgebung ausführt, ohne Zugriff auf eine erforderliche Ressource zu haben, kann der der Umgebung zugrunde liegende CloudFormation Stack in einen ausgefallenen Zustand übergehen. Andere Probleme können ebenfalls zu diesem Status führen, obwohl Berechtigungsprobleme die Hauptursache sind. Aufgrund des ausgefallenen Status des Stacks werden Elastic Beanstalk Beanstalk-Operationsanforderungen daran CloudFormation gehindert, weitere Stack-Updates durchzuführen, was zum Ausfall von Elastic Beanstalk-Vorgängen wie und führt zum Ausfall von Elastic Beanstalk-Vorgängen. UpdateEnvironment RetrieveEnvironmentInfo

An diesem Punkt müssen Sie zunächst die Ursache des zugrunde liegenden Problems korrigieren, um den Stack zu beheben. CloudFormation Der Elastic Beanstalk-Service erkennt dann die Änderung des CloudFormation Stack-Status und setzt Ihre Umgebung anschließend auf einen verfügbaren Status zurück. Zu diesem Zeitpunkt können weitere Operationen erfolgreich abgeschlossen werden.

Berechtigungsprobleme haben in der Regel diese Auswirkung auf den CloudFormation Stack und die Elastic Beanstalk Beanstalk-Umgebung, obwohl out-of-band Änderungen auch Probleme verursachen können.

> **Important**
>
> Um Unterbrechungen Ihrer Umgebung zu vermeiden, empfehlen wir dringend, nur Vorgänge zur Verwaltung und Konfiguration Ihrer Umgebung über den Elastic Beanstalk-Service zu starten. Die Änderung von Ressourcen mithilfe der Konsole, der CLI-Befehle oder des SDK eines anderen Dienstes als Elastic Beanstalk ist eine out-of-band Änderung, die zu einer Verschiebung der Ressourcen führt. Die Ressourcenverschiebung wirkt sich auf den Status des CloudFormation Stacks aus, was wiederum dazu führt, dass die Elastic Beanstalk Beanstalk-Umgebung in einen ungültigen Zustand übergeht.
> Weitere Informationen zur Ressourcenverschiebung finden Sie unter Was ist Drift? im AWS CloudFormation Benutzerhandbuch.

Konfigurieren von Elastic-Beanstalk-Umgebungen

Dieses Thema konzentriert sich auf die Konfigurationsoptionen, die in der Elastic Beanstalk Beanstalk-Konsole verfügbar sind. AWS Elastic Beanstalk bietet eine Vielzahl von Optionen für die Anpassung der Ressourcen in Ihrer Umgebung sowie das Verhalten und die Plattformeinstellungen von Elastic Beanstalk.

In den folgenden Themen wird erläutert, wie Sie Ihre Umgebung in der Konsole konfigurieren. Außerdem werden die zugrunde liegenden Namespaces beschrieben, die den Konsolenoptionen für Konfigurationsdateien oder API-Konfigurationsoptionen entsprechen. Weitere Informationen zu erweiterten Konfigurationsmethoden finden Sie unter Konfigurieren von Umgebungen (Erweitert).

Themen

- Bereitgestellte Ressourcen
- Umgebungskonfiguration mit der Elastic Beanstalk-Konsole
- Die EC2 Amazon-Instances für Ihre Elastic Beanstalk Beanstalk-Umgebung
- Auto Scaling Ihrer Elastic Beanstalk Beanstalk-Umgebungsinstanzen
- Load Balancer Ihrer Elastic-Beanstalk-Umgebung
- Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung
- Sicherheit Ihrer AWS Elastic Beanstalk Umgebung
- Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren
- Umgebungsvariablen und andere Softwareeinstellungen
- Elastic Beanstalk-Umgebungsbenachrichtigungen mit Amazon SNS
- Amazon Virtual Private Cloud (Amazon VPC) mit Elastic Beanstalk konfigurieren
- Der Domainname Ihrer Elastic Beanstalk-Umgebung

Bereitgestellte Ressourcen

Wenn Sie eine Webserver-Umgebung erstellen, erstellt Elastic Beanstalk mehrere Ressourcen, um den Betrieb Ihrer Anwendung zu unterstützen. In diesem Kapitel wird beschrieben, wie Sie diese Ressourcen für Ihre Elastic Beanstalk Beanstalk-Umgebung anpassen können.

- EC2 instance (EC2-Instance) – eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die so konfiguriert ist, dass Web-Apps auf der von Ihnen ausgewählten Plattform ausgeführt werden.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Web-App erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Domain-Sicherheit

Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

Umgebungskonfiguration mit der Elastic Beanstalk-Konsole

In diesem Thema werden die Konfigurationsoptionen beschrieben, die über die Elastic Beanstalk Beanstalk-Konsole verfügbar sind, und es wird erklärt, wie Sie auf den Konfigurationsseiten navigieren.

So zeigen Sie eine Zusammenfassung Ihrer Umgebungskonfiguration an

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

Konfigurationsseite

Auf der Seite Configuration overview (Konfigurationsübersicht) wird eine Gruppe von Konfigurationskategorien angezeigt. Jede Konfigurationskategorie gruppiert eine Reihe verwandter Optionen.

Zugriff auf Services

Die Optionen in dieser Kategorie wählen die Servicerolle und das EC2 Instanzprofil aus, das Elastic Beanstalk zur Verwaltung Ihrer Umgebung verwendet. Wählen Sie optional ein EC2 key pair, um sich sicher bei Ihren EC2 Instances anzumelden.

Netzwerk und Datenbank

Die Optionen in dieser Kategorie konfigurieren VPC-Einstellungen und Subnetze für die EC2 Instances und den Load Balancer der Umgebung. Sie bieten auch die Möglichkeit, eine Amazon RDS-Datenbank einzurichten, die in Ihre Umgebung integriert ist.

Instance-Verkehr und Skalierung

Diese Optionen passen die Kapazität, Skalierung und den Lastausgleich für die EC2 Instanzen der Umgebung an. Sie können Elastic Load Balancing auch so konfigurieren, dass es Logs mit detaillierten Informationen über Anfragen erfasst, die an den Load Balancer gesendet werden.

Die folgenden Optionen für Ihre EC2 Instances sind ebenfalls für die Konfiguration verfügbar:

- Typ, Größe, input/output Betriebsrate (IOPS) und Durchsatz des Root-Volumes.
- Aktivieren des Instanz-Metadatendienstes (IMDS).
- Auswahl von EC2 Sicherheitsgruppen zur Steuerung des Instanzverkehrs.
- CloudWatch Intervall zur Überwachung von Metriken.
- Zeitintervall für die Protokollierung von Metriken.

Aktualisierungen, Überwachung und Protokollierung

In dieser Kategorie werden die folgenden Optionen konfiguriert:

- Umweltgesundheitsberichterstattung, einschließlich der Option, erweiterte Gesundheitsberichterstattung auszuwählen.
- Verwaltete Plattform-Updates, die definieren, wann und wie Elastic Beanstalk Änderungen an der Umgebung einführt.
- Aktivierung des X-Ray-Dienstes zur Erfassung von Daten über das Verhalten Ihrer Anwendung, um Probleme und Optimierungsmöglichkeiten zu identifizieren.
- Plattformspezifische Optionen, einschließlich der Eigenschaften des Proxyservers und der Betriebssystemumgebung.

Auf der Konfigurationsseite navigieren

Wählen Sie in einer Konfigurationskategorie Bearbeiten aus, um die zugehörige Konfigurationsseite aufzurufen, auf der Sie die vollständigen Optionswerte sehen und Änderungen vornehmen können.

Navigation in einer Konfigurationskategorie

Navigieren Sie mit einer der folgenden Aktionen auf einer Seite mit einer der folgenden Kategorien zu einer Konfigurationskategorie:

- Abbrechen — Kehren Sie zur Seite mit der Konfigurationsübersicht zurück, ohne Ihre Konfigurationsänderungen zu übernehmen. Wenn Sie Cancel (Abbrechen) auswählen, gehen ausstehende Änderungen, die Sie an einer Konfigurationskategorie vorgenommen haben, in der Konsole verloren.

  Sie können Ihre Konfigurationsänderungen auch stornieren, indem Sie auf der linken Navigationsseite ein anderes Element auswählen, z. B. Ereignisse oder Protokolle.
- Continue (Weiter) – Gehen Sie zur Seite Configuration overview (Konfigurationsübersicht) zurück. Anschließend können Sie weitere Änderungen vornehmen oder ausstehende Änderungen übernehmen.
- Anwenden — Wenden Sie die Änderungen, die Sie in einer der Konfigurationskategorien vorgenommen haben, auf Ihre Umgebung an. In einigen Fällen werden Sie dazu aufgefordert, zu bestätigen, dass Sie mit den Konsequenzen Ihrer Konfigurationsentscheidungen einverstanden sind.

Navigation auf der Seite mit der Übersicht über die Konfiguration

Wählen Sie in einer Konfigurationskategorie Bearbeiten aus, um eine zugehörige Konfigurationsseite aufzurufen, auf der Sie die vollständigen Optionswerte sehen und Änderungen vornehmen können. Wenn Sie mit dem Anzeigen und Ändern der Optionen fertig sind, können Sie auf der Konfigurationsübersichtsseite eine der folgenden Aktionen auswählen:

- Cancel (Abbrechen) – Gehen Sie zurück zum Dashboard Ihrer Umgebung, ohne die Konfigurationsänderungen zu übernehmen. Wenn Sie Cancel (Abbrechen) auswählen, gehen ausstehende Änderungen, die Sie an einer Konfigurationskategorie vorgenommen haben, in der Konsole verloren.

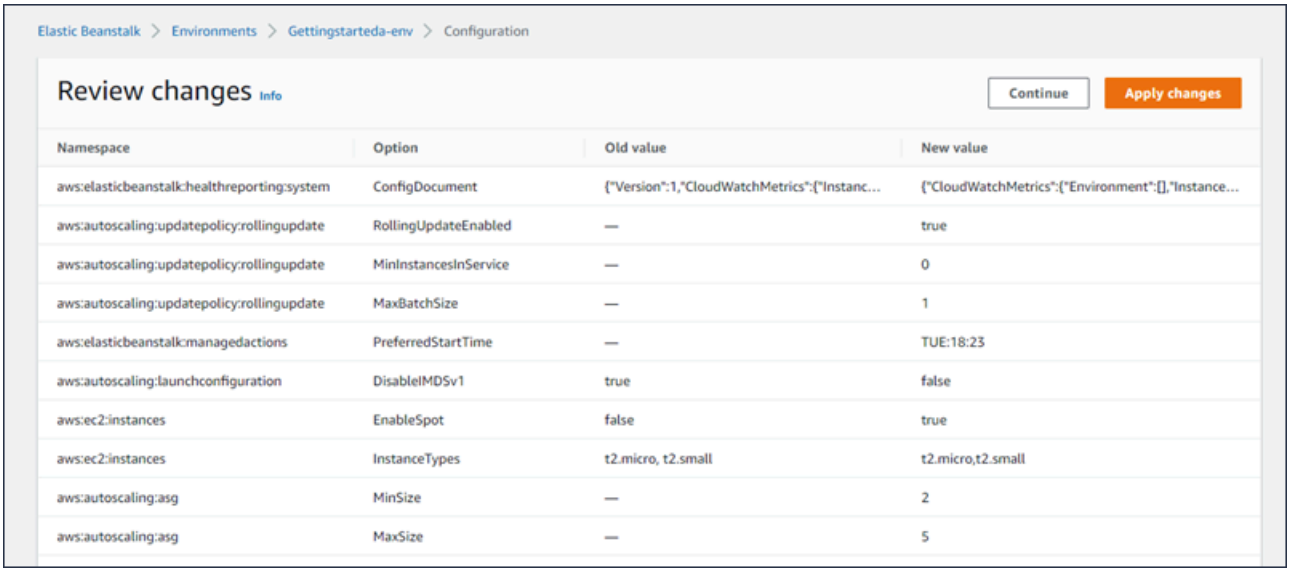
  Sie können Ihre Konfigurationsänderungen auch rückgängig machen, indem Sie auf der linken Navigationsseite ein anderes Element auswählen, z. B. Ereignisse oder Protokolle.
- Review changes (Änderungen überprüfen) – Rufen Sie eine Zusammenfassung aller ausstehenden Änderungen ab, die Sie an einer Konfigurationskategorie vorgenommen haben. Details hierzu finden Sie unter Seite "Änderungen überprüfen".

- Apply changes (Änderungen übernehmen) – Übernehmen Sie die Änderungen, die Sie an einer Konfigurationskategorie in Ihrer Umgebung vorgenommen haben. In einigen Fällen werden Sie dazu aufgefordert, zu bestätigen, dass Sie mit den Konsequenzen Ihrer Konfigurationsentscheidungen einverstanden sind.

Seite "Änderungen überprüfen"

Auf der Seite Review changes (Änderungen überprüfen) wird eine Tabelle mit allen ausstehenden Optionsänderungen angezeigt, die Sie an einer Konfigurationskategorie vorgenommen und noch nicht in Ihrer Umgebung übernommen haben.

In den Tabellen werden die jeweiligen Optionen als Kombination aus Namespace (Namespace) und Option (Option) aufgelistet, anhand derer diese in Elastic Beanstalk identifiziert werden. Details hierzu finden Sie unter Konfigurationsoptionen.



Nach dem Überprüfen Ihrer Änderungen können Sie eine der folgenden Aktionen auswählen:

- Continue (Weiter) – Gehen Sie zur Seite Configuration overview (Konfigurationsübersicht) zurück. Anschließend können Sie weitere Änderungen vornehmen oder ausstehende Änderungen übernehmen.
- Apply changes (Änderungen übernehmen) – Übernehmen Sie die Änderungen, die Sie an einer Konfigurationskategorie in Ihrer Umgebung vorgenommen haben. In einigen Fällen werden Sie dazu aufgefordert, zu bestätigen, dass Sie mit den Konsequenzen Ihrer Konfigurationsentscheidungen einverstanden sind.

Die EC2 Amazon-Instances für Ihre Elastic Beanstalk Beanstalk-Umgebung

In diesem Thema werden die EC2 Amazon-Instances und die Konfigurationsoptionen erläutert, die sich auf Ihre Elastic Beanstalk Beanstalk-Umgebung auswirken.

Wenn Sie eine Webserver-Umgebung erstellen, AWS Elastic Beanstalk erstellt eine oder mehrere virtuelle Amazon Elastic Compute Cloud (Amazon EC2) -Maschinen, die als Instances bezeichnet werden.

Die Instances in Ihrer Umgebung sind so konfiguriert, dass Web-Apps auf der von Ihnen ausgewählten Plattform ausgeführt werden. Sie können Änderungen an verschiedenen Eigenschaften und Verhaltensweisen der Instances Ihrer Umgebung vornehmen, wenn Sie Ihre Umgebung erstellen oder nachdem sie bereits ausgeführt wurde. Sie können diese Änderungen auch bereits vornehmen, indem Sie den Quellcode ändern, den Sie in der Umgebung bereitstellen. Weitere Informationen finden Sie unter the section called “Konfigurationsoptionen”.

> **Note**
>
> Die Auto Scaling Scaling-Gruppe in Ihrer Umgebung verwaltet die EC2 Amazon-Instances, auf denen Ihre Anwendung ausgeführt wird. Wenn Sie Konfigurationsänderungen vornehmen, die in diesem Thema beschrieben werden, ändert sich auch die Startkonfiguration. Die Startkonfiguration ist entweder eine EC2 Amazon-Startvorlage oder eine Auto Scaling Scaling-Gruppenstartkonfigurationsressource. Diese Änderung erfordert den Ersatz aller Instances. Sie löst auch entweder eine Laufende Aktualisierung oder eine unveränderliche Aktualisierung aus, je nachdem was konfiguriert ist.

EC2 Kaufoptionen für Instances

Elastic Beanstalk unterstützt mehrere Kaufoptionen für EC2 Amazon-Instances:

- On-Demand-Instance — Eine On-Demand-Instance ist eine pay-as-you-go Ressource, für deren Nutzung keine langfristige Bindung erforderlich ist.
- Reserviert — Eine Reserved Instance ist ein im Voraus gekaufter Rechnungsrabatt, der automatisch auf entsprechende On-Demand-Instances in Ihrer Umgebung angewendet wird.
- Spot — Eine Spot-Instance ist eine ungenutzte EC2 Amazon-Instance, die für weniger als den On-Demand-Preis erhältlich ist. Sie können die Zuweisung von Spot-Instances in Ihrer Umgebung

aktivieren und konfigurieren. Weitere Informationen finden Sie unter Auto Scaling Ihrer Elastic Beanstalk Beanstalk-Umgebungsinstanzen.

Themen

- EC2 Amazon-Instance-Typen
- Konfiguration von EC2 Amazon-Instances mit der Elastic Beanstalk Beanstalk-Konsole
- EC2-Sicherheitsgruppen verwalten
- Konfiguration von Amazon EC2-Sicherheitsgruppen und Instance-Typen mit dem AWS CLI
- Konfiguration von EC2 Amazon-Instances mit Namespace-Optionen
- Konfiguration des IMDS auf den Instances Ihrer Elastic Beanstalk Beanstalk-Umgebung

EC2 Amazon-Instance-Typen

In diesem Thema wird der Begriff Instance-Typ erklärt. Wenn Sie eine neue Umgebung erstellen, stellt Elastic Beanstalk EC2 Amazon-Instances bereit, die auf den von Ihnen Amazon EC2 Amazon-Instance-Typen basieren. Die von Ihnen ausgewählten Instance-Typen bestimmen die Host-Hardware, auf der Ihre Instances ausgeführt werden. EC2 Instanztypen können danach kategorisiert werden, auf welcher Prozessorarchitektur sie jeweils basieren. Elastic Beanstalk unterstützt Instance-Typen, die auf den folgenden Prozessorarchitekturen basieren: AWS Graviton 64-Bit-Arm-Architektur (arm64), 64-Bit-Architektur (x86) und 32-Bit-Architektur (i386). Elastic Beanstalk wählt standardmäßig die x86-Prozessorarchitektur aus, wenn Sie eine neue Umgebung erstellen.

> **Note**
>
> Die i386 32-Bit-Architektur wird von den meisten Elastic Beanstalk-Plattformen nicht mehr unterstützt. Wir empfehlen, stattdessen die Architekturtypen x86 oder arm64 auszuwählen. Elastic Beanstalk bietet Konfigurationsoptionen für i386-Prozessor-Instance-Typen im `aws:ec2:instances`-Namespace.

Alle Instance-Typen in der Konfiguration für eine bestimmte Elastic Beanstalk-Umgebung müssen dieselbe Art von Prozessorarchitektur haben. Angenommen, Sie fügen einer vorhandenen Umgebung einen neuen Instance-Typ hinzu, die bereits über einen t2.medium-Instance-Typ verfügt, der auf der x86-Architektur basiert. Sie können nur einen anderen Instance-Typ derselben Architektur hinzufügen, wie z. B. t2.small. Wenn Sie die vorhandenen Instance-Typen durch Instance-Typen aus

einer anderen Architektur ersetzen möchten, können Sie dies tun. Stellen Sie jedoch sicher, dass alle Instanztypen im Befehl auf demselben Architekturtyp basieren.

Elastic Beanstalk fügt regelmäßig Unterstützung für neue kompatible Instance-Typen hinzu, nachdem Amazon sie EC2 eingeführt hat. Informationen zu den verfügbaren Instance-Typen finden Sie unter Instance-Typen im EC2 Amazon-Benutzerhandbuch.

> **Note**
>
> Elastic Beanstalk bietet jetzt Unterstützung für Graviton auf allen aktuellen Amazon Linux 2-Plattformen in allen von AWS Graviton unterstützten Regionen. Weitere Informationen zum Erstellen einer Elastic Beanstalk-Umgebung mit arm64-basierten Instance-Typen finden Sie unter Konfiguration von EC2 Amazon-Instances mit der Elastic Beanstalk Beanstalk-Konsole. Erstellen Sie neue Umgebungen, in EC2 denen Amazon-Instances auf der Arm64-Architektur ausgeführt werden, und migrieren Sie Ihre vorhandenen Anwendungen mit den Bereitstellungsoptionen in Elastic Beanstalk darauf.
> Weitere Informationen zu den arm64-basierten Graviton-Prozessoren finden Sie in diesen Ressourcen: AWS
>
> - Vorteile — Der Graviton-Prozessor AWS
> - Erste Schritte und weitere Themen, z. B. sprachspezifische Überlegungen — Artikel Erste Schritte mit Graviton AWS GitHub

Konfiguration von EC2 Amazon-Instances mit der Elastic Beanstalk Beanstalk-Konsole

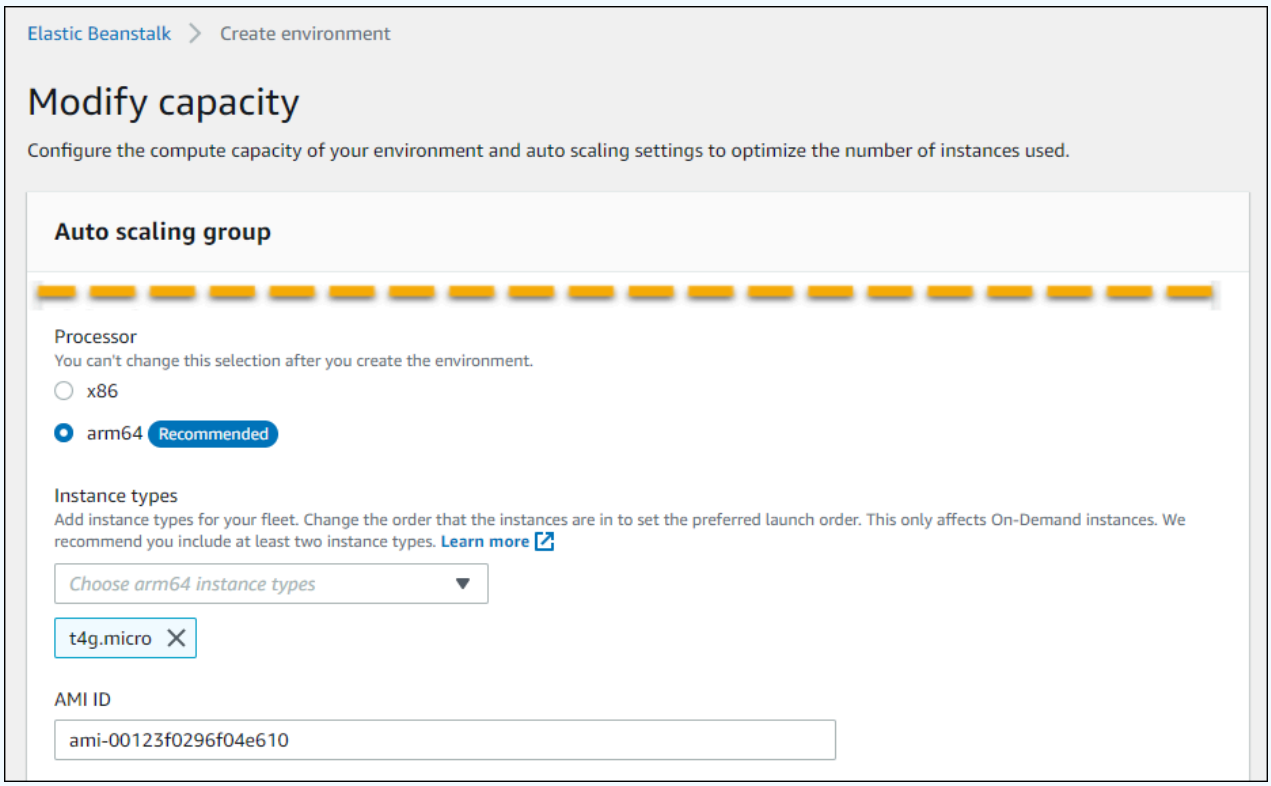
Sie können die EC2 Amazon-Instance-Konfiguration Ihrer Elastic Beanstalk Beanstalk-Umgebung in der Elastic Beanstalk Beanstalk-Konsole erstellen oder ändern.

> **Note**
>
> Die Elastic Beanstalk Beanstalk-Konsole bietet zwar nicht die Möglichkeit, die Prozessorarchitektur einer vorhandenen Umgebung zu ändern, Sie können dies jedoch mit der tun. AWS CLI Befehle finden Sie beispielsweise unter. Konfiguration von Amazon EC2-Sicherheitsgruppen und Instance-Typen mit dem AWS CLI

So konfigurieren Sie EC2 Amazon-Instances in der Elastic Beanstalk Beanstalk-Konsole während der Umgebungserstellung

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus.
3. Wählen Sie Create a new environment (Neue Umgebung erstellen), um mit der Erstellung Ihrer Umgebung zu beginnen.
4. Wählen Sie auf der Hauptseite des Assistenten zuerst Configure more options (Weitere Optionen konfigurieren) und dann Create environment (Umgebung erstellen) aus.
5. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten) aus. Nehmen Sie Änderungen an den Einstellungen in dieser Kategorie vor und wählen Sie dann Apply (Übernehmen). Informationen zum Einstellen von Beschreibungen finden Sie im Abschnitt the section called “Instance-Kategorieeinstellungen” auf dieser Seite.
6. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten). Nehmen Sie Änderungen an den Einstellungen in dieser Kategorie vor und wählen Sie dann Continue (Weiter). Informationen zur Einstellung von Beschreibungen finden Sie im Abschnitt the section called “Einstellungen der Kapazitätskategorien” auf dieser Seite.

> **Auswahl der Prozessorarchitektur**
>
> Scrollen Sie nach unten zu Prozessor, um eine Prozessorarchitektur für Ihre EC2 Instances auszuwählen. Die Konsole listet Prozessorarchitekturen auf, die von der Plattform unterstützt werden, die Sie zuvor im Bereich Create Environment (Umgebung erstellen) ausgewählt haben.
> Wenn die benötigte Prozessorarchitektur nicht angezeigt wird, kehren Sie zur Liste der Konfigurationskategorien zurück, um eine Plattform auszuwählen, die sie unterstützt. Wählen Sie in Modify Capacity (Kapazität ändern) die Option Abbrechen aus. Wählen Sie dann Change platform version (Ändern der Plattformversion), um neue Plattformeinstellungen auszuwählen. Anschließend wählen Sie in der Konfigurationskategorie Capacity (Kapazität) auswählen Edit (Bearbeiten) aus, um sich die Entscheidungen der Prozessorarchitektur noch einmal anzusehen.



7. Wählen Sie Save (Speichern) aus und nehmen Sie dann alle weiteren Konfigurationsänderungen für Ihre Umgebung vor.
8. Wählen Sie Create environment (Umgebung erstellen) aus.

So konfigurieren Sie die EC2 Amazon-Instances einer laufenden Umgebung in der Elastic Beanstalk Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten). Nehmen Sie Änderungen an den Einstellungen in dieser Kategorie vor und wählen Sie dann Apply (Übernehmen). Informationen zum Einstellen von Beschreibungen finden Sie im Abschnitt the section called “Instance-Kategorieeinstellungen” auf dieser Seite.

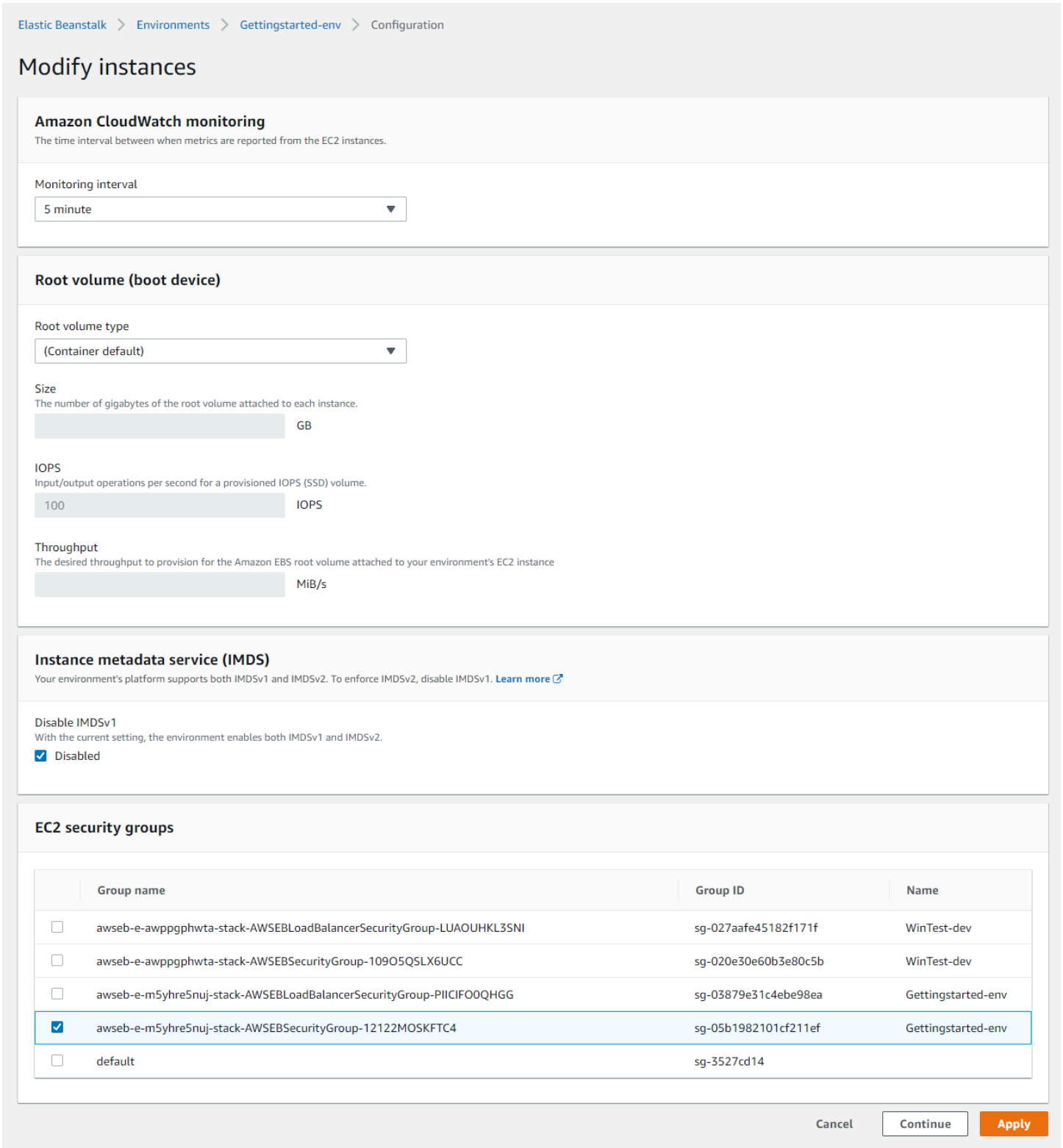
5. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten). Nehmen Sie Änderungen an den Einstellungen in dieser Kategorie vor und wählen Sie dann Continue (Weiter). Informationen zum Einstellen von Beschreibungen finden Sie im Abschnitt the section called “Einstellungen der Kapazitätskategorien” auf dieser Seite.

Instance-Kategorieeinstellungen

Die folgenden Einstellungen für EC2 Amazon-Instances sind in der Kategorie Instance-Konfiguration verfügbar.

Optionen

- Überwachungsintervall
- Root-Volume (Startgerät)
- Instance-Metadatenservice
- EC2 Sicherheitsgruppen


Elastic Beanstalk > Environments > Gettingstarted-env > Configuration
Modify instances
Amazon CloudWatch monitoring
The time interval between when metrics are reported from the EC2 instances.
Monitoring interval
5 minute
Root volume (boot device)
Root volume type
(Container default)
Size
The number of gigabytes of the root volume attached to each instance.
GB
IOPS
Input/output operations per second for a provisioned IOPS (SSD) volume.
100
IOPS
Throughput
The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance
MiB/s
Instance metadata service (IMDS)
Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, disable IMDSv1. Learn more
Disable IMDSv1
With the current setting, the environment enables both IMDSv1 and IMDSv2.
Disabled
EC2 security groups
Group name
Group ID
Name
awseb-e-awppgphwta-stack-AWSEBLoadBalancerSecurityGroup-LUAOUHKL3SNI
sg-027aafe45182f171f
WinTest-dev
awseb-e-awppgphwta-stack-AWSEBSecurityGroup-109O5QSLX6UCC
sg-020e30e60b3e80c5b
WinTest-dev
awseb-e-m5yhre5nuj-stack-AWSEBLoadBalancerSecurityGroup-PIICIFO0QHGG
sg-03879e31c4ebe98ea
Gettingstarted-env
awseb-e-m5yhre5nuj-stack-AWSEBSecurityGroup-12122MOSKFTC4
sg-05b1982101cf211ef
Gettingstarted-env
default
sg-3527cd14
Cancel
Continue
Apply

Überwachungsintervall

Standardmäßig veröffentlichen die Instances in Ihrer Umgebung grundlegende Integritätskennzahlen in Intervallen von fünf Minuten ohne zusätzliche Kosten CloudWatch an Amazon.

Für detailliertere Berichte können Sie das Überwachungsintervall auf 1 Minute festlegen, um die Häufigkeit zu erhöhen, mit der die Ressourcen in Ihrer Umgebung grundlegende Integritätskennzahlen veröffentlichen. CloudWatch CloudWatch Für Metriken im Abstand von einer Minute fallen Servicegebühren an. Weitere Informationen finden Sie auf Amazon CloudWatch.

Root-Volume (Startgerät)

Jede Instance der Umgebung ist mit einem Stamm-Volume konfiguriert. Das Stamm-Volume ist das Amazon EBS-Blockgerät, das der Instance zugeordnet ist und auf dem das Betriebssystem, Bibliotheken, Skripts und der Anwendungsquellcode gespeichert sind. Standardmäßig verwenden alle Plattformen allgemeine SSD-Blockgeräte für die Speicherung.

Sie können den Root volume type (Stamm-Volume-Typ) so ändern, dass Magnetspeicher oder bereitgestellte IOPS-SSD-Volumetypen verwendet werden und bei Bedarf die Volume-Größe erhöhen. Bei bereitgestellten IOPS-Volumes müssen Sie auch die Anzahl der bereitzustellenden IOPS angeben. Durchsatz ist nur für gp3 SSD-Volume-Typen anwendbar. Sie können den gewünschten Durchsatz für die Bereitstellung eingeben. Es kann zwischen 125 und 1000 Mebibyte pro Sekunde (MIB/s) liegen. Wählen Sie den Volume-Typ aus, der Ihren Leistungsanforderungen und Kostenvorstellungen entspricht.

> **Important**
>
> Die `RootVolumeType` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

Weitere Informationen finden Sie unter Amazon EBS-Volumetypen im EC2 Amazon-Benutzerhandbuch und in den Amazon EBS-Produktdetails.

Instance-Metadatenservice

Der Instance-Metadatenservice (IMDS) ist eine On-Instance-Komponente, die von Code auf der Instance verwendet wird, um sicher auf Instance-Metadaten zuzugreifen. Sie können mit einer der folgenden Methoden auf Instance-Metadaten aus einer laufenden Instance zugreifen. Sie sind Instance Metadata Service Version 1 (IMDSv1) oder Instance Metadata Service Version 2 (IMDSv2). IMDSv2 ist sicherer. Zum IMDSv1 Erzwingen deaktivieren IMDSv2. Weitere Informationen finden Sie unter the section called “IMDS”.

> **Note**
>
> Der IMDS-Abschnitt auf dieser Konfigurationsseite wird nur für Plattformversionen angezeigt, die dies unterstützen IMDSv2.

EC2 Sicherheitsgruppen

Die Sicherheitsgruppen, die Ihren Instances zugeordnet sind, bestimmen, welcher Traffic die Instances erreichen und verlassen darf.

Die von Elastic Beanstalk erstellte EC2 Standardsicherheitsgruppe lässt den gesamten eingehenden Datenverkehr aus dem Internet oder Load Balancern an den Standardports für HTTP (80) und SSH (22) zu. Sie können auch Ihre eigenen benutzerdefinierten Sicherheitsgruppen definieren, um Firewall-Regeln für die Instances festzulegen. EC2 Die Sicherheitsgruppen können Datenverkehr über andere Ports oder aus anderen Quellen zulassen. Beispielsweise können Sie eine Sicherheitsgruppe für den SSH-Zugriff erstellen, mit der eingehender Datenverkehr über Port 22 aus einem eingeschränkten IP-Adressbereich oder – für mehr Sicherheit – von einem Bastion-Host, auf den nur Sie zugreifen können, erlaubt ist. Für zusätzliche Sicherheit können Sie auch eine erstellen, die den Datenverkehr von einem Bastion-Host zulässt, auf den nur Sie zugreifen können.

Sie können wählen, ob Sie Ihre Umgebung von der EC2 Standardsicherheitsgruppe ausschließen möchten, indem Sie die `DisableDefaultEC2SecurityGroup` Option im aws:autoscaling:launchconfiguration Namespace auf setzen. `true` Diese Option ist in der Konsole nicht verfügbar, Sie können sie jedoch mit dem AWS CLI festlegen. Weitere Informationen finden Sie unter Verwaltung von EC2-Sicherheitsgruppen.

Weitere Informationen zu EC2 Amazon-Sicherheitsgruppen finden Sie unter Amazon EC2 Security Groups im EC2 Amazon-Benutzerhandbuch.

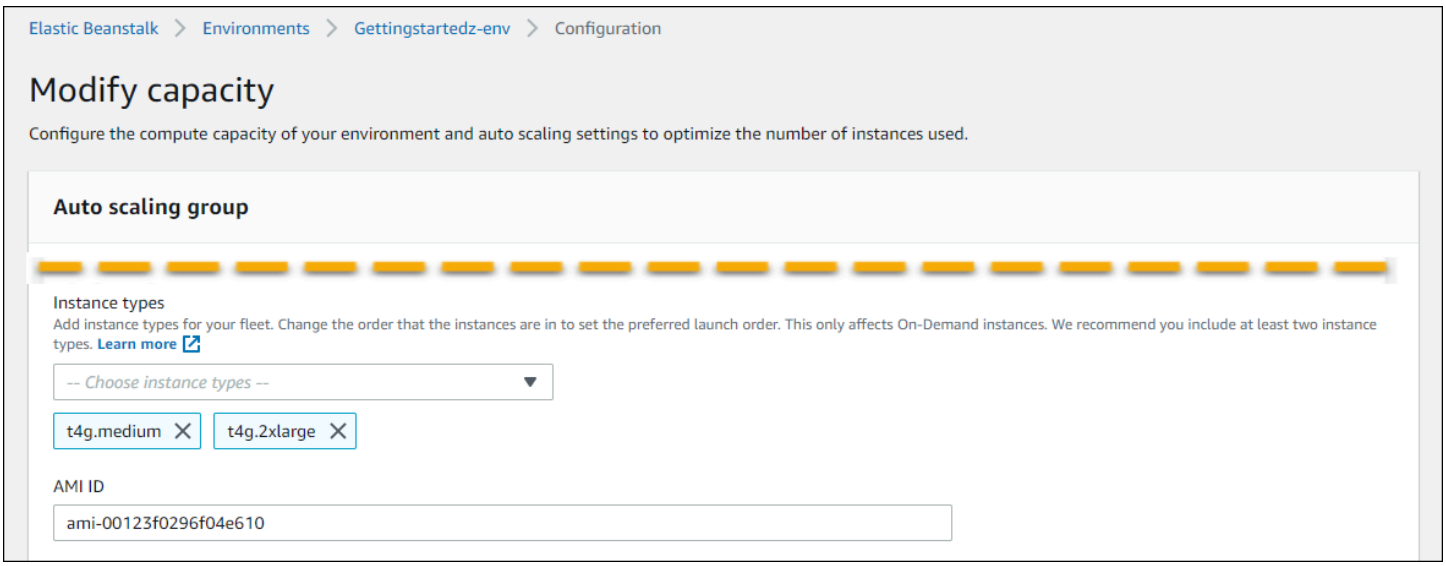
> **Note**
>
> Um den Datenverkehr zwischen Instances der Umgebung A und Instances der Umgebung B zu erlauben, können Sie eine Regel zur Sicherheitsgruppe hinzufügen, die Elastic Beanstalk an Umgebung B angefügt hat. Dann können Sie die Sicherheitsgruppe bestimmen, die Elastic Beanstalk an Umgebung A angefügt hat. Das erlaubt den eingehenden und ausgehenden Datenverkehr für Instances der Umgebung A. Beachten Sie jedoch, dass dadurch eine Abhängigkeit zwischen den beiden Sicherheitsgruppen entsteht. Wenn Sie später versuchen, Umgebung A zu beenden, wird Elastic Beanstalk die Sicherheitsgruppe der Umgebung nicht löschen können, da die Sicherheitsgruppe der Umgebung B von ihr abhängt.
> Daher wird empfohlen, stattdessen eine separate Sicherheitsgruppe zu erstellen. Fügen Sie sie dann der Umgebung A hinzu und geben Sie sie in einer Regel der Sicherheitsgruppe der Umgebung B an.

Einstellungen der Kapazitätskategorien

Die folgenden Einstellungen für EC2 Amazon-Instances sind in der Kategorie Kapazitätskonfiguration verfügbar.

Optionen

- Instance-Typen
- AMI-ID



Instance-Typen

Die Einstellung Instance-Typen bestimmt den Typ der EC2 Amazon-Instance, die zur Ausführung Ihrer Anwendung gestartet wird. Diese Konfigurationsseite zeigt eine Liste der Instance-Typen. Sie können einen oder mehrere Instance-Typen für Ihre Spot-Flotte auswählen. Die Elastic Beanstalk-Konsole zeigt nur die Instance-Typen an, die auf der Prozessorarchitektur basieren, die für Ihre Umgebung konfiguriert ist. Daher können Sie nur Instance-Typen derselben Prozessorarchitektur hinzufügen.

> **Note**
>
> Die Elastic Beanstalk Beanstalk-Konsole bietet zwar nicht die Möglichkeit, die Prozessorarchitektur einer vorhandenen Umgebung zu ändern, Sie können dies jedoch mit der tun. AWS CLI Befehle finden Sie beispielsweise unter. Konfiguration von Amazon EC2-Sicherheitsgruppen und Instance-Typen mit dem AWS CLI

Wählen Sie eine Instance aus, die einerseits leistungsfähig genug ist, um die Anwendung auch bei voller Auslastung auszuführen und die andererseits keine ungenutzte Überkapazität darstellt. Für Entwicklungszwecke bietet die t2-Instance-Familie generell eine moderate Leistung sowie die Möglichkeit einer Leistungssteigerung für kurze Zeiträume. Verwenden Sie für große Hochverfügbarkeitsanwendungen einen Instance-Pool, um sicherzustellen, dass die Kapazität nicht zu stark beeinträchtigt wird, wenn eine einzelne Instance ausfällt. Beginnen Sie mit einem Instance-Typ, mit dem Sie fünf Instances bei moderater Auslastung während des normalen Betriebs ausführen können. Falls eine Instance ausfällt, können die verbleibenden Instances den restlichen Datenverkehr verarbeiten. Der Kapazitätspuffer bietet zudem genügend Zeit, damit die Umgebung zur Bewältigung des steigenden Datenverkehr zu Spitzenzeiten entsprechend hochskalieren kann.

Weitere Informationen zu EC2 Amazon-Instance-Familien und -Typen finden Sie unter Instance-Typen im EC2 Amazon-Benutzerhandbuch. Informationen darüber, welche Instance-Typen Ihren Anforderungen entsprechen und welche Regionen unterstützt werden, finden Sie unter Verfügbare Instance-Typen im EC2 Amazon-Benutzerhandbuch.

AMI-ID

Das Amazon Machine Image (AMI) ist das Amazon Linux- oder Windows Server-Maschinen-Image, das Elastic Beanstalk verwendet, um EC2 Amazon-Instances in Ihrer Umgebung zu starten. Elastic Beanstalk stellt Maschinen-Images mit den für die Anwendungsausführung benötigten Tools und Ressourcen bereit.

Elastic Beanstalk wählt basierend auf der Region, der Plattform und dem Instance-Typ eine Standard-AMI für Ihre Umgebung aus. Wenn Sie ein benutzerdefiniertes AMI erstellt haben, ersetzen Sie die Standard-AMI-ID durch Ihre eigene AMI.

EC2-Sicherheitsgruppen verwalten

Wenn Elastic Beanstalk eine Umgebung erstellt, weist es den EC2-Instances, die damit gestartet werden, eine Standardsicherheitsgruppe zu. Die Sicherheitsgruppen, die Ihren Instances zugeordnet sind, bestimmen, welcher Traffic die Instances erreichen und verlassen darf.

Die von Elastic Beanstalk erstellte standardmäßige EC2-Sicherheitsgruppe lässt den gesamten eingehenden Datenverkehr aus dem Internet oder Load Balancern an den Standardports für HTTP (80) und SSH (22) zu. Sie können auch Ihre eigenen benutzerdefinierten Sicherheitsgruppen definieren, um Firewallregeln für die EC2-Instances festzulegen. Die Sicherheitsgruppen können Datenverkehr über andere Ports oder aus anderen Quellen zulassen. Beispielsweise können Sie eine Sicherheitsgruppe für den SSH-Zugriff erstellen, mit der eingehender Datenverkehr über Port 22 aus einem eingeschränkten IP-Adressbereich oder – für mehr Sicherheit – von einem Bastion-Host, auf den nur Sie zugreifen können, erlaubt ist. Für zusätzliche Sicherheit können Sie auch eine erstellen, die den Datenverkehr von einem Bastion-Host zulässt, auf den nur Sie zugreifen können.

Sie können wählen, ob Sie Ihre Umgebung von der standardmäßigen EC2-Sicherheitsgruppe ausschließen möchten, indem Sie die `DisableDefaultEC2SecurityGroup` Option im aws:autoscaling:launchconfiguration Namespace auf setzen. `true` Verwenden Sie die Konfigurationsdateien AWS CLIoder, um diese Option auf Ihre Umgebung anzuwenden und benutzerdefinierte Sicherheitsgruppen an die EC2-Instances anzuhängen.

Verwaltung von EC2-Sicherheitsgruppen in Umgebungen mit mehreren Instanzen

Wenn Sie eine benutzerdefinierte EC2-Sicherheitsgruppe in einer Umgebung mit mehreren Instanzen erstellen, müssen Sie auch berücksichtigen, wie die Load Balancer und die Regeln für eingehenden Datenverkehr dafür sorgen, dass Ihre Instances sicher und zugänglich sind.

Eingehender Datenverkehr in eine Umgebung mit mehreren EC2-Instances wird vom Load Balancer verwaltet, der den eingehenden Verkehr zwischen allen EC2-Instances leitet. Wenn Elastic Beanstalk eine standardmäßige EC2-Sicherheitsgruppe erstellt, definiert es auch Regeln für eingehenden Datenverkehr, die eingehenden Datenverkehr vom Load Balancer zulassen. Ohne diese Regel für eingehenden Datenverkehr in der Sicherheitsgruppe darf der eingehende Datenverkehr nicht in die Instances gelangen. Diese Bedingung würde die Instances im Wesentlichen vor externen Anfragen schützen.

Wenn Sie die standardmäßige EC2-Sicherheitsgruppe für eine Umgebung mit Lastenausgleich deaktivieren, validiert Elastic Beanstalk einige Konfigurationsregeln. Wenn die Konfiguration die Validierungsprüfungen nicht erfüllt, gibt sie Meldungen aus, in denen Sie aufgefordert werden, die erforderliche Konfiguration bereitzustellen. Die Validierungsprüfungen lauten wie folgt:

- Dem Load Balancer muss mit der `SecurityGroups` Option aws:elbv2:loadbalancer oder mindestens eine Sicherheitsgruppe zugewiesen werdenaws:elb:loadbalancer, je nachdem, ob es sich um einen Application Load Balancer oder einen Classic Load Balancer handelt. Beispiele AWS CLI finden Sie unter. Konfiguration mit dem AWS CLI
- Es müssen Regeln für eingehenden Verkehr existieren, die es Ihren EC2-Instances ermöglichen, Datenverkehr vom Load Balancer zu empfangen. Sowohl Ihre EC2-Sicherheitsgruppen als auch Ihre Load Balancer-Sicherheitsgruppen müssen auf diese Regeln für eingehenden Datenverkehr verweisen. Weitere Informationen finden Sie im nachfolgenden Regeln für eingehenden Datenverkehr-Abschnitt.

Regeln für eingehenden Datenverkehr

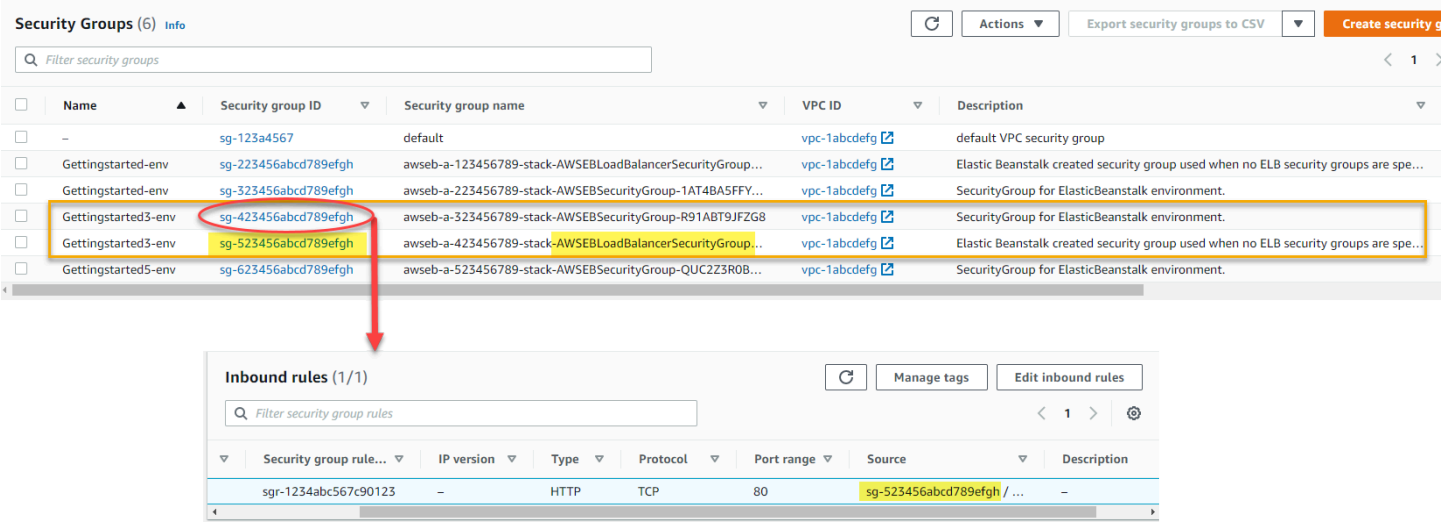
Die EC2-Sicherheitsgruppe (n) für eine Umgebung mit mehreren Instanzen müssen eine Regel für eingehenden Datenverkehr enthalten, die auf die Load Balancer-Sicherheitsgruppe verweist. Dies gilt für Umgebungen mit jeder Art von Load Balancer, ob dediziert oder gemeinsam genutzt, und mit benutzerdefinierten oder standardmäßigen Load Balancer-Sicherheitsgruppen.

Sie können alle Sicherheitsgruppen, die Ihren Umgebungskomponenten zugeordnet sind, in der EC2-Konsole anzeigen. Die folgende Abbildung zeigt die EC2-Konsole, in der die Sicherheitsgruppen aufgelistet sind, die Elastic Beanstalk standardmäßig während des Vorgangs „Umgebung erstellen“ erstellt.

Der Bildschirm „Sicherheitsgruppen“ zeigt Umgebungen und die zugehörigen Sicherheitsgruppen. Sowohl GettingStarted-env als auch GettingStarted3-env sind Umgebungen mit mehreren Instanzen und dedizierten Load Balancern. In jeder dieser Umgebungen sind zwei Sicherheitsgruppen aufgeführt, eine für die EC2-Instances und eine weitere für den Load Balancer. Elastic Beanstalk erstellt diese Sicherheitsgruppen bei der Erstellung der Umgebungen. GettingStarted5-env hat keine Load Balancer-Sicherheitsgruppe, da es nur eine EC2-Instance und somit keinen Load Balancer hat.

Im Fenster Inbound Rules wird die EC2-Sicherheitsgruppe für die Instances von 3-env detailliert beschrieben. GettingStarted In diesem Beispiel werden die Regeln für eingehende Nachrichten für die EC2-Sicherheitsgruppe definiert. Beachten Sie, dass in der Spalte Quelle der Regeln für eingehende Nachrichten die Sicherheitsgruppen-ID der Load Balancer-Sicherheitsgruppe

aufgeführt ist, die im vorherigen Bild aufgeführt ist. Diese Regel ermöglicht es den EC2-Instances von GettingStarted3-env, eingehenden Datenverkehr von diesem speziellen Load Balancer an Port 80 zu empfangen.



Weitere Informationen finden Sie unter Sicherheitsgruppen für Ihre Instance ändern und Elastic Load Balancing Balancing-Regeln im Amazon EC2 EC2-Benutzerhandbuch.

Konfiguration von Amazon EC2-Sicherheitsgruppen und Instance-Typen mit dem AWS CLI

Sie können die AWS Befehlszeilenschnittstelle (AWS CLI) verwenden, um die Amazon EC2 EC2-Instances in Ihren Elastic Beanstalk Beanstalk-Umgebungen zu konfigurieren.

Konfiguration von EC2-Sicherheitsgruppen mit dem AWS CLI

Dieses Thema enthält Beispiele für verschiedene EC2-Sicherheitsgruppenkonfigurationen sowohl für Einzelinstanzen als auch für Umgebungen mit Lastenausgleich (mehrere Instanzen). Weitere Informationen zu den Optionen in diesen Beispielen finden Sie unter aws:autoscaling:launchconfiguration.

> **Hinweise**
>
> Der Vorgang „Umgebung erstellen“ stellt standardmäßig eine EC2-Sicherheitsgruppe bereit. Außerdem wird standardmäßig eine Umgebung mit einem Application Load Balancer erstellt. Der Vorgang zur Aktualisierung der Umgebung kann verwendet werden, um die standardmäßige EC2-Sicherheitsgruppe für Ihre Umgebung mit der booleschen Option

entweder zu deaktivieren oder zu aktivieren. `DisableDefaultEC2SecurityGroup` Beispiel 5 zeigt, wie Sie Ihre Umgebung auf die Standardsicherheitskonfiguration zurücksetzen, wenn Sie sie zuvor geändert haben.

Die folgenden Beispiele zeigen einen Befehl create-environment, der die standardmäßige EC2-Sicherheitsgruppe deaktiviert und stattdessen benutzerdefinierte Sicherheitsgruppen bereitstellt. Da die `DisableDefaultEC2SecurityGroup` Option auf gesetzt ist`true`, wird die standardmäßige EC2-Sicherheitsgruppe, die Elastic Beanstalk normalerweise den EC2-Instances zuordnet, nicht erstellt. Daher müssen Sie anderen Sicherheitsgruppen diese Option zur Verfügung stellen. `SecurityGroups`

Beachten Sie, dass die aws:elasticbeanstalk:environment `EnvironmentType` Option auf gesetzt ist`SingleInstance`. Um eine Einzelinstanzumgebung zu erstellen, müssen Sie diese Option angeben, da dies die Standardoption `LoadBalanced` ist`EnvironmentType`. Da diese Umgebung keinen Load Balancer enthält, müssen wir keine Load Balancer-Sicherheitsgruppe angeben.

Example 1 — Neue Einzelinstanzumgebung mit benutzerdefinierten EC2-Sicherheitsgruppen (Inline-Namespace-Optionen)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 applrunning Node.js 22" \
--option-settings \
Namespace=aws:elasticbeanstalk:environment,OptionName=EnvironmentType,Value=SingleInstance
 \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-
elasticbeanstalk-ec2-role \
Namespace=aws:autoscaling:launchconfiguration,OptionName=DisableDefaultEC2SecurityGroup,Value=t
 \
Namespace=aws:autoscaling:launchconfiguration,OptionName=SecurityGroups,Value=sg-
abcdef01, sg-abcdef02 \
Namespace=aws:autoscaling:launchconfiguration,OptionName=EC2KeyName,Value=my-keypair
```

Verwenden Sie alternativ eine`options.json`-Datei, um die Namespace-Optionen anzugeben, anstatt sie einzubinden.

Example 2 — Neue Einzelinstanzumgebung mit benutzerdefinierten EC2-Sicherheitsgruppen (Namespace-Optionen in der Datei) **`options.json`**

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 running Node.js 22" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  { "Namespace" : "aws:elasticbeanstalk:environment",
    "OptionName" : "EnvironmentType",
    "Value" : "SingleInstance"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "EC2KeyName",
    "Value": "my-keypair"
  }
]
```

Das folgende Beispiel erstellt eine Umgebung mit Lastenausgleich. Es gibt die aws:elasticbeanstalk:environment Namespace-Option an, die `LoadBalancerType` auf

gesetzt ist. `application` Da wir die standardmäßige EC2-Sicherheitsgruppe mit dieser `DisableDefaultEC2SecurityGroup` Option deaktivieren, müssen wir wieder unsere eigenen benutzerdefinierten Sicherheitsgruppen für die EC2-Instances bereitstellen, und zwar mit der aws:autoscaling:launchconfiguration `SecurityGroups` Option, wie im vorherigen Beispiel. Da diese Umgebung über einen Load Balancer zur Weiterleitung des Datenverkehrs verfügt, müssen wir auch Sicherheitsgruppen für den Load Balancer bereitstellen.

Um eine Umgebung mit einem klassischen Load Balancer, aber ansonsten derselben Konfiguration zu erstellen, aktualisieren Sie die Konfiguration für die aws:elasticbeanstalk:environment Namespace-Option auf. `LoadBalancerType` `classic`

Die verschiedenen Load Balancer-Typen haben unterschiedliche Namespaces, die die Optionen zur Angabe der Sicherheitsgruppen enthalten:

- Load Balancer für Anwendungen — Option aws:elbv2:loadbalancer `SecurityGroups`
- klassischer Load Balancer — Option aws:elb:loadbalancer `SecurityGroups`
- Network Load Balancer — da Network Load Balancer keine Sicherheitsgruppen haben, konfigurieren Sie die EC2-Sicherheitsgruppen mit VPC-Identifikatoren. Weitere Informationen finden Sie unter Aktualisieren der Sicherheitsgruppen für Ihren Network Load Balancer im Benutzerhandbuch für Network Load Balancer.

Example 3 — Neue Multi-Instanzen-Umgebung mit benutzerdefinierten EC2-Sicherheitsgruppen (Namespace-Optionen in der Datei) **`options.json`**

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 running Node.js 22" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace" : "aws:elasticbeanstalk:environment",
    "OptionName" : "EnvironmentType",
    "Value" : "LoadBalanced"
```

```
  },
  {
  "Namespace" : "aws:elasticbeanstalk:environment",
    "OptionName" : "LoadBalancerType",
    "Value" : "application"
  },
  {
    "Namespace" : "aws:elbv2:loadbalancer",
    "OptionName" : "SecurityGroups",
    "Value" : "sg-abcdefghikl012345"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "EC2KeyName",
    "Value": "my-keypair"
  }
]
```

Sie können die standardmäßige EC2-Sicherheitsgruppe für eine bestehende Umgebung mit dem Befehl update-environment deaktivieren. Der folgende Beispielbefehl deaktiviert die standardmäßige EC2-Sicherheitsgruppe und weist den EC2-Instances der Umgebung benutzerdefinierte EC2-Sicherheitsgruppen zu.

Verwenden Sie die `options.jason` Beispieldateien in den Beispielen 4 (a), 4 (b) oder 4 (c), je nachdem, ob die Umgebung über einen Lastenausgleich verfügt und welche Art von Load Balancer verwendet wird. Die Konfigurationsdatei 4 (a) spezifiziert die Sicherheitsgruppen für eine Einzelinstanzumgebung. Da kein Load Balancer erforderlich ist, stellen wir nur die

Sicherheitsgruppe für die EC2-Instances bereit. Die Konfigurationsdateien 4 (b) und 4 (c) spezifizieren die Sicherheitsgruppen für einen Application Load Balancer und einen klassischen Load Balancer. In diesen Fällen müssen wir auch Sicherheitsgruppen für den Load Balancer angeben.

Example 4 — Aktualisieren Sie eine bestehende Umgebung, um die standardmäßige EC2-Sicherheitsgruppe zu deaktivieren (Namespace-Optionen in der Datei) **`options.json`**

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 running Node.js 22" \
--option-settings file://options.json
```

Example 4 (a) — Konfigurationsdatei für eine Einzelinstanzumgebung (kein Load Balancer)

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  }
]
```

Um eine Umgebung zu aktualisieren, die einen Application Load Balancer verwendet, verwenden Sie den `aws:elbv2:loadbalancer` Namespace, um die Sicherheitsgruppen für den Load Balancer anzugeben.

Example 4 (b) — Konfigurationsdatei für eine Umgebung mit einem Application Load Balancer

```
### example options.json ###
[
  {
    "Namespace" : "aws:elbv2:loadbalancer",
    "OptionName" : "SecurityGroups",
```

```
    "Value" : "sg-abcdefghikl012345"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  }
]
```

Um eine Umgebung zu aktualisieren, die einen klassischen Load Balancer verwendet, verwenden Sie den `aws:elb:loadbalancer` Namespace, um die Sicherheitsgruppen für den Load Balancer anzugeben.

Example 4 (c) — Konfigurationsdatei für eine Umgebung mit einem klassischen Load Balancer

```
### example options.json ###
[
  {
    "Namespace" : "aws:elb:loadbalancer",
    "OptionName" : "SecurityGroups",
    "Value" : "sg-abcdefghikl012345"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",n
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  }
]
```

Um Ihre Umgebung auf das Standardverhalten und die Standardkonfiguration mit der von Elastic Beanstalk zugewiesenen Standardsicherheitsgruppe zurückzusetzen, verwenden Sie den Befehl update-environment, um den Wert auf zu setzen. `DisableDefaultEC2SecurityGroup` `false` In

einer Umgebung mit mehreren Instanzen verwaltet Elastic Beanstalk auch die Sicherheitsgruppen und Netzwerkverkehrsregeln für den Load Balancer Ihrer Umgebung.

Das folgende Beispiel gilt sowohl für eine Einzelinstanz- als auch für eine Multi-Instance-Umgebung (Load Balancing):

Example 5 — Aktualisieren Sie eine Umgebung, sodass sie wieder die Standardsicherheitsgruppe verwendet (Namespace-Optionen in der Datei) **`options.json`**

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 running Node.js 22" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "false"
  }
]
```

Konfiguration von EC2 mit Instance-Typen unter Verwendung von AWS CLI

Dieses Thema enthält Beispiele für die Konfiguration der Instance-Typen der EC2-Instances in Ihrer Umgebung.

Die ersten beiden Beispiele erstellen eine neue Umgebung. Der Befehl gibt einen Amazon EC2-Instance-Typ, t4g.small, der auf der arm64-Prozessorarchitektur basiert, an. Elastic Beanstalk verwendet die Image ID (AMI) für die EC2-Instances basierend auf Region, Plattformversion und Instance-Typ standardmäßig. Der Instance-Typ entspricht einer Prozessorarchitektur. Der `solution-stack-name`-Parameter gilt für die Plattformversion.

Example 1 — Erstellen Sie eine neue arm64-basierte Umgebung (eingebundene Namespace-Optionen)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small
```

Verwenden Sie alternativ eine`options.json`-Datei, um die Namespace-Optionen anzugeben, anstatt sie einzubinden.

Example 2 — Erstellen Sie eine neue arm64-basierte Umgebung (Namespace-Optionen in **`options.json`**-Datei)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:ec2:instances",
    "OptionName": "InstanceTypes",
    "Value": "t4g.small"
  }
]
```

Die nächsten beiden Beispiele aktualisieren die Konfiguration für eine vorhandene Umgebung mit dem Befehl update-environment. In diesem Beispiel fügen wir einen weiteren Instance-Typ hinzu, der auch auf der arm64-Prozessorarchitektur basiert. Für vorhandene Umgebungen müssen alle hinzugefügten Instance-Typen dieselbe Prozessorarchitektur haben. Wenn Sie die vorhandenen Instance-Typen durch Instance-Typen aus einer anderen Architektur ersetzen möchten, können Sie dies tun. Stellen Sie jedoch sicher, dass alle Instanztypen im Befehl denselben Architekturtyp haben.

Example 3 — aktualisiert eine vorhandene arm64-basierte Umgebung (eingebundene Namespace-Optionen)

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small,t4g.micro
```

Verwenden Sie alternativ eine`options.json`-Datei, um die Namespace-Optionen anzugeben, anstatt sie einzubinden.

Example 4 — aktualisiert eine vorhandene arm64-basierte Umgebung (Namespace-Optionen in **`options.json`**-Datei)

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
```

```
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:ec2:instances",
    "OptionName": "InstanceTypes",
    "Value": "t4g.small, t4g.micro"
  }
]
```

Die nächsten beiden Beispiele zeigen weitere Befehle create-environment. Diese Beispiele enthalten keine Werte für `InstanceTypes`. Wann`InstanceTypes`-Werte werden nicht angegeben, Elastic Beanstalk verwendet standardmäßige die x86-basierte Prozessorarchitektur. Die Image-ID (AMI) für die EC2-Instances der Umgebung wird gemäß der Region, der Plattformversion und dem Standard-Instance-Typ standardmäßig ausgeführt. Der Instance-Typ entspricht einer Prozessorarchitektur.

Example 5 — Erstellen Sie eine neue x86-basierte Umgebung (eingebundene Namespace-Optionen)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role
```

Verwenden Sie alternativ eine`options.json`-Datei, um die Namespace-Optionen anzugeben, anstatt sie einzubinden.

Example 6 — Erstellen Sie eine neue x86-basierte Umgebung (Namespace-Optionen in **`options.json`**-Datei)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  }
]
```

Konfiguration von EC2 Amazon-Instances mit Namespace-Optionen

Sie können die Konfigurationsoptionen im `aws:autoscaling:launchconfiguration`-Namespace zur Konfiguration Ihrer Umgebungs-Instances verwenden, einschließlich zusätzlicher Optionen, die in der Konsole nicht verfügbar sind.

> **Important**
>
> Die `BlockDeviceMappings` Optionseinstellung `DisableIMDSv1RootVolumeType`, oder kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

In der folgenden Beispieldatei werden die grundlegenden Konfigurationsoptionen verwendet, die in diesem Thema erläutert werden. Beispiele für zusätzliche Konfigurationsoptionen, wenn Sie Sicherheitsgruppen für Load Balancer angeben müssen, finden Sie unterKonfiguration mit dem AWS CLI.

```
option_settings:
  aws:autoscaling:launchconfiguration:
    SecurityGroups: my-securitygroup
    MonitoringInterval: "1 minute"
```

```
    DisableIMDSv1: false
    DisableDefaultEC2SecurityGroup: true
    SecurityGroups: "sg-abcdef01, sg-abcdef02"
    EC2KeyName: my-keypair
    IamInstanceProfile: "aws-elasticbeanstalk-ec2-role"
    BlockDeviceMappings: "/dev/sdj=:100,/dev/sdh=snap-51eef269,/dev/sdb=ephemeral0"
  aws:elasticbeanstalk:environment:
    EnvironmentType: SingleInstance
```

Die `DisableDefaultEC2SecurityGroup` und `BlockDeviceMappings` sind in der Konsole nicht verfügbar.

Mit `BlockDeviceMappings` können Sie zusätzliche Blockgeräte für Ihre Instances konfigurieren. Weitere Informationen finden Sie unter Block Device Mapping im EC2 Amazon-Benutzerhandbuch.

Die EB-CLI und Elastic-Beanstalk-Konsole wenden die für die vorhergehenden Optionen empfohlenen Werte an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter Empfohlene Werte.

Konfiguration des IMDS auf den Instances Ihrer Elastic Beanstalk Beanstalk-Umgebung

In diesem Thema wird der Instance Metadata Service (IMDS) beschrieben.

Instance-Metadaten sind Daten, die sich auf eine Amazon Elastic Compute Cloud (Amazon EC2) - Instance beziehen und die Anwendungen zur Konfiguration oder Verwaltung der laufenden Instance verwenden können. Der Instance-Metadatenservice (IMDS) ist eine On-Instance-Komponente, die von Code auf der Instance verwendet wird, um sicher auf Instance-Metadaten zuzugreifen. Dieser Code kann Elastic Beanstalk-Plattformcode auf Ihren Umgebungsinstanzen, das AWS SDK, das Ihre Anwendung möglicherweise verwendet, oder sogar der eigene Code Ihrer Anwendung sein. Weitere Informationen finden Sie unter Instance-Metadaten und Benutzerdaten im EC2 Amazon-Benutzerhandbuch.

Code kann mit einer von zwei Methoden auf Instance-Metadaten von einer laufenden Instance zugreifen: Instance Metadata Service Version 1 (IMDSv1) oder Instance Metadata Service Version 2 (IMDSv2). IMDSv2 verwendet sitzungsorientierte Anfragen und behebt verschiedene Arten von Sicherheitslücken, die für den Zugriff auf das IMDS genutzt werden könnten. Informationen zu diesen beiden Methoden finden Sie unter Konfiguration des Instance-Metadatendienstes im EC2 Amazon-Benutzerhandbuch.

Themen

- Plattformunterstützung für IMDS
- Auswahl der IMDS-Methoden
- Konfigurieren von IMDS mit der Elastic Beanstalk-Konsole
- Der Namespace aws:autoscaling:launchconfiguration

Plattformunterstützung für IMDS

Elastic Beanstalk-Plattformen, die auf Amazon Linux 2 und Amazon Linux 2023 laufen, und Windows Server unterstützen alle sowohl als auch IMDSv1 . IMDSv2 Weitere Informationen finden Sie unter Konfigurieren von IMDS mit der Elastic Beanstalk-Konsole.

Auswahl der IMDS-Methoden

Berücksichtigen Sie bei der Entscheidung über die IMDS-Methoden, die von Ihrer Umgebung unterstützt werden sollen, folgende Anwendungsfälle:

- AWS SDK — Wenn Ihre Anwendung ein AWS SDK verwendet, stellen Sie sicher, dass Sie die neueste Version des SDK verwenden. AWS SDKsSie führen IMDS-Aufrufe durch und verwenden, IMDSv2 wann immer möglich, neuere SDK-Versionen. Wenn Sie sie jemals deaktivieren IMDSv1 oder wenn Ihre Anwendung eine alte SDK-Version verwendet, schlagen IMDS-Aufrufe möglicherweise fehl.
- Ihr Anwendungscode — Wenn Ihre Anwendung IMDS-Aufrufe durchführt, sollten Sie erwägen, das AWS SDK zu verwenden, damit Sie die Aufrufe tätigen können, anstatt direkte HTTP-Anfragen zu stellen. Auf diese Weise müssen Sie keine Codeänderungen vornehmen, um zwischen IMDS-Methoden zu wechseln. Das AWS SDK verwendet IMDSv2 wann immer möglich.
- Elastic Beanstalk-Plattformcode — Unser Code führt IMDS-Aufrufe über das AWS SDK durch und wird daher IMDSv2 auf allen unterstützenden Plattformversionen verwendet. Wenn Ihr Code ein up-to-date AWS SDK verwendet und alle IMDS-Aufrufe über das SDK tätigt, können Sie es problemlos deaktivieren. IMDSv1

Konfigurieren von IMDS mit der Elastic Beanstalk-Konsole

Sie können die EC2 Amazon-Instance-Konfiguration Ihrer Elastic Beanstalk Beanstalk-Umgebung in der Elastic Beanstalk Beanstalk-Konsole ändern.

> **Important**
>
> Die `DisableIMDSv1` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

So konfigurieren Sie IMDS auf Ihren EC2 Amazon-Instances in der Elastic Beanstalk Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Instance traffic and scaling (Instance, Datenverkehr und Skalierung) die Option Edit (Bearbeiten) aus.
5. Stellen Sie Deaktivieren ein, um es IMDSv1 zu erzwingen. IMDSv2 Deaktivieren Sie Deaktivieren IMDSv1, um sowohl als auch IMDSv1 zu aktivieren IMDSv2.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Der Namespace aws:autoscaling:launchconfiguration

Sie können eine Konfigurationsoption im `aws:autoscaling:launchconfiguration`-Namespace verwenden, um IMDS für die Instances Ihrer Umgebung zu konfigurieren.

> **Important**
>
> Die `DisableIMDSv1` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu

sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

Im folgenden Beispiel für eine Konfigurationsdatei wird die IMDSv1 Verwendung der `DisableIMDSv1` Option deaktiviert.

```
option_settings:
  aws:autoscaling:launchconfiguration:
    DisableIMDSv1: true
```

Stellen Sie Disable IMDSv1 auf ein, um `true` es zu deaktivieren IMDSv1 und zu erzwingen IMDSv2.

Stellen Sie Disable IMDSv1 auf ein`false`, um IMDSv1 sowohl als auch zu aktivieren IMDSv2.

Auto Scaling Ihrer Elastic Beanstalk Beanstalk-Umgebungsinstanzen

In diesem Thema wird beschrieben, wie Sie die Auto Scaling Scaling-Funktionen anpassen können, um die Arbeitslast Ihrer Elastic Beanstalk Beanstalk-Umgebung zu verwalten. Sie können Auto Scaling für Ihre Umgebung mit der Elastic Beanstalk Beanstalk-Konsole, den Namespace-Konfigurationsoptionen AWS CLI, der oder der EB CLI konfigurieren.

Umgebungen mit Lastenausgleich oder Einzelinstanzen

Ihre AWS Elastic Beanstalk Umgebung umfasst eine Auto Scaling Scaling-Gruppe, die die Amazon EC2 EC2-Instances in Ihrer Umgebung verwaltet. In einer Umgebung mit nur einer Instance stellt die Auto Scaling-Gruppe sicher, dass immer eine Instance ausgeführt wird. In einer Umgebung mit Lastausgleich konfigurieren Sie die Gruppe mit einem auszuführenden Instance-Bereich, und Auto Scaling fügt Instances je nach Auslastung hinzu bzw. entfernt sie.

EC2-Instance-Konfiguration

Die Auto Scaling Scaling-Gruppe wendet auch Ihre Konfigurationsoptionen an, um die EC2-Instances in Ihrer Umgebung bereitzustellen und zu verwalten. Sie können die EC2-Konfiguration ändern,

um den Instance-Typ, das key pair, den Amazon Elastic Block Store (Amazon EBS) -Speicher und andere Einstellungen zu ändern, die nur beim Starten einer Instance konfiguriert werden können.

On-Demand- und Spot-Instances

Optional kann Elastic Beanstalk Spot-Instances in Ihre Umgebung aufnehmen und diese in Kombination mit On-Demand-Instances verwalten. Sie können Amazon EC2 Auto-Scaling zur Überwachung von Veränderungen und zur automatisierten Reaktion auf Änderungen, die sich auf die Verfügbarkeit Ihrer Spot-Instances auswirken, konfigurieren, indem Sie Capacity Rebalancing (Kapazitäten neu ausgleichen) aktivieren. Sie können auch die konfigurierenStrategie für die Spot-Zuweisung, die der Auto Scaling Scaling-Dienst verwendet, um Spot-Instances für Ihre Umgebung bereitzustellen.

Erforderliche Berechtigungen bei der Aktivierung von Spot-Instances

Das Aktivieren von Spot-Instance-Anforderungen erfordert die Verwendung von Amazon-EC2-Startvorlagen. Wenn Sie diese Funktion während der Erstellung oder Aktualisierung der Umgebung konfigurieren, versucht Elastic Beanstalk, Ihre Umgebung so zu konfigurieren, dass Amazon-EC2-Startvorlagen verwendet werden (sofern die Umgebung sie nicht bereits verwendet). In diesem Fall schlägt die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Ihre Benutzerrichtlinie nicht die erforderlichen Berechtigungen besitzt. Daher empfehlen wir, unsere Richtlinie für verwaltete Benutzer zu verwenden oder die erforderlichen Berechtigungen zu Ihren benutzerdefinierten Richtlinien hinzuzufügen. Weitere Details zu den erforderlichen Berechtigungen finden Sie unter Erforderliche Berechtigungen für -Startvorlagen.

Auto-Scaling-Auslöser

Die Auto Scaling Scaling-Gruppe verwendet zwei CloudWatch Amazon-Alarme, um Skalierungsvorgänge auszulösen. Mit den Standardauslösern wird die Skalierung durchgeführt, wenn der durchschnittliche ausgehende Netzwerkdatenverkehr einer Instance innerhalb von 5 Minuten 6 MiB über- oder 2 MiB unterschreitet. Um Auto Scaling effektiv nutzen zu können, müssen Sie Auslöser konfigurieren, die für Ihre Anwendung, Ihren Instance-Typ und die Serviceanforderungen geeignet sind. Sie können die Skalierung anhand unterschiedlicher Statistiken wie Latenz, Festplatten-I/O-Vorgänge, CPU-Auslastung und Anforderungsanzahl ausführen.

Auto Scaling Scaling-Aktionen planen

Damit die Amazon EC2-Instances Ihrer Umgebung während erwarteter Spitzenlasten beim Datenverkehr optimal genutzt werden, sollten Sie Ihre Auto Scaling-Gruppe so konfigurieren, dass

die Anzahl der Instances nach einem festen Plan angepasst wird. Sie können Änderungen an der Konfiguration der Gruppe auf Tags- oder Wochenbasis planen oder einmalige Änderungen planen, um Marketing-Ereignisse abzufangen, durch die viel Datenverkehr auf Ihre Website ausgelöst wird.

Auto Scaling Scaling-Zustandsprüfung

Außerdem wird mit Auto Scaling auch der Zustand der einzelnen gestarteten Amazon EC2-Instances überwacht. Falls eine Instance unerwartet beendet wird, erkennt Auto Scaling dies und startet eine andere Instance als Ersatz. Weitere Informationen dazu, wie Sie die Gruppe so konfigurieren, dass die Zustandsprüfung von Load Balancer verwendet wird, finden Sie unter Einstellung für die Auto Scaling Scaling-Integritätsprüfung für Ihre Elastic Beanstalk Beanstalk-Umgebung.

Themen

- Migrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates
- Spot-Instance-Unterstützung für Ihre Elastic Beanstalk Beanstalk-Umgebung
- Auto Scaling-Trigger für Ihre Elastic Beanstalk Beanstalk-Umgebung
- Geplante Auto Scaling Scaling-Aktionen für Ihre Elastic Beanstalk Beanstalk-Umgebungen
- Einstellung für die Auto Scaling Scaling-Integritätsprüfung für Ihre Elastic Beanstalk Beanstalk-Umgebung

Migrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates

Ab dem 1. Oktober 2024 unterstützt Amazon EC2 Auto Scaling keine Startkonfigurationen mehr für neue Konten. Konten, die vor diesem Datum erstellt wurden, verfügen möglicherweise über Startkonfigurationen.

Wir empfehlen, zu Startvorlagen zu migrieren, um die folgenden Vorteile zu erzielen:

- Verbesserte Verfügbarkeit Ihrer Anwendungen
- Bessere Optimierung der Workloads in Ihren Auto Scaling Scaling-Gruppen
- Zugriff auf die neuesten EC2- und Auto Scaling Scaling-Funktionen

Weitere Informationen finden Sie unter Auto Scaling Scaling-Startkonfigurationen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Optionseinstellungen für Startvorlagen

Um Ihre Umgebung von Startkonfigurationen zu Startvorlagen zu migrieren, legen Sie eine der folgenden Konfigurationsoptionen fest:

- `RootVolumeType`Option auf gp3 gesetzt. Sie können diese Option mit der Konsole oder dem Namespace setzen.
- `BlockDeviceMappings`Die Option enthält gp3. Sie können diese Option mit der Konsole oder dem Namespace festlegen.
- `DisableIMDSv1`Option auf true gesetzt. Wir empfehlen, dass Sie diese Option mithilfe des Namespace festlegen.
- `EnableSpot`Option auf true gesetzt. Weitere Informationen finden Sie unter Spot-Instances aktivieren.

> **Important**
>
> Wenn eine Umgebung anfängt, Startvorlagen zu verwenden, kehrt Elastic Beanstalk nicht zu Startkonfigurationen zurück, selbst wenn Sie die Konfigurationsoptionen entfernen, die ursprünglich die Verwendung von Startvorlagen ausgelöst haben.

Prüfen Sie, ob Ihre Umgebung über Startkonfigurationen oder Startvorlagen verfügt

Sie können anhand der CloudFormation Stack-Vorlage überprüfen, ob Ihre Umgebung bereits Startvorlagen verwendet oder ob Startkonfigurationen verwendet werden.

Um die CloudFormation Stack-Vorlage Ihrer Umgebung zu überprüfen

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die AWS Region aus, in der Sie die Umgebung erstellt haben.
3. Sehen Sie sich auf der Stacks-Seite der CloudFormation Konsole die Spalte Beschreibung an.

   Suchen Sie den Stack für die Elastic Beanstalk Beanstalk-Umgebung und wählen Sie ihn aus. CloudFormation zeigt die Stack-Details für die Umgebung an.
4. Wählen Sie unter Stack-Details die Registerkarte Vorlage aus.

Mithilfe der Seitensuche Ihres Browsers können Sie den Vorlagentext nach launchtemplate oder launchconfiguration durchsuchen.

Weitere Informationen finden Sie im Benutzerhandbuch unter Stack-Informationen anzeigen.AWS CloudFormation

Erforderliche Berechtigungen für -Startvorlagen

Die standardmäßige Elastic Beanstalk Managed Service-Rollenrichtlinie AWSElasticBeanstalkManagedUpdatesCustomerRolePolicybietet die erforderlichen Berechtigungen zum Erstellen und Verwalten von Startvorlagen. Elastic Beanstalk muss Startvorlagen verwalten, um viele Umgebungsvorgänge, einschließlich der Erstellung von Umgebungen, abzuschließen.

Wenn Sie benutzerdefinierte Richtlinien an eine Elastic Beanstalk-Servicerolle anhängen, stellen Sie sicher, dass die Servicerolle die folgenden Berechtigungen für die Erstellung von Startvorlagen umfasst. Diese Berechtigungen ermöglichen es Elastic Beanstalk, erfolgreich Umgebungen in Ihrem Konto zu erstellen und zu aktualisieren:

Erforderliche Berechtigungen für Amazon EC2-Startvorlagen

- `ec2:RunInstances`
- `ec2:CreateLaunchTemplate`
- `ec2:CreateLaunchTemplateVersions`
- `ec2:DeleteLaunchTemplate`
- `ec2:DeleteLaunchTemplateVersions`
- `ec2:DescribeLaunchTemplate`
- `ec2:DescribeLaunchTemplateVersions`

Das folgende Beispiel für eine IAM-Richtlinie beinhaltet diese Berechtigungen.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
```

```
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersions",
        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteLaunchTemplateVersions",
        "ec2:DescribeLaunchTemplate",
        "ec2:DescribeLaunchTemplateVersions"
      ],
      "Resource": [
                "*"
            ]
    }
  ]
}
```

Weitere Informationen erhalten Sie unter Rollen von Elastic Beanstalk Service verwalten und Elastic Beanstalk-Benutzerrichtlinien verwalten.

Weitere Informationen zu Startvorlagen

Weitere Informationen zu Startvorlagen finden Sie unter Auto Scaling Scaling-Startvorlagen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Weitere Informationen über die AWS Umstellung auf Startvorlagen und die damit verbundenen Vorteile finden Sie im AWS Compute-Blog unter Amazon EC2 Auto Scaling wird Launch Configurations keine Unterstützung mehr für neue EC2-Funktionen hinzufügen.

> **Important**
>
> Sie müssen das in diesem Blogartikel beschriebene Verfahren nicht befolgen, um eine ältere Umgebung auf Startvorlagen umzustellen. Um eine bestehende Elastic Beanstalk Beanstalk-Umgebung zum Starten von Vorlagen zu migrieren, legen Sie eine der unter aufgeführten Optionen fest. Optionseinstellungen für Startvorlagen

Spot-Instance-Unterstützung für Ihre Elastic Beanstalk Beanstalk-Umgebung

In diesem Thema werden die Konfigurationsoptionen beschrieben, die Ihnen zur Verwaltung der Kapazität und des Lastenausgleichs von Spot-Instances in Ihrer Elastic Beanstalk Beanstalk-Umgebung zur Verfügung stehen. Es enthält auch Details und Beispiele für die Methoden, mit

denen Sie diese Optionen konfigurieren können. Sie können die Elastic Beanstalk Beanstalk-Konsole, die Namespace-Konfigurationsoptionen, die oder die EB-CLI verwenden AWS CLI, um die Konfigurationsoptionen zu verwalten.

Minimiere Unterbrechungen von Spot-Instances mit Capacity Rebalancing

Um die Auswirkungen von Spot-Instance-Unterbrechungen auf Ihre Anwendung zu minimieren, können Sie die in Amazon EC2 Auto Scaling enthaltene Option Capacity Rebalancing aktivieren.

> **Important**
>
> Die Nachfrage nach Spot-Instances kann sich von einem Moment zum anderen erheblich unterscheiden und auch die Verfügbarkeit von Spot-Instances kann sich abhängig von der Verfügbarkeit nicht genutzter Amazon EC2-Instances erheblich unterscheiden. Zudem besteht immer die Möglichkeit, dass Ihre Spot-Instance unterbrochen wird.

Wenn Sie Capacity Rebalancing aktivieren, versucht EC2 automatisch, Spot-Instances in einer Auto Scaling Scaling-Gruppe zu ersetzen, bevor sie unterbrochen werden. So konfigurieren Sie Ihre Auto Scaling-Gruppe in der Elastic Beanstalk-Konsole. Alternativ können Sie die Elastic Beanstalk`EnableCapacityRebalancing`-Konfigurationsoption auf `true` im aws:autoscaling:asg-Namespace setzen.

Weitere Informationen finden Sie unter Capacity Rebalancing im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch und unter Spot-Instance-Unterbrechungen im Amazon EC2-Benutzerhandbuch.

Ältere Instance-Typen und Spot-Instance-Unterstützung

Einige ältere AWS Konten bieten Elastic Beanstalk möglicherweise Standard-Instance-Typen, die Spot-Instances nicht unterstützen. Wenn Sie Spot-Instance-Anfragen aktivieren und der Fehler angezeigt wird, dass keiner der von Ihnen angegebenen Instance-Typen Spot unterstützt, aktualisieren Sie Ihre Konfiguration mit Instance-Typen, die Spot-Instances unterstützen. Verwenden Sie zur Auswahl von Spot-Instance-Typen Spot Instance Advisor.

Themen

- Spot-Instances für Ihre Umgebung aktivieren
- Strategie zur Zuweisung von Spot-Instances
- Verwaltung von On-Demand-Instances und Spot-Instances

- Kapazitätskonfiguration für Ihre Elastic Beanstalk Beanstalk-Umgebung

Spot-Instances für Ihre Umgebung aktivieren

Um die Vorteile von Amazon EC2 Spot-Instances zu nutzen, legen Sie die `EnableSpot` Option für Ihre Umgebung fest. Die Auto Scaling-Gruppe Ihrer Umgebung kombiniert dann Amazon EC2-Kaufoptionen und verwaltet eine Mischung aus On Demand- und Spot-Instances.

Sie können die Elastic Beanstalk Beanstalk-Konsole, die Namespace-Konfigurationsoptionen, die oder die EB-CLI verwenden AWS CLI, um Spot-Instance-Anfragen für Ihre Umgebung zu aktivieren.

Bevor Sie Spot-Instances für Ihre Umgebung aktivieren, sollten Sie sich mit den verfügbaren Konfigurationsoptionen für Auto Scaling, Kapazität und Lastenausgleich vertraut machen. Die Anforderungen Ihrer Anwendung in Bezug auf die Arbeitslast, die Auswirkungen von Instance-Unterbrechungen und die Preisgestaltung sind allesamt wichtige Überlegungen bei Ihrer Planung zur Aktivierung von Spot-Instances.

In den folgenden Themen finden Sie Einzelheiten zu den Optionen Auto Scaling und Kapazitätsmanagement und zu den Auswirkungen ihrer kombinierten Verwendung auf Ihre Umgebung. Es gibt Verfahren und Beispielkonfigurationen, die Sie über die verschiedenen Optionen und deren Konfiguration informieren und unterstützen. Wir bieten auch Tools und Funktionen, mit denen Sie Ihre Konfiguration verwalten und auf Ereignisse reagieren können. Sie können automatische Änderungen an Ihrer Konfiguration auf der Grundlage vorhersehbarer Datenverkehrsperioden planen, Trigger konfigurieren, um auf Faktoren wie das Verkehrsaufkommen zu reagieren, und Auto Scaling Scaling-Überwachung und Zustandsprüfungen konfigurieren.

Ausführlichere Informationen zu Spot-Instances, einschließlich Erläuterungen der wichtigsten Konzepte und bewährten Methoden, finden Sie unter Spot-Instances im Amazon EC2 EC2-Benutzerhandbuch.

> **Important**
>
> Die `EnableSpot` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration

aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

Strategie zur Zuweisung von Spot-Instances

Sie können eine der in diesem Thema aufgeführten Allokationsstrategien für Ihre Elastic Beanstalk Beanstalk-Umgebung auswählen. Verwenden Sie die Elastic Beanstalk Beanstalk-Konsole, die Namespace-Konfigurationsoptionen oder die AWS CLI, um die Zuweisungsstrategie für Spot-Instances und zugehörige Attribute für Ihre Umgebung festzulegen und zu konfigurieren.

Amazon EC2 wendet eine Zuweisungsstrategie an, um Spot-Instances für Ihre Umgebung zu verwalten und bereitzustellen. Jede Zuweisungsstrategie optimiert die zugewiesenen Instances auf der Grundlage ihrer Definition, um die verfügbare Kapazität, den Preis und die Auswahl der Instance-Typen zu handhaben.

Amazon EC2 Auto Scaling bietet die folgenden Zuweisungsstrategien für Spot-Instances.

- Kapazitätsoptimiert (Standard)
  - Fordert Spot-Instances aus dem Pool mit optimaler Kapazität für die Anzahl der Instances an, die gestartet werden.
  - Diese Strategie eignet sich gut für Workloads, bei denen die Möglichkeit einer Betriebsunterbrechung minimiert werden muss.
- Preis, Kapazität optimiert
  - Fordert Spot-Instances aus den Pools an, die die geringste Wahrscheinlichkeit einer Unterbrechung und den niedrigstmöglichen Preis haben.
  - Dies ist die bevorzugte Wahl für die meisten Spot-Workloads.
- Kapazität optimiert, priorisiert
  - Fordert Spot-Instances zunächst auf der Grundlage der Kapazitätsverfügbarkeit an und berücksichtigt dabei die von Ihnen gewählte Priorisierung des Instance-Typs nach bestem Wissen und Gewissen. Sie können eine nach Priorität geordnete Liste von Instance-Typen bereitstellen, wenn Sie Spot-Instance-Optionen für Elastic Beanstalk konfigurieren.
  - Diese Strategie eignet sich für Workloads, die nur minimale Serviceunterbrechungen erfordern, und die Priorisierung eines bestimmten Instance-Typs ist wichtig.
- niedrigster Preis
  - Fordert Spot-Instances aus dem Pool mit den günstigsten verfügbaren Instances an.

- Bei der Verwendung dieser Strategie ist es wichtig, Vorsichtsmaßnahmen zu treffen, da nur der Instance-Preis und nicht die Kapazitätsverfügbarkeit berücksichtigt wird, was zu hohen Unterbrechungsraten führen wird.

Weitere Informationen zu den einzelnen Zuweisungsstrategien finden Sie unter Zuweisungsstrategien für mehrere Instance-Typen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Um zu verstehen, welche Zuweisungsstrategie für die Anforderungen Ihrer Umgebung am besten geeignet ist, finden Sie im Amazon EC2 EC2-Benutzerhandbuch unter Auswahl der geeigneten Spot-Zuweisungsstrategie.

Verwaltung von On-Demand-Instances und Spot-Instances

Sie können eine Flotte von On-Demand-Instances und Spot-Instances innerhalb einer einzigen Auto-Scaling-Gruppe starten und automatisch skalieren. Die folgenden Optionen können zusammen verwendet werden, um zu konfigurieren, wie der Auto Scaling Scaling-Dienst Spot-Instances und On-Demand-Instances in Ihrer Umgebung verwaltet.

Sie können diese Optionen für Ihre Umgebung mit der Elastic Beanstalk Beanstalk-Konsole, den Namespace-Konfigurationsoptionen AWS CLI, der oder der EB-CLI konfigurieren.

Diese Optionen sind Teil des Namespace aws:ec2:instances:

- `EnableSpot`- Wenn `true` diese Einstellung aktiviert ist, werden Spot-Instance-Anfragen für Ihre Umgebung aktiviert.
- `SpotFleetOnDemandBase`- Legt die Mindestanzahl von On-Demand-Instances fest, die Ihre Auto Scaling Scaling-Gruppe bereitstellt, bevor Spot-Instances bei der Skalierung Ihrer Umgebung berücksichtigt werden.
- `SpotFleetOnDemandAboveBasePercentage`- Der Prozentsatz der On-Demand-Instances als Teil der zusätzlichen Kapazität, die Ihre Auto Scaling Scaling-Gruppe über die `SpotOnDemandBase` Instances hinaus bereitstellt.

Die zuvor aufgelisteten Optionen entsprechen den folgenden Optionen im aws:autoscaling:asg-Namespace:

- `MinSize`- Die Mindestanzahl von Instances, die Sie in Ihrer Auto Scaling Scaling-Gruppe haben möchten.

- `MaxSize`- Die maximale Anzahl von Instances, die Sie in Ihrer Auto Scaling Scaling-Gruppe haben möchten.

> **Important**
>
> Die `EnableSpot` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

Beide Gruppen von Namespace-Optionen anwenden

In den folgenden Punkten wird beschrieben, wie sich die Kombination dieser Optionseinstellungen auf die Skalierung für Ihre Umgebung auswirkt.

- Von `MinSize` hängt die anfängliche Kapazität Ihrer Umgebung ab, d.h. die Anzahl der Instances, die Sie mindestens ausführen möchten.
- `SpotFleetOnDemandBase` hat keinen Einfluss auf die Anfangskapazität. Wenn Spot aktiviert ist, bestimmt diese Option, wie viele On-Demand-Instances bereitgestellt werden, bevor Spot-Instances berücksichtigt werden.
- Achten Sie darauf, wann `SpotFleetOnDemandBase` kleiner ist als `MinSize`. Sie erhalten immer noch genau `MinSize` Instances als Anfangskapazität. Mindestens `SpotFleetOnDemandBase` von ihnen sollten On Demand-Instances sein.
- Achten Sie darauf, wann `SpotFleetOnDemandBase` größer ist als `MinSize`. Wenn die Skalierung Ihrer Umgebung erfolgt, erhalten Sie mindestens die Anzahl zusätzlicher Instances, die der Differenz zwischen den beiden Werten entspricht. Mit anderen Worten, Sie erhalten garantiert mindestens zusätzliche `(SpotFleetOnDemandBase - MinSize)`-Instances, die in Bereitstellung sind, bevor die `SpotFleetOnDemandBase` Anforderung gesättigt wird.

Umgebungen mit einer Instance

In Produktionsumgebungen sind Spot-Instances in skalierbaren Umgebungen mit Lastausgleich besonders nützlich. Wir empfehlen nicht, Spot-Instances in Umgebungen mit einer einzigen Instance zu verwenden. Wenn keine Spot-Instances verfügbar sind, verlieren Sie möglicherweise die gesamte Kapazität (eine einzelne Instance) Ihrer Umgebung. Möglicherweise möchten Sie dennoch eine Spot-Instance in einer Umgebung mit einer einzelnen Instance für die Entwicklung oder das Testen verwenden. Stellen Sie in diesem Fall sicher, dass Sie sowohl `SpotFleetOnDemandBase` als auch `SpotFleetOnDemandAboveBasePercentage` auf Null setzen. Alle anderen Einstellungen führen zu einer On-Demand-Instance.

Beispiele für Einstellungen für Skalierungsoptionen

Die folgenden Beispiele veranschaulichen verschiedene Szenarien zum Festlegen der verschiedenen Skalierungsoptionen. Alle Beispiele gehen von einer Umgebung mit Lastausgleich aus, in der Spot-Instance-Anforderungen aktiviert wurden.

Example 1: On-Demand- und Spot-Instances als Teil der Anfangskapazität

Optionseinstellungen

Option	Namespace	Wert
`MinSize`	`aws:autoscaling:asg`	`10`
`MaxSize`	`aws:autoscaling:asg`	`24`
`SpotFleetOnDemandBase`	`aws:ec2:instances`	`4`
`SpotFleetOnDemandAboveBasePercentage`	`aws:ec2:instances`	`50`

In diesem Beispiel wird in der Umgebung mit zehn Instances begonnen, von denen sieben On-Demand-Instances (vier Basis-Instances und 50 % der sechs oberhalb der Basis) und drei Spot-Instances sind. Die Umgebung kann bis zu 24 Instances skalieren. Bei der Skalierung wird der Anteil an On-Demand-Instances in dem Flottenteil oberhalb der vier On-Demand-Basis-Instances bei 50 % gehalten, bis hin zu einem Maximum von insgesamt 24 Instances, von denen 14 On-Demand-Instances (vier On-Demand-Basis-Instances und 50 % der 20 oberhalb der Basis) und zehn Spot-Instances sind.

Example 2: Gesamte On-Demand-Anfangskapazität

Optionseinstellungen

Option	Namespace	Wert
`MinSize`	`aws:autoscaling:asg`	`4`
`MaxSize`	`aws:autoscaling:asg`	`24`
`SpotFleetOnDemandBase`	`aws:ec2:instances`	`4`
`SpotFleetOnDemandAboveBasePercentage`	`aws:ec2:instances`	`50`

In diesem Beispiel wird in der Umgebung mit vier Instances begonnen, wovon alle On-Demand-Instances sind. Die Umgebung kann bis zu 24 Instances skalieren. Bei der Skalierung wird der Anteil an On-Demand-Instances in dem Flottenteil oberhalb der vier On-Demand-Basis-Instances bei 50 % gehalten, bis hin zu einem Maximum von insgesamt 24 Instances, von denen 14 On-Demand-Instances (vier On-Demand-Basis-Instances und 50 % der 20 oberhalb der Basis) und zehn Spot-Instances sind.

Example 3: Zusätzliche On-Demand-Basis über die anfängliche Kapazität hinaus

Optionseinstellungen

Option	Namespace	Wert
`MinSize`	`aws:autoscaling:asg`	`3`
`MaxSize`	`aws:autoscaling:asg`	`24`
`SpotFleetOnDemandBase`	`aws:ec2:instances`	`4`
`SpotFleetOnDemandAboveBasePercentage`	`aws:ec2:instances`	`50`

In diesem Beispiel wird in der Umgebung mit drei Instances begonnen, wovon alle On-Demand-Instances sind. Die Umgebung kann bis zu 24 Instances skalieren. Die erste zusätzliche Instance oberhalb der ersten drei ist On-Demand, um die vier On-Demand-Basis-Instances zu vervollständigen. Bei der weiteren Skalierung wird der Anteil an On-Demand-Instances in dem Flottenteil oberhalb der vier On-Demand-Basis-Instances bei 50 % gehalten, bis hin zu einem Maximum von insgesamt 24 Instances, von denen 14 On-Demand-Instances (vier On-Demand-Basis-Instances und 50 % der 20 oberhalb der Basis) und zehn Spot-Instances sind.

Kapazitätskonfiguration für Ihre Elastic Beanstalk Beanstalk-Umgebung

In diesem Thema werden die verschiedenen Ansätze zur Konfiguration der Auto Scaling-Kapazität für Ihre Elastic Beanstalk Beanstalk-Umgebung beschrieben. Sie können die Elastic Beanstalk Beanstalk-Konsole, die EB-CLI AWS CLI, die oder Namespace-Optionen verwenden.

> **Important**
>
> Die `EnableSpot` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

Konfiguration mithilfe der Konsole

Sie können das Kapazitätsmanagement einer Auto Scaling Scaling-Gruppe konfigurieren, indem Sie Capacity auf der Konfigurationsseite der Umgebung in der Elastic Beanstalk Beanstalk-Konsole bearbeiten.

So konfigurieren Sie die Auto Scaling Scaling-Gruppenkapazität in der Elastic Beanstalk Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Konfigurieren Sie im Abschnitt Auto Scaling Group (Auto Scaling-Gruppe) die folgenden Einstellungen.

   - Environment type (Umgebungstyp) – Wählen Sie Load balanced (Lastverteilt) aus.
   - Min instances (Mindestanzahl Instances) – Die Mindestanzahl an EC2-Instances, die die Gruppe stets enthalten sollte. Die Gruppe beginnt mit der Mindestanzahl und fügt Instances hinzu, wenn eine Hochskalierung ausgelöst wird.
   - Max instances (Höchstanzahl Instances) – Die Höchstanzahl an EC2-Instances, die die Gruppe stets enthalten sollte.

     > **Note**
     >
     > Wenn Sie mit fortlaufenden Updates arbeiten, stellen Sie sicher, dass die Höchstanzahl an Instances über der Einstellung Minimum instances in service (Mindestanzahl ausgeführter Instances) für fortlaufende Updates liegt.

   - Zusammensetzung der Flotte – Der Standardwert ist On-Demand-Instances. Wählen Sie Kombinierte Kaufoptionen und Instances aus, um Spot-Instance-Anfragen zu aktivieren.

     > **Important**
     >
     > Die `EnableSpot` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

   Die folgenden Optionen sind bei Spot-Instance-Anfragen aktiviert:

   - Spot-Zuweisungsstrategie — Bestimmt die Methode zur Verwaltung und Bereitstellung der Spot-Instances in Ihrer Umgebung auf der Grundlage der verfügbaren Kapazität, des

Preises und der Auswahl der Instance-Typen. Wählen Sie zwischen Kapazitätsoptimiert (Standard), Preis, Kapazitätsoptimiert, Kapazitätsoptimiert, priorisiert oder Niedrigster Preis. Eine Beschreibung der einzelnen Zuweisungsstrategien und weitere Informationen finden Sie unterthe section called “Strategie für die Spot-Zuweisung”.

- Maximaler Spot-Preis — Empfehlungen zu Höchstpreisoptionen für Spot-Instances finden Sie in der Preisentwicklung für Spot-Instances im Amazon EC2 EC2-Benutzerhandbuch.
- On-Demand-Basis – Die Mindestanzahl von On-Demand-Instances, die von der Auto-Scaling-Gruppe bereitgestellt werden, bevor Spot-Instances bei der Skalierung Ihrer Umgebung in Betracht gezogen werden.
- On-Demand über Basis – Der Prozentsatz der On-Demand-Instances, die als Teil der zusätzlichen Kapazität der Auto-Scaling-Gruppe, über die -Instances hinaus, bereitgestellt wird.

> **Note**
>
> Die Optionen On-Demand-Basis und On-Demand über Basis korrelieren mit den Min (Min.) und Max (Max.) Instances-Optionen, die weiter oben aufgeführt sind. Weitere Informationen zu diesen Optionen und Beispielen finden Sie unter the section called “Spot-Instance-Unterstützung”.

- Kapazitätsausgleich — Diese Option ist nur relevant, wenn in Ihrer Auto Scaling Scaling-Gruppe mindestens eine Spot-Instance vorhanden ist. Wenn diese Funktion aktiviert ist, versucht EC2 automatisch, Spot-Instances in der Auto Scaling Scaling-Gruppe zu ersetzen, bevor sie unterbrochen werden, wodurch Spot-Instance-Unterbrechungen Ihrer Anwendungen minimiert werden. Weitere Informationen finden Sie unter Capacity Rebalancing im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch

- Architektur — Die Prozessorarchitektur für Ihre EC2-Instances. Die Prozessorarchitektur bestimmt die EC2-Instance-Typen, die im nächsten Feld verfügbar werden.
- Instance-Typen — Die Typen von Amazon EC2 EC2-Instances, die zur Ausführung Ihrer Anwendung gestartet wurden. Details hierzu finden Sie unter the section called “Instance-Typen”.
- AMI ID (AMI-ID) – Das Maschinen-Image, das von Elastic Beanstalk zum Starten von Amazon EC2-Instances in Ihrer Umgebung verwendet wird. Details hierzu finden Sie unter the section called “AMI-ID”.
- Availability Zones (Verfügbarkeitszonen) – Legen Sie die Anzahl an Availability Zones fest, auf die die Instances der Umgebung verteilt werden. Standardmäßig startet die Auto-Scaling-

Gruppe Instances gleichmäßig über alle nutzbaren Zonen verteilt. Um Instances auf weniger Zonen zu verteilen, legen Sie die Anzahl der zu verwendenden Zonen fest. Verwenden Sie für Produktionsumgebungen mindestens zwei Zonen, um sicherzustellen, dass Ihre Anwendung auch beim Ausfall einer Availability Zone weiterhin verfügbar ist.

- Placement (Platzierung) (optional) – Wählen Sie die zu verwendenden Availability Zones aus. Verwenden Sie diese Einstellung, wenn Ihre Instances auf Ressourcen in bestimmten Zonen zugreifen müssen oder wenn Sie zonenspezifische Reserved Instances (Vorbehaltene Instances) gekauft haben. Wenn Sie Ihre Umgebung in einer benutzerdefinierten VPC starten, können Sie diese Option nicht konfigurieren. In einer benutzerdefinierten VPC wählen Sie Availability Zones für die Subnetze aus, die Sie der Umgebung zuweisen.
- Scaling cooldown (Ruhephase für die Skalierung) – Die Wartezeit (in Sekunden) für das Starten oder Beenden von Instances nach einer Skalierung, bevor Auslöser weiter ausgewertet werden. Weitere Informationen finden Sie unter Skalieren von Ruhephasen.

6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Konfiguration mithilfe von Namespace-Optionen

Elastic Beanstalk stellt Konfigurationsoptionen für die Auto-Scaling-Einstellungen in zwei Namespaces bereit: `aws:autoscaling:asg` und `aws:ec2:instances`.

Der Namespace aws:autoscaling:asg

Der `aws:autoscaling:asg`-Namespace bietet Optionen für die allgemeine Skalierung und Verfügbarkeit.

Im folgenden Beispiel einer Konfigurationsdatei wird die Auto-Scaling-Gruppe so konfiguriert, dass sie zwei bis vier Instances, bestimmte Availability Zones und eine Ruhephase von 12 Minuten (720 Sekunden) verwendet. Es ermöglicht den Kapazitätsausgleich für Spot-Instances. Diese `EnableCapacityRebalancing` Option `EnableSpot` ist nur wirksam, wenn sie `true` im `aws:ec2:instances`Namespace auf gesetzt ist, wie in der nachfolgenden Konfigurationsdatei gezeigt.

```
option_settings:
  aws:autoscaling:asg:
    Availability Zones: Any
    Cooldown: '720'
    Custom Availability Zones: 'us-west-2a,us-west-2b'
```

```
    MaxSize: '4'
    MinSize: '2'
    EnableCapacityRebalancing: true
```

Der Namespace aws:ec2:instances

> **Note**
>
> Wenn Sie die Umgebungskonfiguration aktualisieren und einen oder mehrere Instance-Typen aus der `InstanceTypes`-Option entfernen, beendet Elastic Beanstalk alle Amazon-EC2-Instances, die auf einem der entfernten Instance-Typen ausgeführt werden. Die Auto-Scaling-Gruppe Ihrer Umgebung startet dann nach Bedarf neue Instances, um die gewünschte Kapazität zu vervollständigen, unter Verwendung der aktuell angegebenen Instance-Typen.

Der `aws:ec2:instances` Namespace bietet Optionen, die sich auf die Instances Ihrer Umgebung beziehen, einschließlich der Spot-Instance-Verwaltung. Ergänzt werden `aws:autoscaling:launchconfiguration` und `aws:autoscaling:asg`.

Folgendes Beispiel einer Konfigurationsdatei konfiguriert die Auto Scaling-Gruppe, um Spot-Instance-Anforderungen für Ihre Umgebung zu aktivieren. Sie bezeichnet drei mögliche Instanztypen, die verwendet werden können. Mindestens eine On-Demand-Instancewird fürBaseline-Kapazität und dauerhaft 33 % der On-Demand-Instances werden für jede zusätzliche Kapazität verwendet.

Die Konfiguration legt die Spot-Zuweisungsstrategie auf fest. `capacity-optimized-prioritized` Diese spezielle Zuweisungsstrategie priorisiert die Instance-Starts aus dem Pool auf der Grundlage der in der Option angegebenen Reihenfolge der Instance-Typen. `InstanceTypes` Wenn nicht angegeben, `SpotAllocationStrategy` ist die Standardeinstellung. `capacity-optimized`

```
option_settings:
  aws:ec2:instances:
    EnableSpot: true
    InstanceTypes: 't2.micro,t3.micro,t3.small'
    SpotAllocationStrategy: capacity-optimized-prioritized
    SpotFleetOnDemandBase: '1'
    SpotFleetOnDemandAboveBasePercentage: '33'
```

Verwenden Sie zur Auswahl von Spot-Instance-Typen Spot Instance Advisor.

> **Important**
>
> Die `EnableSpot` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

Konfiguration mit dem AWS CLI

Dieser Abschnitt enthält Beispiele dafür, wie Sie den Befehl AWS CLI create-environment verwenden können, um Ihre Umgebung mit den in diesen Abschnitten beschriebenen Auto Scaling- und Capacity-Optionen zu konfigurieren. Sie werden feststellen, dass die Namespace-Einstellungen für `aws:autoscaling:asg`und `aws:ec2:instances`, wie im vorherigen Abschnitt mit den Namespace-Konfigurationsoptionen beschrieben, ebenfalls anhand dieses Beispiels konfiguriert wurden.

Die AWS Befehlszeilenschnittstelle bietet Befehle zum Erstellen und Konfigurieren von Elastic Beanstalk Beanstalk-Umgebungen. Mit der `--option-settings` Option können Sie Namespace-Optionen übergeben, die von Elastic Beanstalk unterstützt werden. Das bedeutet, dass die zuvor beschriebenen Namespace-Konfigurationsoptionen in entsprechende AWS CLI Befehle zur Konfiguration Ihrer Elastic Beanstalk Beanstalk-Umgebung übergeben werden können.

> **Note**
>
> Sie können auch den Befehl update-environment mit verwenden, um Namespace-Optionen hinzuzufügen oder `--option-settings` zu aktualisieren. Wenn Sie Namespace-Optionen aus Ihrer Umgebung entfernen müssen, verwenden Sie den Befehl with. update-environment `--options-to-remove`

Das folgende Beispiel erstellt eine neue Umgebung. Weitere Informationen zu den übergebenen Optionen finden Sie im vorherigen Thema Namespace-Konfigurationsoptionen.

Die erste `IamInstanceProfile` im aws:autoscaling:launchconfiguration Namespace aufgeführte Option ist das Elastic Beanstalk Beanstalk-Instance-Profil. Es ist erforderlich, wenn Sie eine neue Umgebung erstellen.

Example— Umgebung mit Auto Scaling Scaling-Optionen erstellen (Namespace-Optionen inline)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v4.3.0 running Python 3.12" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role
Namespace=aws:autoscaling:asg,OptionName=Availability Zones,Value=Any \
Namespace=aws:autoscaling:asg,OptionName=Cooldown,Value=720 \
Namespace=aws:autoscaling:asg,OptionName=Custom Availability Zones,Value=us-west-2a,us-west-2b \
Namespace=aws:autoscaling:asg,OptionName=MaxSize,Value=4 \
Namespace=aws:autoscaling:asg,OptionName=MinSize,Value=2 \
Namespace=aws:autoscaling:asg,OptionName=EnableCapacityRebalancing,Value=true \
Namespace=aws:ec2:instances,OptionName=EnableSpot,Value=true \
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t2.micro,t3.micro,t3.small \
Namespace=aws:ec2:instances,OptionName=SpotAllocationStrategy,Value=capacity-optimized-prioritized \
Namespace=aws:ec2:instances,OptionName=SpotFleetOnDemandBase,Value=1 \
Namespace=aws:ec2:instances,OptionName=SpotFleetOnDemandAboveBasePercentage,Value=33
```

> **Important**
>
> Die `EnableSpot` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

Verwenden Sie alternativ eine`options.json`-Datei, um die Namespace-Optionen anzugeben, anstatt sie einzubinden.

Example—create-environment mit Auto Scaling Scaling-Optionen (Namespace-Optionen in der Datei) options.json

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v4.3.0 running Python 3.12"
--option-settings file://options.json
```

Example

```
### example options.json ###
[
    {
        "Namespace": "aws:autoscaling:launchconfiguration",
        "OptionName": "IamInstanceProfile",
        "Value": "aws-elasticbeanstalk-ec2-role"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "Availability Zones",
        "Value": "Any"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "Cooldown",
        "Value": "720"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "Custom Availability Zones",
        "Value": "us-west-2a,us-west-2b"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MaxSize",
        "Value": "4"
    },
```

```
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MinSize",
        "Value": "2"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "EnableCapacityRebalancing",
        "Value": "true"
    },
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "EnableSpot",
        "Value": "true"
    },
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "InstanceTypes",
        "Value": "t2.micro,t3.micro,t3.small"
    },
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "SpotAllocationStrategy",
        "Value": "capacity-optimized-prioritized"
    },
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "SpotFleetOnDemandBase",
        "Value": "1"
    },
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "SpotFleetOnDemandAboveBasePercentage",
        "Value": "33"
    }
]
```

Konfiguration mit der EB CLI

Wenn Sie mit dem Befehl eb create eine Umgebung erstellen, können Sie einige Optionen bezüglich der Auto-Scaling-Gruppe Ihrer Umgebung angeben. Mit diesen Optionen können Sie die Kapazität der Umgebung steuern.

`--single`

Erstellt eine Umgebung mit einer Amazon-EC2-Instance ohne Load Balancer. Wenn Sie diese Option nicht verwenden, wird ein Load Balancer der Umgebung hinzugefügt, die erstellt wurde.

`--enable-spot`

Aktivieren Sie Spot-Instance-Anforderungen für Ihre Umgebung.

> **Important**
>
> Die `enable-spot` Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.

Die folgenden Optionen für den Befehl eb create können nur zusammen mit `--enable-spot` verwendet werden.

`--instance-types`

Listen Sie die Amazon-EC2-Instance-Typen auf, die Ihre Umgebung verwenden soll.

`--spot-max-price`

Der Höchstpreis pro Einheitsstunde in USD, den Sie für eine Spot-Instance zu zahlen bereit sind. Empfehlungen zu Höchstpreisoptionen für Spot-Instances finden Sie in der Preisentwicklung für Spot-Instances im Amazon EC2 EC2-Benutzerhandbuch.

`--on-demand-base-capacity`

Die Mindestanzahl von On-Demand-Instances, die von der Auto-Scaling-Gruppe bereitgestellt werden, bevor Spot-Instances bei der Skalierung Ihrer Umgebung in Betracht gezogen werden.

`--on-demand-above-base-capacity`

Der Anteil der On-Demand-Instances als Teil der zusätzlichen Kapazität, den Ihre Auto-Scaling-Gruppe über die durch die `--on-demand-base-capacity`- Option angegebene Anzahl von Instances hinaus bereitstellt.

Im folgenden Beispiel wird eine Umgebung erstellt und die Auto-Scaling-Gruppe so konfiguriert, dass Spot-Instance-Anforderungen für die neue Umgebung aktiviert werden. Dabei sind drei Instance-Typen möglich. Für dieses Beispiel, können drei mögliche Instance-Typen verwendet werden.

```
$ eb create --enable-spot --instance-types "t2.micro,t3.micro,t3.small"
```

Important

Es gibt eine andere ähnlich benannte Option, die `--instance-type` (keine “s”), die die EB-CLI nur bei der Verarbeitung von On-Demand-Instances erkennt. Verwenden Sie nicht `--instance-type` (kein „s“) mit der `--enable-spot`-Option. Wenn Sie dies tun, ignoriert die EB-CLI es. Verwenden Sie stattdessen `--instance-types` (mit „s“) mit der `--enable-spot`-Option.

Auto Scaling-Trigger für Ihre Elastic Beanstalk Beanstalk-Umgebung

Die Auto Scaling Scaling-Gruppe in Ihrer Elastic Beanstalk Beanstalk-Umgebung verwendet zwei CloudWatch Amazon-Alarme, um Skalierungsvorgänge auszulösen. Mit den Standardauslösern wird die Skalierung durchgeführt, wenn der durchschnittliche ausgehende Netzwerkdatenverkehr von der jeweiligen Instance innerhalb von 5 Minuten 6 MB über- oder 2 MB unterschreitet. Um Amazon EC2 Auto Scaling effektiv zu nutzen, konfigurieren Sie Trigger, die für Ihre Anwendung, Ihren Instance-Typ und Ihre Serviceanforderungen geeignet sind. Sie können die Skalierung anhand unterschiedlicher Statistiken wie Latenz, Festplatten-I/O-Vorgänge, CPU-Auslastung und Anforderungsanzahl ausführen.

Weitere Informationen zu CloudWatch Metriken und Alarmen finden Sie unter Amazon CloudWatch Concepts im CloudWatch Amazon-Benutzerhandbuch.

Konfigurieren von Auto-Scaling-Auslöser

Sie können die Auslöser konfigurieren, über die die Anzahl der Instances in der Auto-Scaling-Gruppe Ihrer Umgebung in der Elastic-Beanstalk-Konsole angepasst werden.

So konfigurieren Sie Auslöser in der Elastic-Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Konfigurieren Sie im Abschnitt Scaling triggers (Skalierungsauslöser) die folgenden Einstellungen:
   - Metric (Metrik) – Metrik für Ihren Auto Scaling-Auslöser.
   - Statistic (Statistik) – Vom Auslöser zu verwendende Statistikberechnung, z. B. `Average`.
   - Unit (Einheit) – Einheit für die Metrik des Auslösers, z. B. Bytes.
   - Zeitraum — Gibt an, wie oft Amazon die Metriken für Ihren Trigger CloudWatch misst.
   - Breach duration (Verletzungsdauer) – Zeitdauer (in Minuten), die eine Metrik außerhalb der Ober- und Untergrenzen liegen kann, bevor eine Skalierungsoperation ausgelöst wird.
   - Upper threshold (Obergrenze) – Falls die Metrik diesen Wert der Verletzungsdauer überschreitet, wird eine Skalierungsoperation ausgelöst.
   - Skalierungsinkrement — Die Anzahl der EC2 Amazon-Instances, die hinzugefügt werden müssen, wenn eine Skalierungsaktivität ausgeführt wird.
   - Lower threshold (Untergrenze) – Wenn die Metrik diesen Wert der Verletzungsdauer unterschreitet, wird eine Skalierungsoperation ausgelöst.
   - Skalierungsinkrement — Die Anzahl der EC2 Amazon-Instances, die entfernt werden müssen, wenn eine Skalierungsaktivität ausgeführt wird.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Der Namespace aws:autoscaling:trigger

Elastic Beanstalk stellt Konfigurationsoptionen für die Auto Scaling-Einstellungen im Namespace `aws:autoscaling:trigger` bereit. Die Einstellungen in diesem Namespace sind nach Ressourcen gruppiert, auf die sie angewendet werden.

```
option_settings:
  AWSEBAutoScalingScaleDownPolicy.aws:autoscaling:trigger:
    LowerBreachScaleIncrement: '-1'
  AWSEBAutoScalingScaleUpPolicy.aws:autoscaling:trigger:
    UpperBreachScaleIncrement: '1'
  AWSEBCloudwatchAlarmHigh.aws:autoscaling:trigger:
    UpperThreshold: '6000000'
  AWSEBCloudwatchAlarmLow.aws:autoscaling:trigger:
    BreachDuration: '5'
    EvaluationPeriods: '1'
    LowerThreshold: '2000000'
    MeasureName: NetworkOut
    Period: '5'
    Statistic: Average
    Unit: Bytes
```

Geplante Auto Scaling Scaling-Aktionen für Ihre Elastic Beanstalk Beanstalk-Umgebungen

Um die Nutzung von EC2 Amazon-Instances in Ihrer Umgebung in vorhersehbaren Spitzenzeiten zu optimieren, konfigurieren Sie Ihre Amazon EC2 Auto Scaling Scaling-Gruppe so, dass die Anzahl der Instances nach einem Zeitplan geändert wird. Sie können Ihre Umgebung mit einer wiederkehrenden Aktion so konfigurieren, dass sie jeden Morgen hochskaliert und nachts bei geringem Datenverkehr herunterskaliert wird. Wenn Sie beispielsweise während eines Marketing-Ereignisses für einen begrenzten Zeitraum ein erhöhtes Datenvolumen erwarten, können Sie ein einmaliges Ereignis planen, um die Umgebung zu Beginn des Marketing-Ereignisses hochzuskalieren und an dessen Ende wieder herunterzuskalieren.

Es können bis zu 120 aktive geplante Aktionen pro Umgebung definiert werden. Elastic Beanstalk behält darüber hinaus bis zu 150 abgelaufene geplante Aktionen bei, die Sie zum Aktualisieren Ihrer Einstellungen wiederverwenden können.

Konfigurieren von geplanten Aktionen

Sie können geplante Aktionen für die Auto-Scaling-Gruppe Ihrer Umgebung in der Elastic-Beanstalk-Konsole erstellen.

So konfigurieren Sie geplante Aktionen in der Elastic-Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Wählen Sie im Abschnitt Time-based Scaling (Zeitbasierte Skalierung) die Option Add scheduled action (Geplante Aktion hinzufügen) aus.
6. Geben Sie die folgenden Einstellungen für die geplante Aktion ein:

   - Name – Geben Sie einen eindeutigen Namen mit bis zu 255 alphanumerischen Zeichen ohne Leerzeichen ein.
   - Instances – Wählen Sie die minimale und maximale Instance-Anzahl für die Auto Scaling-Gruppe aus.
   - Desired capacity (Gewünschte Kapazität) (optional) – Legen Sie die gewünschte Anfangskapazität für die Auto Scaling-Gruppe fest. Nachdem die geplante Aktion angewendet wurde, wird die gewünschte Kapazität basierend auf den Einstellungen angepasst.
   - Occurrence (Häufigkeit) – Wählen Sie Recurring (Wiederkehrend) aus, um die Skalierungsaktion nach einem festen Zeitplan zu wiederholen.
   - Start time (Startzeit) – Wählen Sie für einmalige Aktionen das Datum und die Uhrzeit aus, zu der die Aktion ausgeführt werden soll.

     Für wiederkehrende Aktionen ist eine Startzeit optional. Geben Sie dies an, um den frühesten Zeitpunkt der Aktionsausführung auszuwählen. Ab diesem Zeitpunkt wird die Aktion entsprechend dem Ausdruck Recurrence (Wiederholung) wiederholt.
   - Recurrence (Wiederholung) – Geben Sie in einem Cron-Ausdruck an, wie häufig eine geplante Aktion ausgeführt werden soll. Mit der Option `30 6 * * 2` wird die Aktion beispielsweise jeden Dienstag um 6:30 Uhr UTC ausgeführt.

- Endtime (Endzeit) (optional) – Optional für wiederkehrende Aktionen. Bei Angabe wird die Aktion gemäß dem Ausdruck Recurrence (Wiederholung) wiederholt und ab diesem Zeitpunkt nicht mehr ausgeführt.

  Wenn eine geplante Aktion endet, wird Auto-Scaling nicht automatisch auf die vorherigen Einstellungen zurückgesetzt. Konfigurieren Sie daher gegebenenfalls eine zweite geplante Aktion, damit Auto-Scaling zu den ursprünglichen Einstellungen zurückkehrt.

7. Wählen Sie Add aus.
8. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

> **Note**
>
> Geplante Aktionen werden erst gespeichert, nachdem sie angewendet wurden.

Der Namespace aws:autoscaling:scheduledaction

Wenn Sie viele geplante Aktionen konfigurieren müssen, können Sie Konfigurationsdateien oder die Elastic-Beanstalk-API verwenden, um Änderungen an den Konfigurationsoptionen über eine YAML- oder JSON-Datei vorzunehmen. Mit diesen Methoden können Sie auch auf die `Suspend`-Option zugreifen, um eine wiederkehrende geplante Aktion vorübergehend zu deaktivieren.

> **Note**
>
> Wenn Sie außerhalb der Konsole an den Konfigurationsoptionen von geplanten Aktionen arbeiten, verwenden Sie das ISO 8601-Zeitformat, um Start- und Endzeiten in UTC anzugeben. Beispiel: 2015-04-28T04:07:02Z Weitere Informationen zum ISO 8601-Zeitformat finden Sie unter Date and Time Formats. Die Datums- und Uhrzeitangaben müssen für alle geplanten Aktionen eindeutig sein.

Elastic Beanstalk stellt Konfigurationsoptionen für die Einstellungen von geplanten Aktionen im Namespace `aws:autoscaling:scheduledaction` bereit. Verwenden Sie das Feld `resource_name`, um den Namen der geplanten Aktion anzugeben.

Example S -time-long.config cheduled-scale-up-specific

Diese Konfigurationsdatei weist Elastic Beanstalk an, um 2015-12-12T00:00:00Z von fünf Instances auf 10 Instances hochzuskalieren.

```
option_settings:
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: MinSize
    value: '5'
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: MaxSize
    value: '10'
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: DesiredCapacity
    value: '5'
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: StartTime
    value: '2015-12-12T00:00:00Z'
```

Example S cheduled-scale-up-specific -time.config

Um die Syntax-Kurznotation mit der EB-CLI oder Konfigurationsdateien zu verwenden, stellen Sie den Ressourcennamen und den Namespace voran.

```
option_settings:
  ScheduledScaleUpSpecificTime.aws:autoscaling:scheduledaction:
    MinSize: '5'
    MaxSize: '10'
    DesiredCapacity: '5'
    StartTime: '2015-12-12T00:00:00Z'
```

Example S cheduled-scale-down-specific -time.config

Diese Konfigurationsdatei weist Elastic Beanstalk an, um 2015-12-12T07:00:00Z herunterzuskalieren.

```
option_settings:
  ScheduledScaleDownSpecificTime.aws:autoscaling:scheduledaction:
    MinSize: '1'
    MaxSize: '1'
```

```
    DesiredCapacity: '1'
    StartTime: '2015-12-12T07:00:00Z'
```

Example cheduled-periodic-scale-upS .config

Diese Konfigurationsdatei weist Elastic Beanstalk an, jeden Tag um 9 Uhr morgens hochzuskalieren. Für die Aktion ist als Startzeit der 14. Mai 2015 und als Endzeit der 12. Januar 2016 geplant.

```
option_settings:
  ScheduledPeriodicScaleUp.aws:autoscaling:scheduledaction:
    MinSize: '5'
    MaxSize: '10'
    DesiredCapacity: '5'
    StartTime: '2015-05-14T07:00:00Z'
    EndTime: '2016-01-12T07:00:00Z'
    Recurrence: 0 9 * * *
```

Example S .config cheduled-periodic-scale-down

Diese Konfigurationsdatei weist Elastic Beanstalk an, jeden Tag um 18 Uhr auf eine nicht laufende Instance herunterzuskalieren. Wenn Sie wissen, dass die Anwendung außerhalb der Geschäftszeiten normalerweise ohne Last arbeitet, können Sie eine entsprechend geplante Aktion erstellen. Wenn die Anwendung außerhalb der Geschäftszeiten heruntergefahren werden muss, ändern Sie `MaxSize` auf `0`.

```
option_settings:
  ScheduledPeriodicScaleDown.aws:autoscaling:scheduledaction:
    MinSize: '0'
    MaxSize: '1'
    DesiredCapacity: '0'
    StartTime: '2015-05-14T07:00:00Z'
    EndTime: '2016-01-12T07:00:00Z'
    Recurrence: 0 18 * * *
```

Example S .config cheduled-weekend-scale-down

Diese Konfigurationsdatei weist Elastic Beanstalk an, jeden Freitag um 18 Uhr herunterzuskalieren. Wenn Sie wissen, dass für die Anwendung am Wochenende nicht sehr viel Datenverkehr entsteht, können Sie eine ähnliche Aktion planen.

```
option_settings:
  ScheduledWeekendScaleDown.aws:autoscaling:scheduledaction:
```

```
    MinSize: '1'
    MaxSize: '4'
    DesiredCapacity: '1'
    StartTime: '2015-12-12T07:00:00Z'
    EndTime: '2016-01-12T07:00:00Z'
    Recurrence: 0 18 * * 5
```

Einstellung für die Auto Scaling Scaling-Integritätsprüfung für Ihre Elastic Beanstalk Beanstalk-Umgebung

Amazon EC2 Auto Scaling überwacht den Zustand jeder Amazon Elastic Compute Cloud (Amazon EC2) -Instance, die es startet. Falls eine Instance unerwartet beendet wird, erkennt Auto Scaling dies und startet eine andere Instance als Ersatz. Standardmäßig verwendet die für Ihre Umgebung erstellte Auto Scaling Scaling-Gruppe EC2 Amazon-Statuschecks. Wenn eine Instance in Ihrer Umgebung eine EC2 Amazon-Statusprüfung nicht besteht, wird sie von Auto Scaling heruntergefahren und ersetzt.

EC2 Amazon-Statusprüfungen decken nur den Zustand einer Instance ab, nicht den Zustand Ihrer Anwendung, Ihres Servers oder aller Docker-Container, die auf der Instance ausgeführt werden. Falls die Anwendung ausfällt, aber die Instance, auf der die Anwendung ausgeführt wird, noch stabil ist, wird sie vermutlich aus dem Load Balancer genommen, aber nicht automatisch von Auto-Scaling ersetzt. Das Standardverhalten ist sinnvoll für die Fehlerbehebung. Falls die Instance von der Auto-Scaling-Funktion direkt nach dem Absturz der Anwendung ersetzt wird, bemerken Sie den Absturz möglicherweise gar nicht, auch wenn dies direkt nach dem Start geschehen ist.

Sollen die Instances, deren Anwendung nicht mehr reagiert, von Auto-Scaling ersetzt werden, können Sie die Auto Scaling-Gruppe mithilfe einer Konfigurationsdatei zur Verwendung der Zustandsprüfungen von Elastic-Load-Balancing konfigurieren. Im folgenden Beispiel wird die Gruppe so eingerichtet, dass sie zusätzlich zur EC2 Amazon-Statusprüfung die Integritätsprüfungen des Load Balancers verwendet, um den Zustand einer Instance zu ermitteln.

Example.ebextensions/autoscaling.config

```
Resources:
  AWSEBAutoScalingGroup:
    Type: "AWS::AutoScaling::AutoScalingGroup"
    Properties:
      HealthCheckType: ELB
      HealthCheckGracePeriod: 300
```

Weitere Informationen zu den `HealthCheckGracePeriod` Eigenschaften `HealthCheckType` und finden Sie unter AWS::AutoScaling::AutoScalingGroup im AWS CloudFormation Benutzerhandbuch und Health Checks for Auto Scaling Instances im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Standardmäßig ist die Elastic-Load-Balancing-Zustandsprüfung so konfiguriert, dass über Port 80 eine TCP-Verbindung zur Instance hergestellt wird. Dies bestätigt, dass der in der Instance ausgeführte Web-Server Verbindungen akzeptiert. Sie können jedoch auch die Load-Balancer-Zustandsprüfung anpassen, um sicherzustellen, dass sich Ihre Anwendung – und nicht nur der Webserver – in einem guten Zustand befindet. Mit der Einstellung für die Übergangsfrist wird in Sekunden festgelegt, wie lange die Zustandsprüfung einer Instance fehlschlagen darf, bevor diese beendet und ersetzt wird. Instances können, auch wenn sie aus der Lastenverteilung genommen wurden, wiederhergestellt werden, daher sollten Sie einen für die Anwendung angemessenen Zeitraum vorgeben.

Load Balancer Ihrer Elastic-Beanstalk-Umgebung

Ein Load Balancer verteilt den Datenverkehr auf die Instances Ihrer Umgebung. Wenn Sie Load Balancing aktivieren, AWS Elastic Beanstalk wird ein Elastic Load Balancing Load Balancer erstellt, der speziell auf Ihre Umgebung zugeschnitten ist. Elastic Beanstalk verwaltet diesen Load Balancer vollständig und kümmert sich um die Sicherheitseinstellungen und um die Terminierung des Load Balancers, wenn Sie Ihre Umgebung terminieren.

Alternativ können Sie einen Load Balancer für mehrere Elastic Beanstalk-Umgebungen freigeben. Mit einem freigegebenen Load Balancer sparen Sie Betriebskosten, da nicht für jede Umgebung ein dedizierter Load Balancer erstellt wird. Sie übernehmen auch einen größere Verantwortung für die Verwaltung des freigegebenen Load Balancer, den Ihre Umgebungen verwenden.

Elastic Load Balancing verfügt über diese Load Balancer-Typen:

- Classic Load Balancer – Der Load Balancer der vorherigen Generation. Leitet den Datenverkehr aus HTTP-, HTTPS- oder TCP-Anforderungen an verschiedene Ports auf Instances in der Umgebung weiter.
- Application Load Balancer – Ein Application-Layer Load Balancer. Leitet den Datenverkehr aus HTTP- oder HTTPS-Anforderungen auf Grundlage des Anforderungspfads an verschiedene Ports auf Instances in der Umgebung weiter.

- Network Load Balancer – Ein Network-Layer Load Balancer. Leitet Datenverkehr aus TCP-Anforderungen an verschiedene Ports auf Instances in der Umgebung weiter. Unterstützt aktive und passive Zustandsprüfungen.

Elastic Beanstalk unterstützt alle drei Load Balancer-Typen. Die folgende Tabelle zeigt, welche Typen Sie mit den beiden Nutzungsmustern verwenden können:

Load balancer type (Load-Balancer-Typ)	Dediziert	Freigegeben
Classic Load Balancer	✓ Ja	X Nein
Application Load Balancer	✓ Ja	✓ Ja
Network Load Balancer	✓ Ja	X Nein

> **Note**
>
> Die Option „Classic Load Balancer (CLB)“ ist im Konsolenassistenten Umgebung erstellen deaktiviert. Wenn Sie eine bestehende Umgebung mit einem Classic Load Balancer konfiguriert haben, können Sie eine neue erstellen, indem Sie die bestehende Umgebung entweder mit der Elastic-Beanstalk-Konsole oder der EB CLI klonen. Sie haben auch die Möglichkeit, die EB-CLI zu verwenden oder eine neue Umgebung AWS CLIzu erstellen, die mit einem Classic Load Balancer konfiguriert ist. Diese Befehlszeilentools erstellen eine neue Umgebung mit einem CLB, auch wenn in Ihrem Konto noch keiner vorhanden ist.

Standardmäßig erstellt Elastic Beanstalk einen Application Load Balancer für Ihre Umgebung, wenn Sie das Load-Balancing mit der Elastic Beanstalk-Konsole oder der EB CLI aktivieren. Der Load Balancer ist so konfiguriert, dass er den HTTP-Datenverkehr auf Port 80 überwacht und diesen Datenverkehr an die Instances auf demselben Port weiterleitet. Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen später ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern.

> **Note**
>
> Ihre Umgebung muss sich in einer VPC mit Subnetzen in mindestens zwei Availability Zones befinden, damit ein Application Load Balancer erstellt wird. Alle neuen AWS Konten enthalten Standardkonten VPCs , die diese Anforderung erfüllen.

In den folgenden Themen erhalten Sie weitere Informationen zu den einzelnen von Elastic Beanstalk unterstützten Load Balancer-Typen, zu ihrer Funktionalität und zu ihre Konfiguration und Verwaltung in einer Elastic Beanstalk-Umgebung. Sie erfahren auch, wie Sie einen Load Balancer für das Hochladen von Zugriffsprotokollen in Amazon S3 konfigurieren.

Themen

- Konfigurieren eines Classic Load Balancers
- Konfigurieren eines Application Load Balancer
- Gemeinsamen Application Load Balancer konfigurieren
- Konfigurieren eines Network Load Balancers
- Konfiguration von Elastic Beanstalk Beanstalk-Loadbalancern mit zwei Stacks
- Konfigurieren von Zugriffsprotokollen

Konfigurieren eines Classic Load Balancers

Wenn Sie Load Balancing aktivieren, ist Ihre AWS Elastic Beanstalk Umgebung mit einem Elastic Load Balancing Load Balancer ausgestattet, der den Traffic auf die Instances in Ihrer Umgebung verteilt. Elastic Load Balancing unterstützt verschiedene Load Balancer-Typen. Weitere Informationen dazu finden Sie im Elastic Load Balancing-Benutzerhandbuch. Elastic Beanstalk kann einen Load Balancer für Sie erstellen oder Sie einen von Ihnen erstellten freigegebenen Load Balancer angeben lassen.

Dieses Thema beschreibt die Konfiguration eines Classic Load Balancers, den Elastic Beanstalk erstellt und Ihrer Umgebung zuweist. Weitere Informationen zur Konfiguration aller Load Balancer-Typen, die Elastic Beanstalk unterstützt, finden Sie unter Load Balancer Ihrer Elastic-Beanstalk-Umgebung.

> **Note**
>
> Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen später ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern.

Einführung

Ein Classic Load Balancer ist der Elastic Load Balancer der vorherigen Generation. Er unterstützt das Weiterleiten des Datenverkehrs aus HTTP-, HTTPS- oder TCP-Anforderungen an verschiedene Ports auf Instances in der Umgebung.

Wenn Ihre Umgebung einen Classic Load Balancer verwendet, konfiguriert Elastic Beanstalk diesen standardmäßig auf listen für HTTP-Datenverkehr über Port 80 und leitet ihn an Instances am selben Port weiter. Sie können den Standard-Listener für Port 80 zwar nicht löschen, aber Sie können ihn deaktivieren. Dadurch wird dieselbe Funktionalität erreicht, indem der Datenverkehr blockiert wird. Beachten Sie, dass Sie andere Listener hinzufügen oder löschen können. Zur Unterstützung sicherer Verbindungen können Sie einen Listener auf Port 443 und ein TLS-Zertifikat für den Load Balancer konfigurieren.

Der Load Balancer verwendet eine Integritätsprüfung, um festzustellen, ob die EC2 Amazon-Instances, auf denen Ihre Anwendung ausgeführt wird, fehlerfrei sind. Die Zustandsprüfung sendet in festgelegten Intervallen eine Anfrage an eine angegebene URL. Wenn die URL eine Fehlermeldung zurückgibt oder nicht innerhalb eines definierten Timeouts antwortet, schlägt die Zustandsprüfung fehl.

Sofern sich die Anwendungsleistung verbessern lässt, wenn mehrere Anforderungen vom gleichen Client auf einem einzigen Server verarbeitet werden, können Sie den Load Balancer für die Nutzung von Sticky Sessions konfigurieren. Bei Sticky Sessions fügt der Load Balancer HTTP-Antworten ein Cookie hinzu, das die EC2 Amazon-Instance identifiziert, die die Anfrage bearbeitet hat. Bei der nächsten eingehenden Anforderung vom gleichen Client nutzt der Load Balancer das Cookie, um die Anforderung an dieselbe Instance zu senden.

Wenn zonenübergreifendes Load Balancing aktiviert ist, verteilt jeder Load Balancer-Knoten für Ihren Classic Load Balancer Anfragen gleichmäßig auf die registrierten Instances in allen aktivierten Availability Zones. Wenn zonenübergreifendes Load Balancing deaktiviert ist, verteilt jeder Load

Balancer-Knoten Anfragen gleichmäßig nur auf die registrierten Instances in seiner aktivierten Availability Zone.

Falls eine Instance aus dem Load Balancer entfernt wird (aufgrund ihres instabilen Zustands oder wegen einer Herunterskalierung der Umgebung), kann die Instance mithilfe von Connection Draining die Anforderungen abschließen, bevor die Verbindung zwischen der Instance und dem Load Balancer getrennt wird. Sie können den Zeitraum ändern, in dem Instances eine Antwort senden können, oder Connection Draining vollständig deaktivieren.

> **Note**
>
> Connection Draining wird standardmäßig aktiviert, wenn Sie eine Umgebung mit der Elastic Beanstalk-Konsole oder der EB CLI erstellen. Bei anderen Clients können Sie die Aktivierung in den Konfigurationsoptionen vornehmen.

Sie können die erweiterten Load Balancer-Einstellungen verwenden, um Listener an beliebigen Ports zu konfigurieren, zusätzliche Sticky-Session-Einstellungen zu ändern und den Load Balancer so zu konfigurieren, dass er sich sicher mit Instances verbindet. EC2 Diese Einstellungen stehen über Konfigurationsoptionen zur Verfügung, die Sie mithilfe von Konfigurationsdateien im Quellcode oder über die Elastic Beanstalk-API direkt in der Umgebung festlegen können. Viele dieser Einstellungen sind auch in der Elastic Beanstalk-Konsole verfügbar. Außerdem können Sie den Load Balancer für den Upload von Zugriffsprotokollen in Amazon S3 konfigurieren.

Classic Load Balancer mit der Elastic Beanstalk-Konsole konfigurieren

Sie können die Elastic Beanstalk-Konsole zum Konfigurieren der Ports, des HTTPS-Zertifikats und anderer Einstellungen eines Classic Load Balancer verwenden, während die Umgebung erstellt wird oder später, wenn die Umgebung bereits ausgeführt wird.

> **Note**
>
> Die Option „Classic Load Balancer (CLB)“ ist im Konsolenassistenten Umgebung erstellen deaktiviert. Wenn Sie eine bestehende Umgebung mit einem Classic Load Balancer konfiguriert haben, können Sie eine neue erstellen, indem Sie die bestehende Umgebung entweder mit der Elastic-Beanstalk-Konsole oder der EB CLI klonen. Sie haben auch die Möglichkeit, die EB-CLI zu verwenden oder eine neue Umgebung AWS CLIzu erstellen, die

mit einem Classic Load Balancer konfiguriert ist. Diese Befehlszeilentools erstellen eine neue Umgebung mit einem CLB, auch wenn in Ihrem Konto noch keiner vorhanden ist.

So konfigurieren Sie den Classic Load Balancer einer laufenden Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.

   > **Note**
   >
   > Ist in der Konfigurationskategorie Load balancer die Schaltfläche Edit (Bearbeiten) nicht vorhanden, verfügt Ihre Umgebung über keinen Load Balancer. Informationen zur Einrichtung finden Sie unter Ändern des Umgebungstyps.

5. Nehmen Sie die für Ihre Umgebung erforderlichen Konfigurationsänderungen am Classic Load Balancer vor.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Classic Load Balancer-Einstellungen

- Listener
- Sitzungen
- Zonenübergreifendes Load Balancing
- Connection Draining
- Gesundheitscheck

Listener

Verwenden Sie diese Liste, um Listener für Ihren Load Balancer anzugeben. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports mittels eines festgelegten Protokolls an Ihre Instances weiter. Anfänglich wird in der Liste der Standard-Listener angezeigt, der eingehenden HTTP-Datenverkehr auf Port 80 an Instance-Server Ihrer Umgebung weiterleitet, die HTTP-Port 80 überwachen.

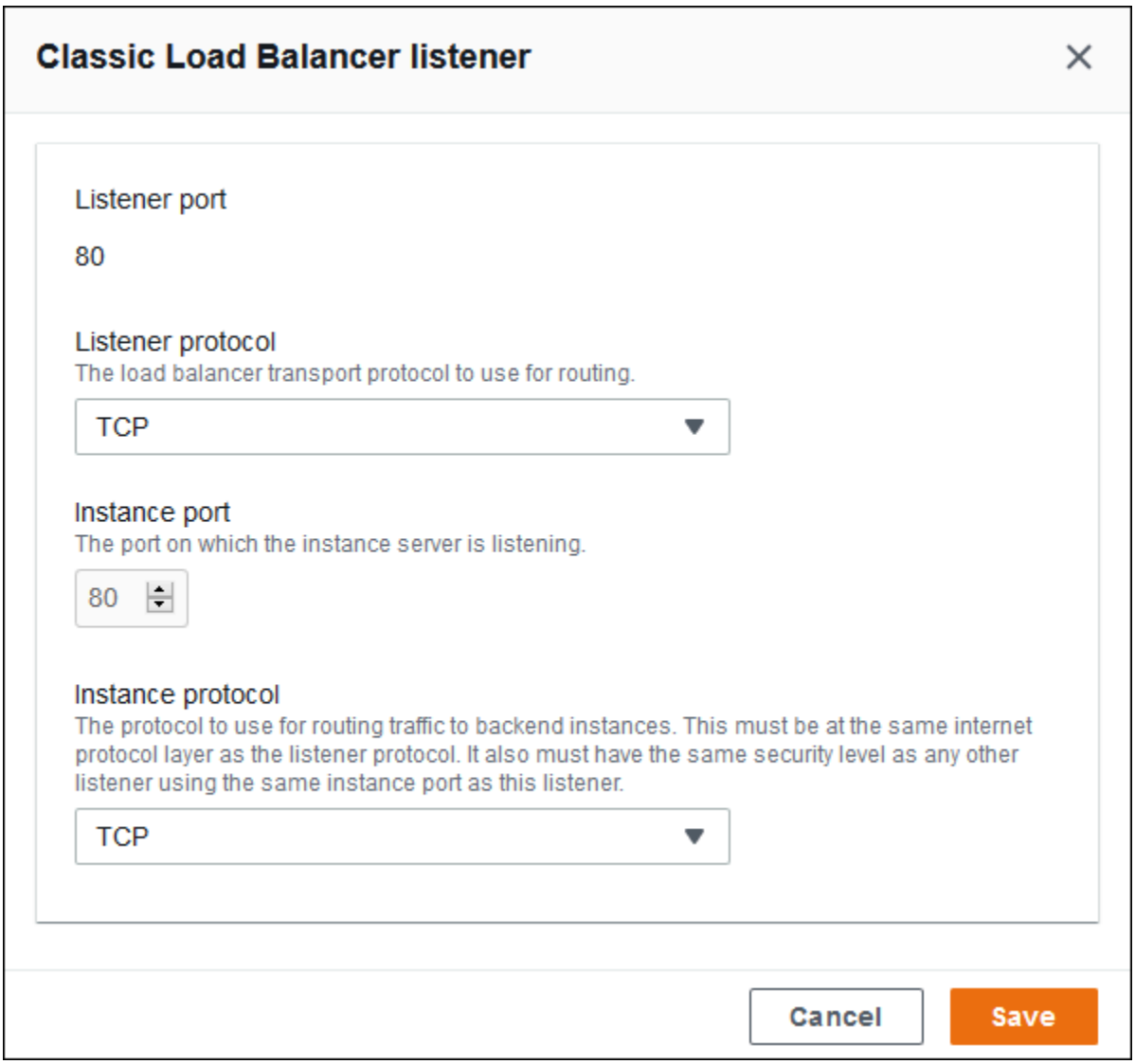
> **Note**
>
> Sie können den Standard-Listener für Port 80 zwar nicht löschen, aber Sie können ihn deaktivieren. Dadurch wird dieselbe Funktionalität erreicht, indem der Datenverkehr blockiert wird.



So konfigurieren Sie einen vorhandenen Listener

1. Aktivieren Sie das Kontrollkästchen neben dem Tabelleneintrag und wählen Sie Actions (Aktionen) und dann die gewünschte Aktion aus.
2. Falls Sie Edit (Bearbeiten) ausgewählt haben, verwenden Sie das Dialogfeld Classic Load Balancer listener (Listener Ihres Classic Load Balancers), um die Einstellungen zu bearbeiten, und wählen Sie dann Save (Speichern) aus.

Beispielsweise können Sie den Standard-Listener bearbeiten und die Einstellung Protocol (Protokoll) von HTTP in TCP ändern, wenn der Load Balancer die Anforderungen unverändert weiterleiten soll. Dadurch wird verhindert, dass der Load Balancer Header umschreibt (einschließlich `X-Forwarded-For`). Die Technik funktioniert nicht mit Sticky Sessions.



So fügen Sie einen Listener hinzu

1. Wählen Sie Add listener (Listener hinzufügen) aus.
2. Konfigurieren Sie im Dialogfeld Classic Load Balancer listener (Listener Ihres Classic Load Balancers) die gewünschten Einstellungen und wählen Sie dann Add (Hinzufügen) aus.

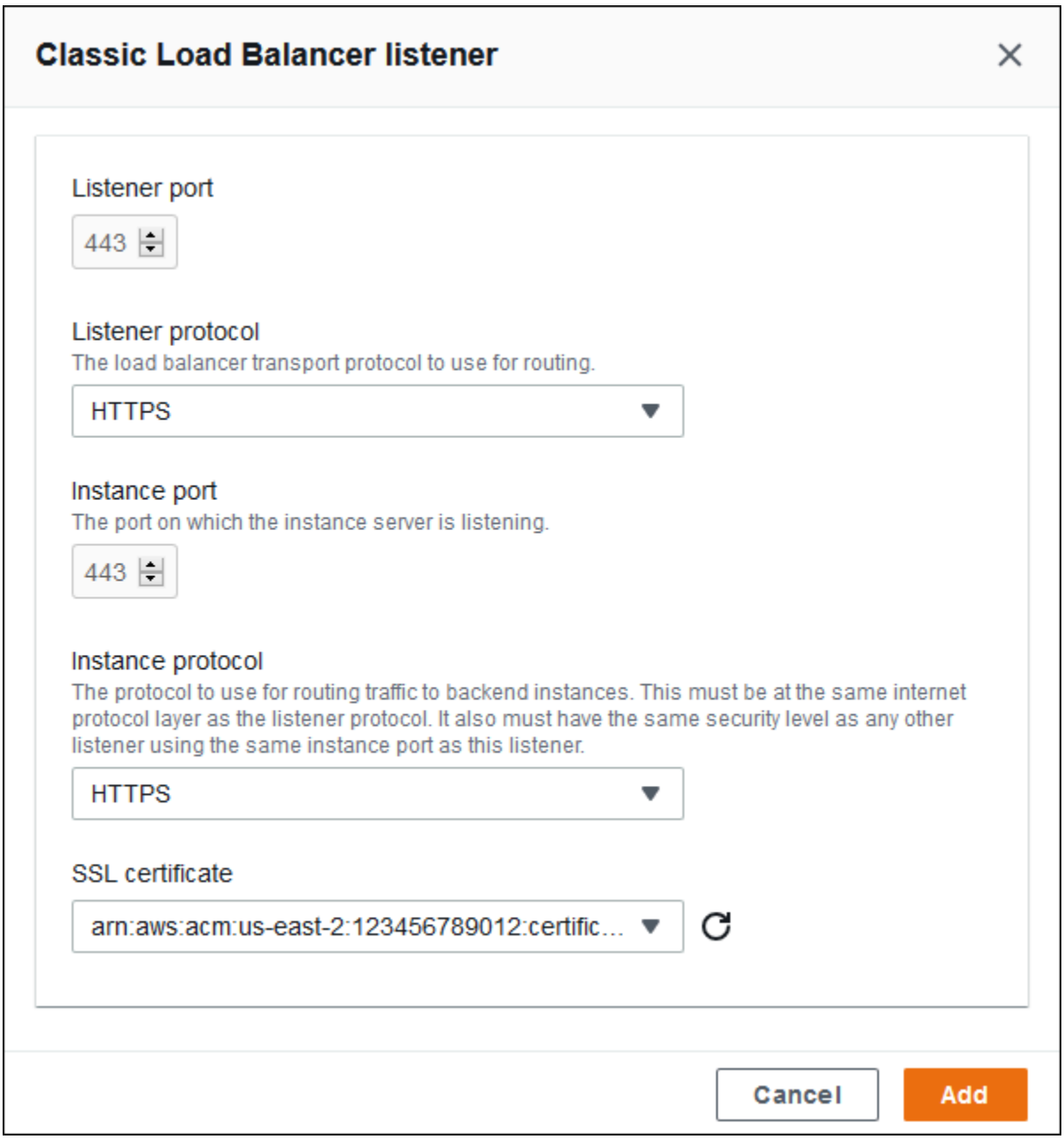
Ein häufiger Anwendungsfall ist das Hinzufügen eines sicheren Listener. Im Beispiel im folgenden Image wird ein Listener für HTTPS-Datenverkehr auf Port 443 hinzugefügt. Dieser Listener leitet den eingehenden Datenverkehr an Instance-Server der Umgebung weiter, die Port 443 auf HTTPS-Datenverkehr überwachen.

Bevor Sie einen HTTPS-Listener konfigurieren können, stellen Sie sicher, dass Sie über ein gültiges SSL-Zertifikat verfügen. Führen Sie eine der folgenden Aktionen aus:

- Wenn AWS Certificate Manager (ACM) in Ihrer AWS Region verfügbar ist, erstellen oder importieren Sie ein Zertifikat mit ACM. Weitere Informationen zum Anfordern eines ACM-Zertifikats finden Sie unter Zertifikat anfordern im AWS Certificate Manager -Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in ACM finden Sie unter Importieren von Zertifikaten im AWS Certificate Manager -Benutzerhandbuch.
- Wenn ACM in Ihrer AWS Region nicht verfügbar ist, laden Sie Ihr vorhandenes Zertifikat und Ihren Schlüssel in IAM hoch. Weitere Informationen zum Erstellen und Hochladen von Zertifikaten in IAM finden Sie unter Arbeiten mit Serverzertifikaten im IAM-Benutzerhandbuch.

Weitere Informationen zum Konfigurieren von HTTPS und zum Verwenden von Zertifikaten in Elastic Beanstalk finden Sie unter Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung.

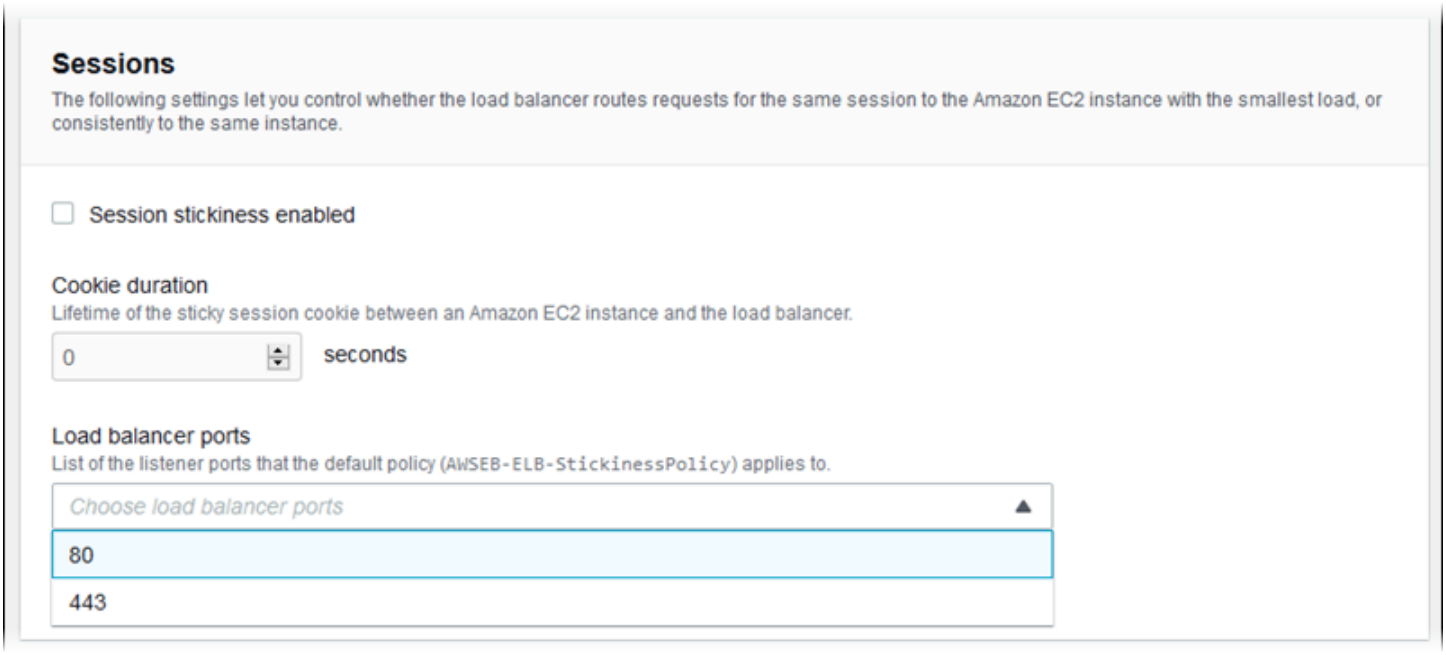
Wählen Sie unter SSL certificate (SSL-Zertifikat) den ARN des SSL-Zertifikats aus. Beispiel: `arn:aws:iam::123456789012:server-certificate/abc/certs/build` oder `arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`.



Informationen zum Konfigurieren von HTTPS und zum Verwenden von Zertifikaten in Elastic Beanstalk finden Sie unter Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung.

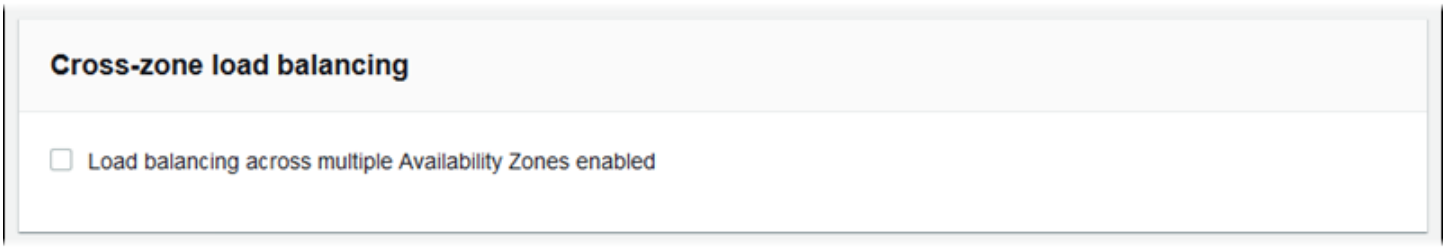
Sitzungen

Aktivieren oder deaktivieren Sie das Kontrollkästchen Session stickiness enabled (Gebundene Sitzung aktiviert) zum Aktivieren oder Deaktivieren von Sticky Sessions. Verwenden Sie Cookie duration (Cookie-Dauer) zum Konfigurieren der Dauer einer Sticky Session (auf bis zu **`1000000`** Sekunden). Wählen Sie in der Liste Load balancer ports (Load Balancer-Ports) Listener-Ports aus, für die die Standardrichtlinie (`AWSEB-ELB-StickinessPolicy`) gilt.



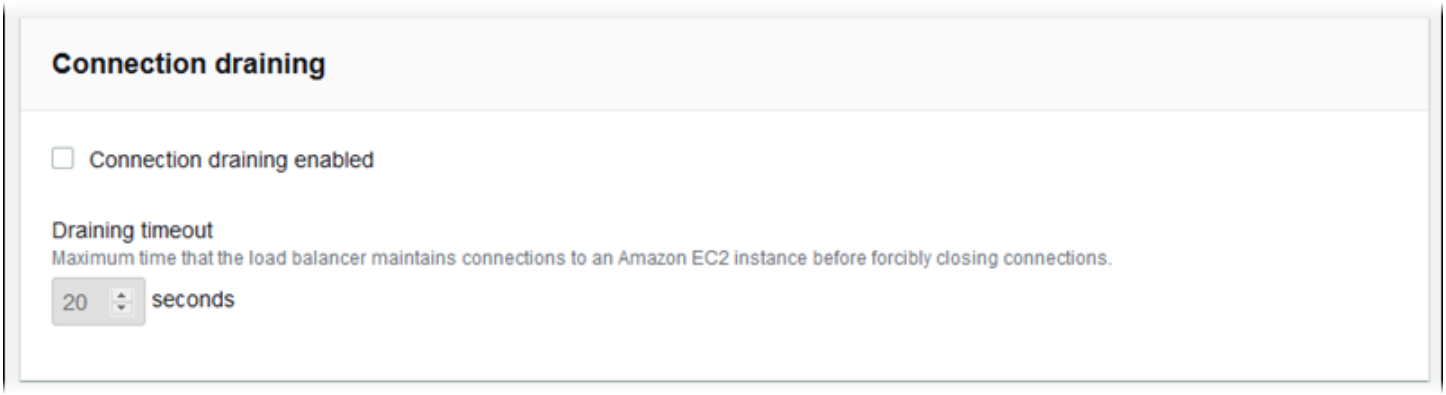
Zonenübergreifendes Load Balancing

Aktivieren oder deaktivieren Sie das Kontrollkästchen Load balancing across multiple Availability Zones enabled (Das Load Balancing ist für mehrere Availability Zones aktiviert) zum Aktivieren oder Deaktivieren von zonenübergreifendem Load Balancing.



Connection Draining

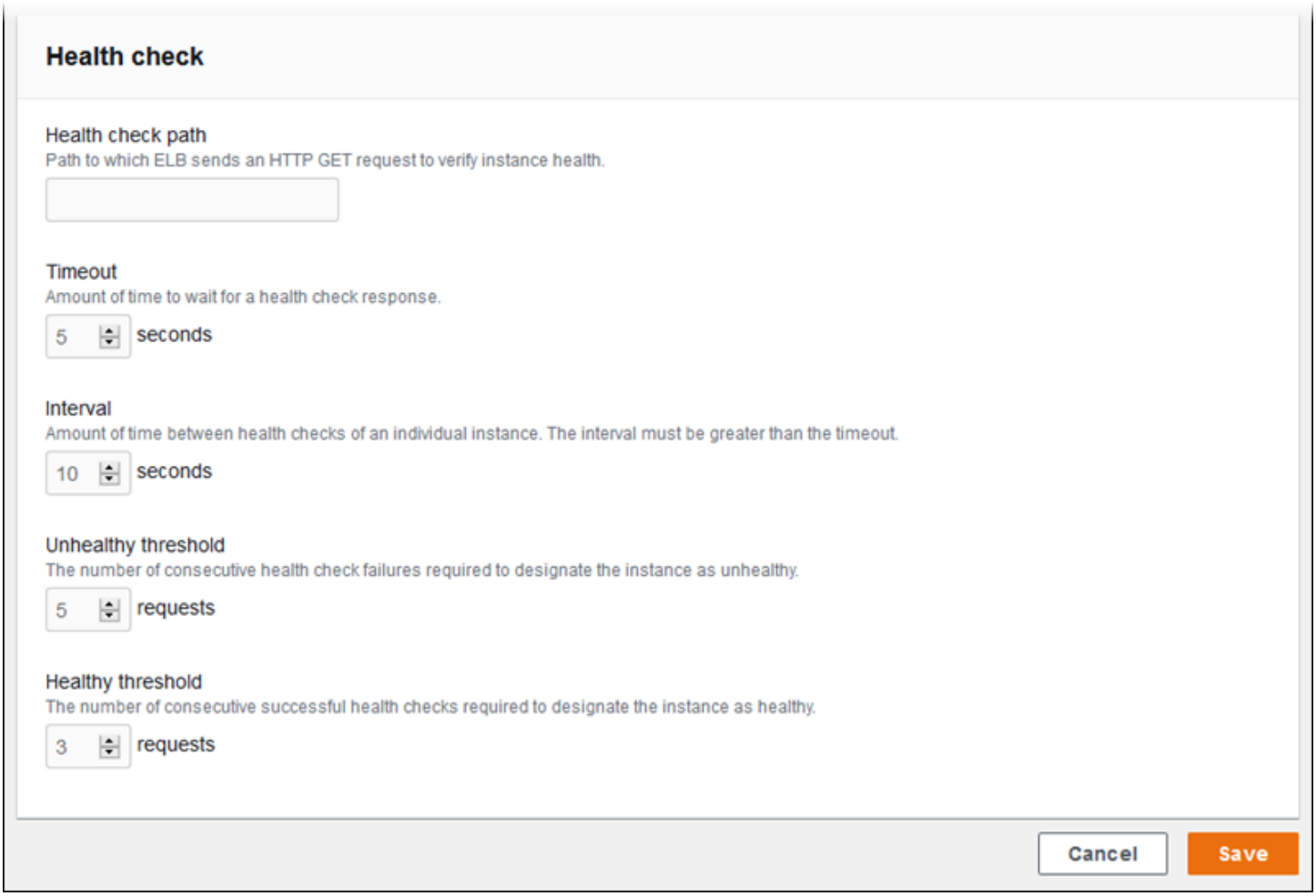
Aktivieren oder deaktivieren Sie das Kontrollkästchen Connection draining enabled (Connection Draining aktiviert) zum Aktivieren oder Deaktivieren von Connection Draining. Legen Sie das Draining timeout (Draining-Timeout) fest (auf bis zu **`3600`** Sekunden).



Gesundheitscheck

Verwenden Sie die folgenden Einstellungen zum Konfigurieren von Zustandsprüfungen des Load Balancer:

- Health check path (Pfad für die Zustandsprüfung) – Der Pfad, an den der Load Balancer Zustandsprüfungsanforderungen sendet Falls Sie den Pfad nicht festlegen, versucht der Load Balancer, über Port 80 eine TCP-Verbindung herzustellen, um die Zustandsprüfung auszuführen.
- Timeout – Die Wartezeit in Sekunden, bis eine Antwort von der Zustandsprüfung eingeht
- Interval (Intervall) – Der Zeitraum in Sekunden zwischen Zustandsprüfungen einer einzelnen Instance Der Intervall-Wert muss größer als der Timeout-Wert sein.
- Unhealthy threshold, Healthy threshold – Die Anzahl der Zustandsprüfungen, die nicht bestanden bzw. bestanden sein müssen, bevor Elastic Load Balancing den Zustand der Instance ändert.



> **Note**
>
> Die Zustandsprüfung für Elastic Load Balancing hat keinen Einfluss auf das Verhalten der Zustandsprüfung der Auto Scaling-Gruppe einer Umgebung. Instances, die eine Elastic Load Balancing Balancing-Zustandsprüfung nicht bestehen, werden nicht automatisch durch Amazon EC2 Auto Scaling ersetzt, es sei denn, Sie konfigurieren Amazon EC2 Auto Scaling manuell entsprechend. Details dazu finden Sie unter Einstellung für die Auto Scaling Scaling-Integritätsprüfung für Ihre Elastic Beanstalk Beanstalk-Umgebung.

Weitere Informationen zu Zustandsprüfungen und deren Auswirkungen auf den Gesamtstatus der Umgebung finden Sie unter Grundlegende Zustandsberichte.

Classic Load Balancer mit der EB CLI konfigurieren

Sie werden von der EB CLI aufgefordert, einen Load Balancer-Typ auszuwählen, wenn Sie eb create ausführen.

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 1):
```

Drücken Sie die Eingabetaste, um `classic` auszuwählen.

Sie können einen Load Balancer-Typ auch mithilfe der `--elb-type`-Option angeben.

```
$ eb create test-env --elb-type classic
```

Classic Load Balancer-Konfiguration in Namespaces

In den folgenden Namespaces finden Sie Einstellungen im Zusammenhang mit Classic Load Balancern:

- `aws:elb:healthcheck` – Konfigurieren Sie Grenzwerte und überprüfen Sie Intervall sowie Timeout für Load Balancer-Zustandsprüfungen.
- `aws:elasticbeanstalk:application` – Konfigurieren Sie die Zustandsprüfungs-URL.
- `aws:elb:loadbalancer` – Aktivieren Sie zonenübergreifendes Load Balancing. Ordnen Sie dem Load Balancer Sicherheitsgruppen zu und überschreiben Sie die von Elastic Beanstalk erstellte Standardsicherheitsgruppe. In diesem Namespace finden Sie auch veraltete Optionen für die Konfiguration von Standard- und sicheren Listenern, die durch Optionen im Namespace `aws:elb:listener` ersetzt wurden.
- `aws:elb:listener` – Konfigurieren Sie den Standard-Listener auf Port 80, einen sicheren Listener auf Port 443 oder weitere Listener für beliebige Protokolle auf einem frei wählbaren

Port. Sofern Sie `aws:elb:listener` als Namespace angeben, gelten die Einstellungen für den Standard-Listener auf Port 80. Wenn Sie einen Port festlegen (z. B. `aws:elb:listener:443`), wird ein Listener auf diesem Port konfiguriert.

- `aws:elb:policies` – Konfigurieren Sie zusätzliche Einstellungen für den Load Balancer. Verwenden Sie die Optionen in diesem Namespace, um Listener auf beliebigen Ports zu konfigurieren, zusätzliche Sticky-Session-Einstellungen zu ändern und den Load Balancer so zu konfigurieren, dass er sich sicher mit EC2 Amazon-Instances verbindet.

Die EB-CLI und Elastic-Beanstalk-Konsole wenden die für die vorhergehenden Optionen empfohlenen Werte an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter Empfohlene Werte.

Example.ebextensions/loadbalancer-terminatehttps.config

Die folgende Beispielkonfigurationsdatei wird zur Erstellung eines HTTPS-Listeners auf Port 443, für die Zuweisung eines Zertifikats, mit dem der Load Balancer die sichere Verbindung beendet, und zur Deaktivierung des Standard-Listeners auf Port 80 verwendet. Der Load Balancer leitet die entschlüsselten Anfragen auf HTTP:80 an die EC2 Instances in Ihrer Umgebung weiter.

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: HTTPS
    SSLCertificateId: arn:aws:acm:us-
east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678
    InstancePort: 80
    InstanceProtocol: HTTP
  aws:elb:listener:
    ListenerEnabled: false
```

Konfigurieren eines Application Load Balancer

Wenn Sie Load Balancing aktivieren, ist Ihre AWS Elastic Beanstalk Umgebung mit einem Elastic Load Balancing Load Balancer ausgestattet, der den Traffic auf die Instances in Ihrer Umgebung verteilt. Elastic Load Balancing unterstützt verschiedene Load Balancer-Typen. Weitere Informationen dazu finden Sie im Elastic Load Balancing-Benutzerhandbuch. Elastic Beanstalk kann einen Load Balancer für Sie erstellen oder Sie einen von Ihnen erstellten freigegebenen Load Balancer angeben lassen.

Dieses Thema beschreibt die Konfiguration eines Application Load Balancers, den Elastic Beanstalk erstellt und Ihrer Umgebung zuweist. Siehe auch the section called “Geteile Application Load Balancer”. Weitere Informationen zur Konfiguration aller Load Balancer-Typen, die Elastic Beanstalk unterstützt, finden Sie unter the section called “Load Balancer”.

> **Note**
>
> Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern. Sie können auch nicht von einem dedizierten zu einem freigegebenen Load Balancer wechseln und umgekehrt.

Einführung

Ein Application Load Balancer untersucht Datenverkehr auf Protokollebene des Anwendungsnetzwerks, um den Pfad der Anforderung zu bestimmen, sodass er Anforderungen für verschiedene Pfade an verschiedene Ziele leiten kann.

Wenn Ihre Umgebung einen Application Load Balancer verwendet, konfiguriert Elastic Beanstalk diesen standardmäßig so, dass er die gleiche Funktion wie ein Classic Load Balancer ausführt. Der standardmäßige Listener akzeptiert HTTP-Anfragen auf Port 80 und leitet sie an die Instances in Ihrer Umgebung weiter. Sie können einen sicheren Listener auf Port 443 mit einem Zertifikat zur Verschlüsselung von HTTPS-Datenverkehr hinzufügen, Zustandsprüfungsverhalten konfigurieren und Zugriffsprotokolle vom Load Balancer an einen Amazon Simple Storage Service (Amazon S3)-Bucket übermitteln.

> **Note**
>
> Im Gegensatz zu einem Classic Load Balancer oder einem Network Load Balancer kann ein Application Load Balancer weder über TCP noch über Listener der Transportschicht (Schicht 4) verfügen. SSL/TLS Er unterstützt nur HTTP- und HTTPS-Listener. Darüber hinaus kann er keine Backend-Authentifizierung zum Authentifizieren von HTTPS-Verbindungen zwischen dem Load Balancer und Backend-Instances verwenden.

In einer Elastic Beanstalk-Umgebung können Sie einen Application Load Balancer verwenden, um Datenverkehr für bestimmte Pfade zu einem anderen Prozess auf Ihren Webserver-Instance zu routen. Mit einem Classic Load Balancer wird der gesamte Datenverkehr zu einem Listener an einen einzelnen Port auf den Backend-Instances weitergeleitet. Mit einem Application Load Balancer können Sie mehrere Regeln auf dem Listener so konfigurieren, dass Anforderungen für bestimmte Pfade an verschiedene Backend-Prozesse weitergeleitet werden. Sie konfigurieren jeden Prozess mit dem Port, den der Prozess überwacht.

Sie können z. B. eine Anmeldung separat von der Hauptanwendung durchführen. Während die Hauptanwendung auf den Instances Ihrer Umgebung die meisten Anforderungen akzeptiert und auf Port 80 überwacht, überwacht Ihr Anmeldevorgang auf Port 5000 und akzeptiert Anforderungen an den `/login`-Pfad. Alle eingehenden Anforderungen von Clients gehen auf Port 80 ein. Mit einem Application Load Balancer können Sie einen einzigen Listener für eingehenden Datenverkehr auf Port 80 mit zwei Regeln konfigurieren, die den Datenverkehr je nach Pfad in der Anforderung an zwei separate Prozesse weiterleiten. Sie fügen eine benutzerdefinierte Regel hinzu, die den Datenverkehr für `/login` an den Anmeldeprozess weiterleitet, der Port 5000 überwacht. Die Standardregel leitet sämtlichen weiteren Datenverkehr an die Prozessüberwachung für die Hauptanwendung an Port 80 weiter.

Eine Application Load Balancer-Regel ordnet eine Anfrage einer Zielgruppe zu. In Elastic Beanstalk wird eine Zielgruppe durch einen Prozess dargestellt. Sie können einen Prozess mit Einstellungen für Protokoll, Port und Zustandsprüfung konfigurieren. Der Prozess stellt den Prozess dar, der auf Instances in Ihrer Umgebung ausgeführt wird. Der Standardprozess ist ein Listener auf Port 80 des Reverse Proxy (nginx oder Apache), der vor Ihrer Anwendung ausgeführt wird.

> **Note**
>
> Außerhalb von Elastic Beanstalk entspricht eine Zielgruppe einer Gruppe von Instances. Ein Listener kann Regeln und Zielgruppen für die Weiterleitung des Datenverkehrs an verschiedene Instances basierend auf dem Pfad verwenden. Innerhalb von Elastic Beanstalk sind alle Instances in Ihrer Umgebung identisch, sodass die Unterscheidung zwischen Prozessen gemacht wird, die auf verschiedenen Ports überwachen.

Ein Classic Load Balancer verwendet eine einzige Zustandsprüfungspfad für die gesamte Umgebung. Bei einem Application Load Balancer verfügt jeder Prozess über einen separaten Zustandsprüfungpfad, der durch den Load Balancer und der von Elastic Beanstalk unterstützte Zustandsprüfung überwacht wird.

Zur Verwendung Ihres Application Load Balancer muss Ihre Umgebung in einer Standard- oder benutzerdefinierten VPC sein und über eine Servicerolle mit dem Standardsatz an Berechtigungen verfügen. Wenn Sie über eine ältere Servicerolle verfügen, müssen Sie ggf. ihre Berechtigungen aktualisieren, um `elasticloadbalancing:DescribeTargetHealth` und `elasticloadbalancing:DescribeLoadBalancers` einzuschließen. Weitere Informationen über Application Load Balancer finden Sie unter What Is an Application Load Balancer?.

> **Note**
>
> Die Zustandsprüfung bei Application Load Balancer verwendet nicht den Zustandsprüfungspfad für Elastic Beanstalk. Stattdessen wird der spezifische Pfad verwendet, der für jeden Prozess separat konfiguriert ist.

Application Load Balancer mit der Elastic Beanstalk-Konsole konfigurieren

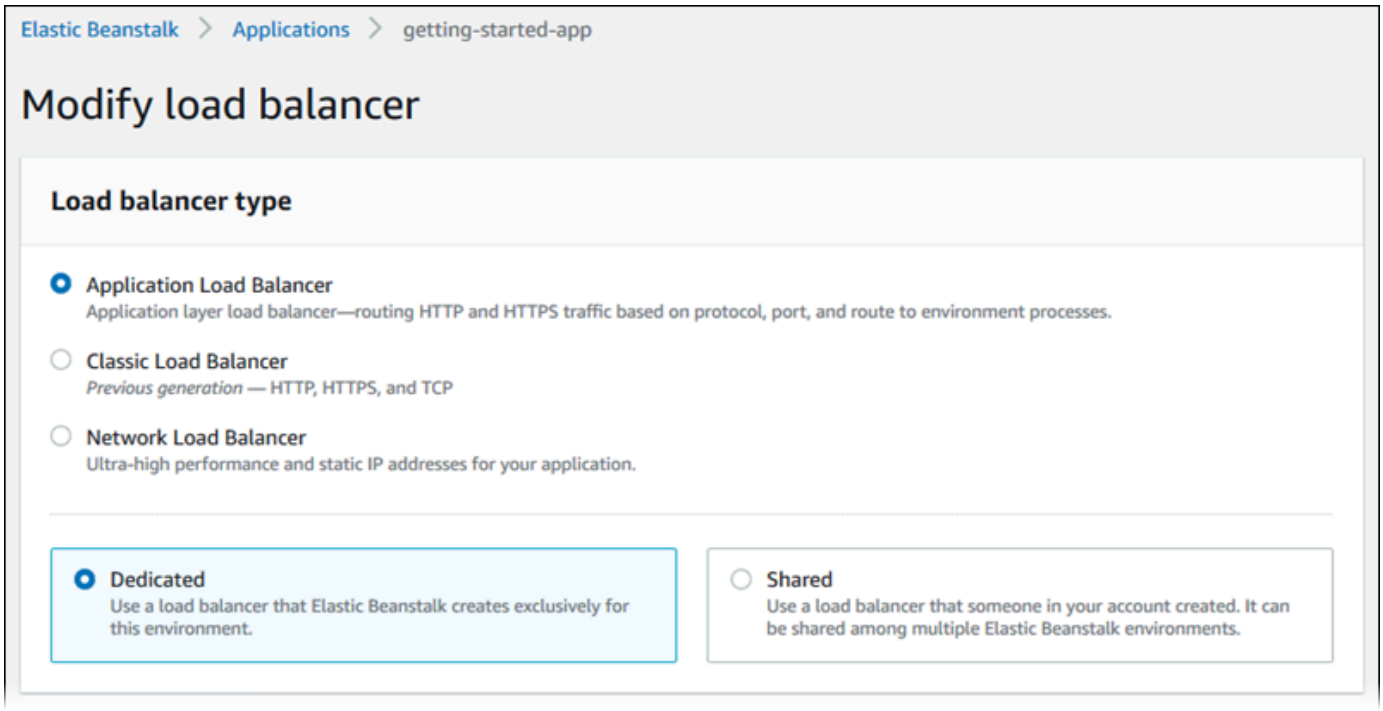
Sie können die Elastic Beanstalk-Konsole verwenden, um die Listener, Prozesse und Regeln eines Application Load Balancers während der Erstellung der Umgebung oder später (wenn Ihre Umgebung bereits ausgeführt wird) zu konfigurieren.

So konfigurieren Sie einen Application Load Balancer in der Elastic Beanstalk-Konsole während der Erstellung der Umgebung:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus.
3. Wählen Sie Create a new environment (Neue Umgebung erstellen), um mit der Erstellung Ihrer Umgebung zu beginnen.
4. Wählen Sie auf der Hauptseite des Assistenten zuerst Configure more options (Weitere Optionen konfigurieren) und dann Create environment (Umgebung erstellen) aus.
5. Wählen Sie die Konfigurationsvoreinstellung High availability (Hohe Verfügbarkeit) aus.

   Alternativ können Sie in der Konfigurationskategorie Capacity (Kapazität) einen Umgebungstyp mit Load balanced (Lastenverteilung) konfigurieren. Details hierzu finden Sie unter Capacity (Kapazität).
6. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.

7. Wählen Sie die Optionen Application Load Balancer und Dedicated (Dediziert) aus, falls sie nicht bereits ausgewählt sind.



8. Nehmen Sie alle Konfigurationsänderungen am Application Load Balancer vor, die für Ihre Umgebung erforderlich sind.
9. Wählen Sie Save (Speichern) aus und nehmen Sie dann alle weiteren Konfigurationsänderungen für Ihre Umgebung vor.
10. Wählen Sie Create environment (Umgebung erstellen) aus.

So konfigurieren Sie den Application Load Balancer einer laufenden Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.

> **Note**
>
> Ist in der Konfigurationskategorie Load balancer die Schaltfläche Edit (Bearbeiten) nicht vorhanden, verfügt Ihre Umgebung über keinen Load Balancer. Informationen zur Einrichtung finden Sie unter Ändern des Umgebungstyps.

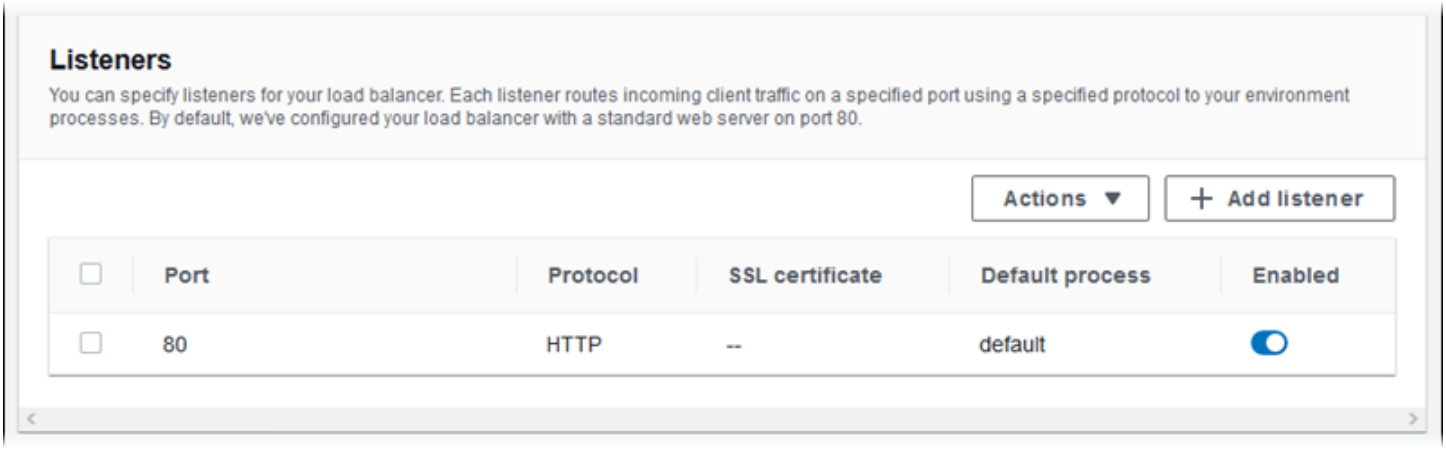
5. Nehmen Sie die Konfigurationsänderungen am Application Load Balancer vor, die Ihre Umgebung erfordert.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Einstellungen für Application Load Balancer

- Listener
- Prozesse
- Regeln
- Zugangsprotokollerfassung

Listener

Verwenden Sie diese Liste, um Listener für Ihren Load Balancer anzugeben. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports mittels eines festgelegten Protokolls an mindestens einen Prozess auf Ihren Instances weiter. Zunächst wird in der Liste der Standard-Listener angezeigt, der den eingehenden HTTP-Datenverkehr auf Port 80 an einen Prozess mit der Bezeichnung default (Standard) weiterleitet.



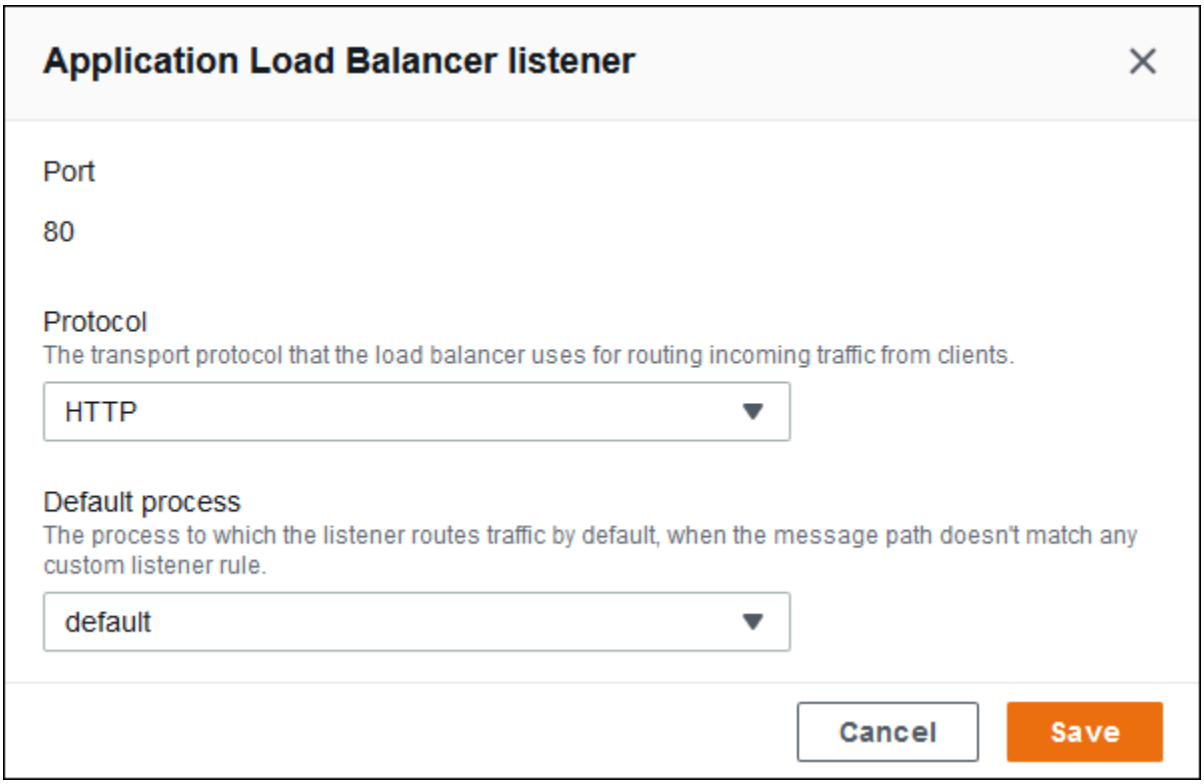
So konfigurieren Sie einen vorhandenen Listener

1. Aktivieren Sie das Kontrollkästchen neben seinem Tabelleneintrag und wählen Sie dann Actions (Aktionen), Edit (Bearbeiten) aus.
2. Verwenden Sie das Dialogfeld Application Load Balancer listener, um die Einstellungen zu bearbeiten, und wählen Sie dann Save (Speichern) aus.

So fügen Sie einen Listener hinzu

1. Wählen Sie Add listener (Listener hinzufügen) aus.
2. Konfigurieren Sie im Dialogfeld Application Load Balancer listener (Application Load Balancer-Listener) die gewünschten Einstellungen. Wählen Sie anschließend Add (Hinzufügen) aus.

Verwenden Sie die Dialogfeldeinstellungen unter Application Load Balancer listener (Application Load Balancer-Listener), um den Port und das Protokoll auszuwählen, auf dem der Listener den Datenverkehr überwacht, sowie den Prozess, an den der Datenverkehr weitergeleitet werden soll. Bei Auswahl des HTTPS-Protokolls konfigurieren Sie die SSL-Einstellungen.



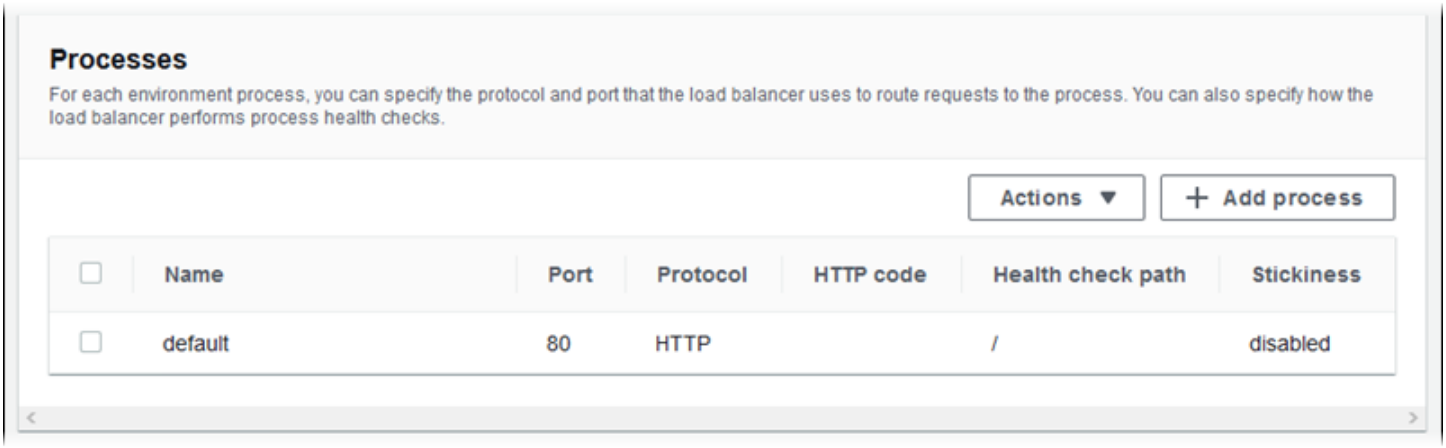
Bevor Sie einen HTTPS-Listener konfigurieren können, stellen Sie sicher, dass Sie über ein gültiges SSL-Zertifikat verfügen. Führen Sie eine der folgenden Aktionen aus:

- Wenn AWS Certificate Manager (ACM) in Ihrer AWS Region verfügbar ist, erstellen oder importieren Sie ein Zertifikat mit ACM. Weitere Informationen zum Anfordern eines ACM-Zertifikats finden Sie unter Zertifikat anfordern im AWS Certificate Manager -Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in ACM finden Sie unter Importieren von Zertifikaten im AWS Certificate Manager -Benutzerhandbuch.
- Wenn ACM in Ihrer AWS Region nicht verfügbar ist, laden Sie Ihr vorhandenes Zertifikat und Ihren Schlüssel in IAM hoch. Weitere Informationen zum Erstellen und Hochladen von Zertifikaten in IAM finden Sie unter Arbeiten mit Serverzertifikaten im IAM-Benutzerhandbuch.

Weitere Informationen zum Konfigurieren von HTTPS und zum Verwenden von Zertifikaten in Elastic Beanstalk finden Sie unter Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung.

Prozesse

Verwenden Sie diese Liste, um Prozesse für Ihren Load Balancer anzugeben. Ein Prozess ist ein Ziel für Listener, an das Datenverkehr weitergeleitet wird. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports mittels eines festgelegten Protokolls an mindestens einen Prozess auf Ihren Instances weiter. Anfänglich zeigt die Liste den Standardprozess, der eingehenden HTTP-Datenverkehr auf Port 80 überwacht.



Sie können die Einstellungen eines vorhandenen Prozesses bearbeiten oder einen neuen Prozess hinzufügen. Um mit dem Bearbeiten eines Prozesses auf der Liste zu beginnen oder Prozesse zur Liste hinzuzufügen, verwenden Sie dieselben Schritte, die für die Liste der Listener aufgelistet sind. Das Dialogfeld Environment process (Umgebungsprozess) wird geöffnet.

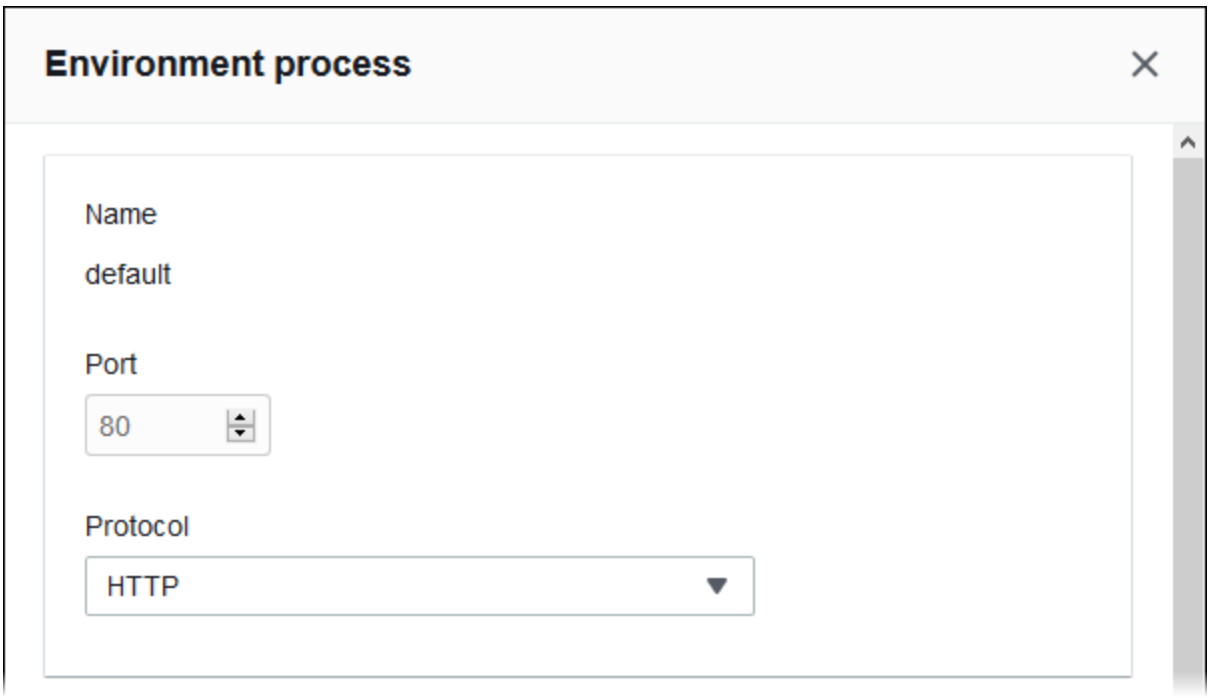
Prozesseinstellungen im Dialogfeld für die Umgebung des Application Load Balancers

- Definition

- Gesundheitscheck
- Sitzungen

Definition

Verwenden Sie diese Einstellungen zum Definieren des Prozesses: Name, Port und Protocol (Protokoll) für die Überwachung auf Anforderungen.

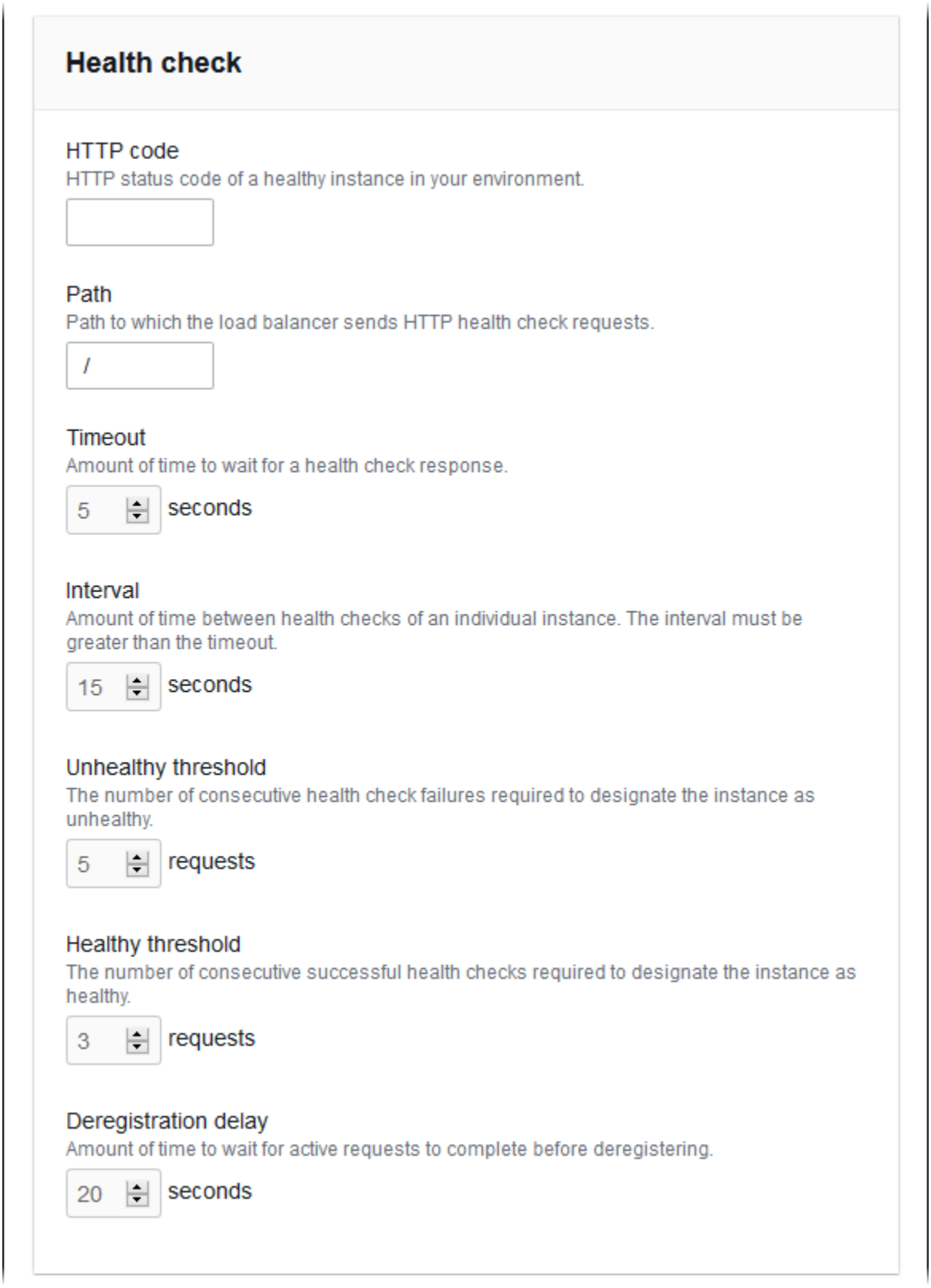


Gesundheitscheck

Verwenden Sie die folgenden Einstellungen zum Konfigurieren von Zustandsprüfungen für den Prozess:

- HTTP code (HTTP-Code) – Der HTTP-Statuscode, der einen funktionsfähigen Prozess bestimmt
- Path (Pfad) – Der Anforderungspfad der Zustandsprüfung für den Prozess
- Timeout – Die Wartezeit in Sekunden, bis eine Antwort von der Zustandsprüfung eingeht
- Interval (Intervall) – Der Zeitraum in Sekunden zwischen Zustandsprüfungen einer einzelnen Instance Der Intervall-Wert muss größer als der Timeout-Wert sein.
- Unhealthy threshold, Healthy threshold – Die Anzahl der Zustandsprüfungen, die nicht bestanden bzw. bestanden sein müssen, bevor Elastic Load Balancing den Zustand der Instance ändert.

- Deregistration delay (Verzögerung der Registrierungsaufhebung) – Die Zeit in Sekunden, die gewartet werden muss, bis aktive Anforderungen abgeschlossen sind, ehe eine Instance abgemeldet wird

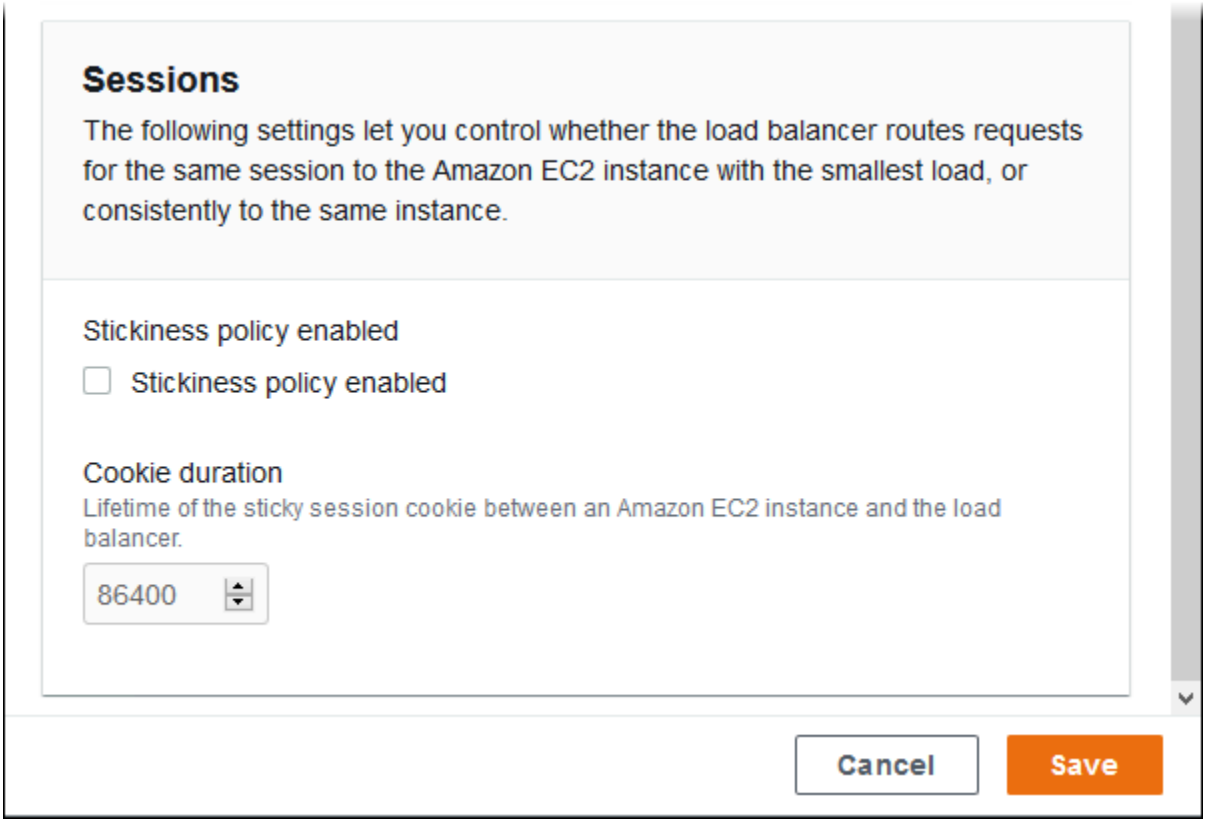

Health check
HTTP code
HTTP status code of a healthy instance in your environment.
Path
Path to which the load balancer sends HTTP health check requests.
/
Timeout
Amount of time to wait for a health check response.
5
seconds
Interval
Amount of time between health checks of an individual instance. The interval must be greater than the timeout.
15
seconds
Unhealthy threshold
The number of consecutive health check failures required to designate the instance as unhealthy.
5
requests
Healthy threshold
The number of consecutive successful health checks required to designate the instance as healthy.
3
requests
Deregistration delay
Amount of time to wait for active requests to complete before deregistering.
20
seconds

> **Note**
>
> Die Zustandsprüfung für Elastic Load Balancing hat keinen Einfluss auf das Verhalten der Zustandsprüfung der Auto Scaling-Gruppe einer Umgebung. Instances, die eine Elastic Load Balancing Balancing-Zustandsprüfung nicht bestehen, werden nicht automatisch durch Amazon EC2 Auto Scaling ersetzt, es sei denn, Sie konfigurieren Amazon EC2 Auto Scaling manuell entsprechend. Details dazu finden Sie unter Einstellung für die Auto Scaling Scaling-Integritätsprüfung für Ihre Elastic Beanstalk Beanstalk-Umgebung.

Weitere Informationen zu Zustandsprüfungen und deren Auswirkungen auf den Gesamtstatus der Umgebung finden Sie unter Grundlegende Zustandsberichte.

Sitzungen

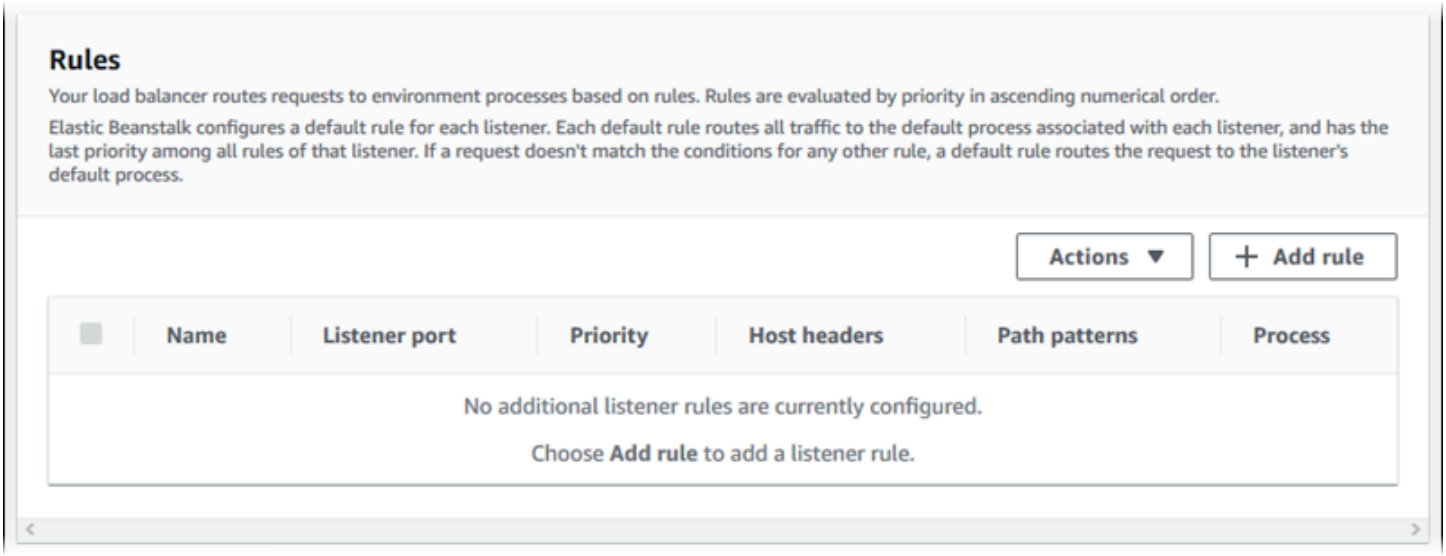
Aktivieren oder deaktivieren Sie das Kontrollkästchen Stickiness policy enabled (Sticky-Richtlinie aktiviert) zum Aktivieren oder Deaktivieren von Sticky Sessions. Verwenden Sie Cookie duration (Cookie-Dauer) zum Konfigurieren der Dauer einer Sticky Session (auf bis zu **604800** Sekunden).



Regeln

In dieser Liste können Sie benutzerdefinierte Listener-Regeln für Ihren Load Balancer angeben. Eine Regel ordnet Anforderungen, die der Listener unter einem spezifischen Pfadmuster erhält, einem Zielprozess zu. Jeder Listener kann mehrere Regeln aufweisen und Anforderungen auf verschiedenen Pfaden an verschiedene Prozesse auf Ihren Instances weiterleiten.

Regeln weisen numerische Prioritäten auf, die die Priorität ihrer Anwendung auf eingehende Anforderungen bestimmen. Für jeden Listener, den Sie hinzufügen, fügt Elastic Beanstalk eine Standardregel hinzu, die den Datenverkehr sämtlicher Listener an den Standardprozess weiterleitet. Die Standardregel weist die geringste Priorität auf. Sie wird nur angewendet, wenn für denselben Listener keine weitere Regel der eingehenden Anforderung entspricht. Wenn Sie keine benutzerdefinierten Regeln hinzugefügt haben, ist die Liste zunächst leer. Listener-Standardregeln werden nicht angezeigt.

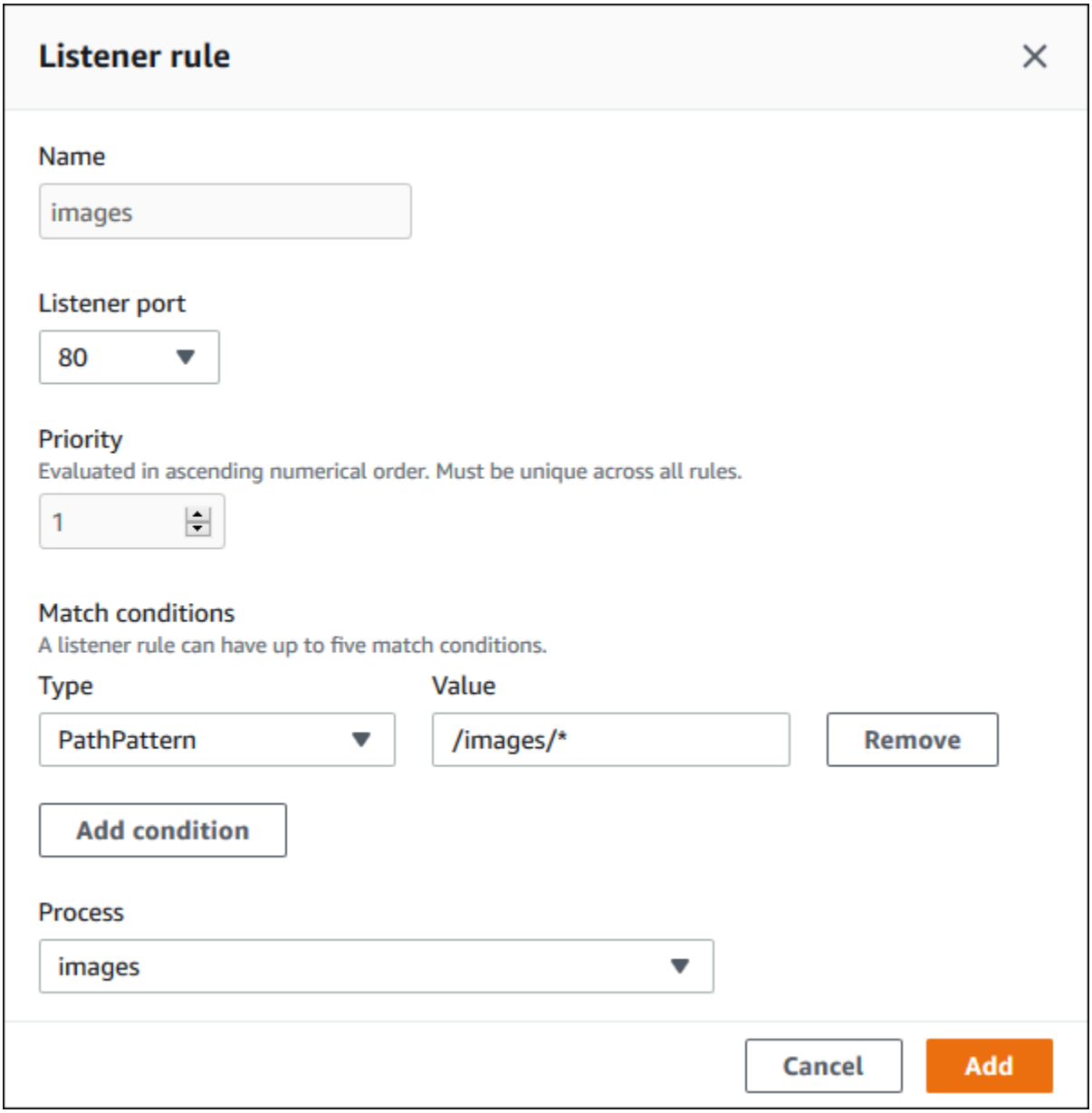


Sie können die Einstellungen einer vorhandenen Regel bearbeiten oder eine neue Regel hinzufügen. Um mit dem Bearbeiten einer Regel auf der Liste zu beginnen oder die Regel zur Liste hinzuzufügen, verwenden Sie dieselben Schritte, die für die Liste der Listener aufgelistet sind. Das Dialogfeld Listener rule (Listener-Regel) mit den folgenden Einstellungen wird geöffnet:

- Name – Der Name der Regel
- Listener port (Listener-Port) – Der Port des Listener, für den die Regel gilt
- Priority (Priorität) – Die Priorität der Regel Je niedriger die Zahl für die Priorität, desto höher die Priorität. Prioritäten der Regeln eines Listener müssen eindeutig sein.

- Match conditions (Übereinstimmungsbedingungen) – Eine Liste von Anforderungs-URL-Bedingungen, für die die Regel gilt. Es gibt zwei Arten von Bedingungen: HostHeader(den Domain-Teil der URL) und PathPattern(den Pfadteil der URL). Sie können bis zu fünf Bedingungen hinzufügen. Jeder Bedingungswert darf bis zu 128 Zeichen enthalten, darunter auch Platzhalterzeichen.
- Process (Prozess) – Der Prozess, in dem der Load Balancer Anforderungen weiterleitet, die der Regel entsprechen

Beim Bearbeiten einer vorhandenen Regel können Sie Name und Listener port (Listener-Port) nicht ändern.

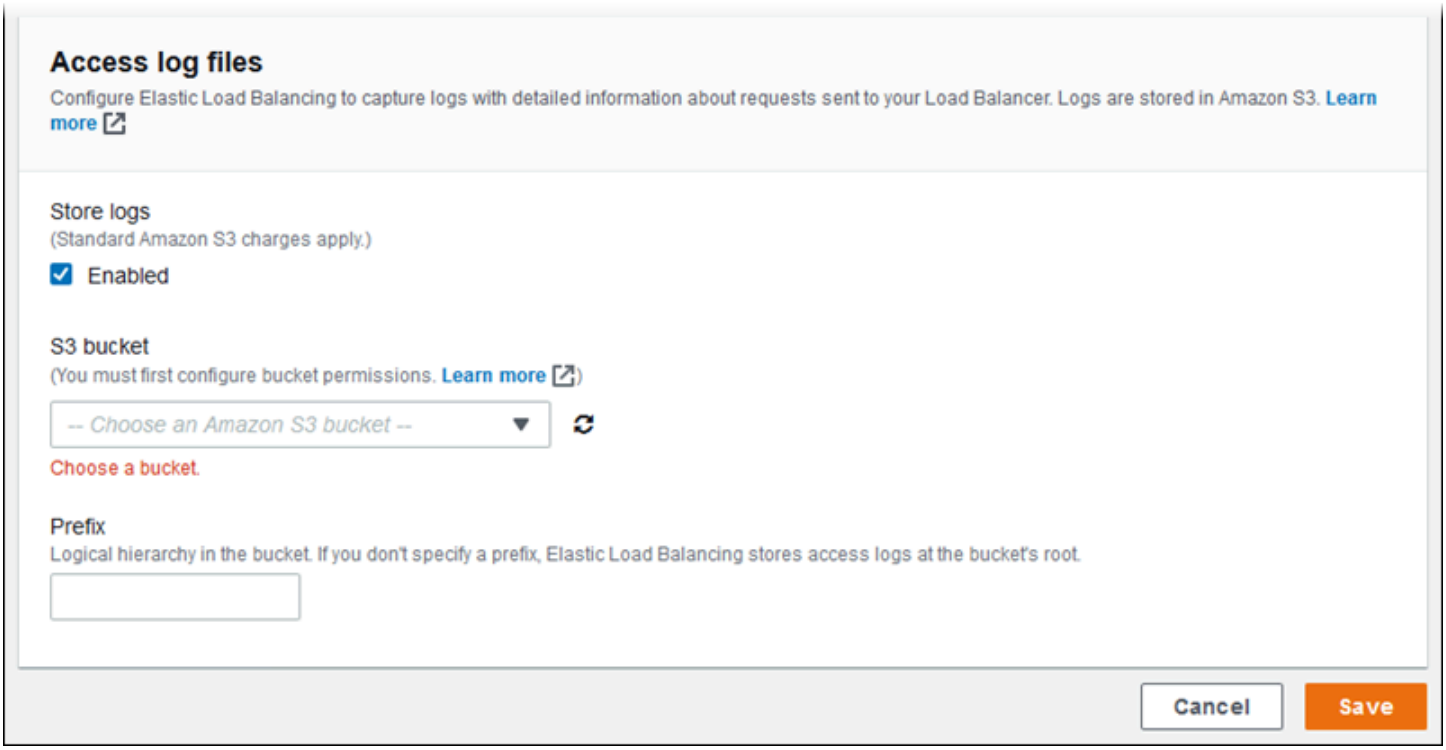


Zugangsprotokollerfassung

Verwenden Sie diese Einstellungen, um Elastic Load Balancing so zu konfigurieren, dass Protokolle mit detaillierten Informationen über an Ihren Application Load Balancer gesendete Anfragen erfasst werden. Access Log Capture (Auf Protokollerfassung zugreifen) ist standardmäßig deaktiviert. Wenn Store logs (Protokolle speichern) aktiviert ist, speichert Elastic Load Balancing die Protokolle in dem von Ihnen konfigurierten S3-Bucket. Die Präfix-Einstellung gibt im Bucket für die Protokolle einen übergeordneten Ordner an. Elastic Load Balancing speichert die Protokolle in einem Ordner mit dem Namen `AWSLogs` unter Ihrem Präfix. Wenn Sie kein Präfix angeben, legt Elastic Load Balancing seinen Ordner auf der Stammebene des Buckets ab.

> **Note**
>
> Wenn der Amazon S3 S3-Bucket, den Sie für die Erfassung von Zugriffsprotokollen konfigurieren, nicht der Bucket ist, den Elastic Beanstalk für Ihr Konto erstellt hat, stellen Sie sicher, dass Sie Ihren AWS Identity and Access Management (IAM-) Benutzern eine Benutzerrichtlinie mit den entsprechenden Berechtigungen hinzufügen. Die verwalteten Benutzerrichtlinien, die Elastic Beanstalk bietet, decken nur Berechtigungen für von Elastic Beanstalk verwaltete Ressourcen ab.

Einzelheiten zu Zugriffsprotokollen, einschließlich Berechtigungen und anderer Anforderungen, finden Sie unter Zugriffsprotokolle für Ihren Application Load Balancer.



Beispiel: Application Load Balancer mit einem sicheren Listener und zwei Prozessen

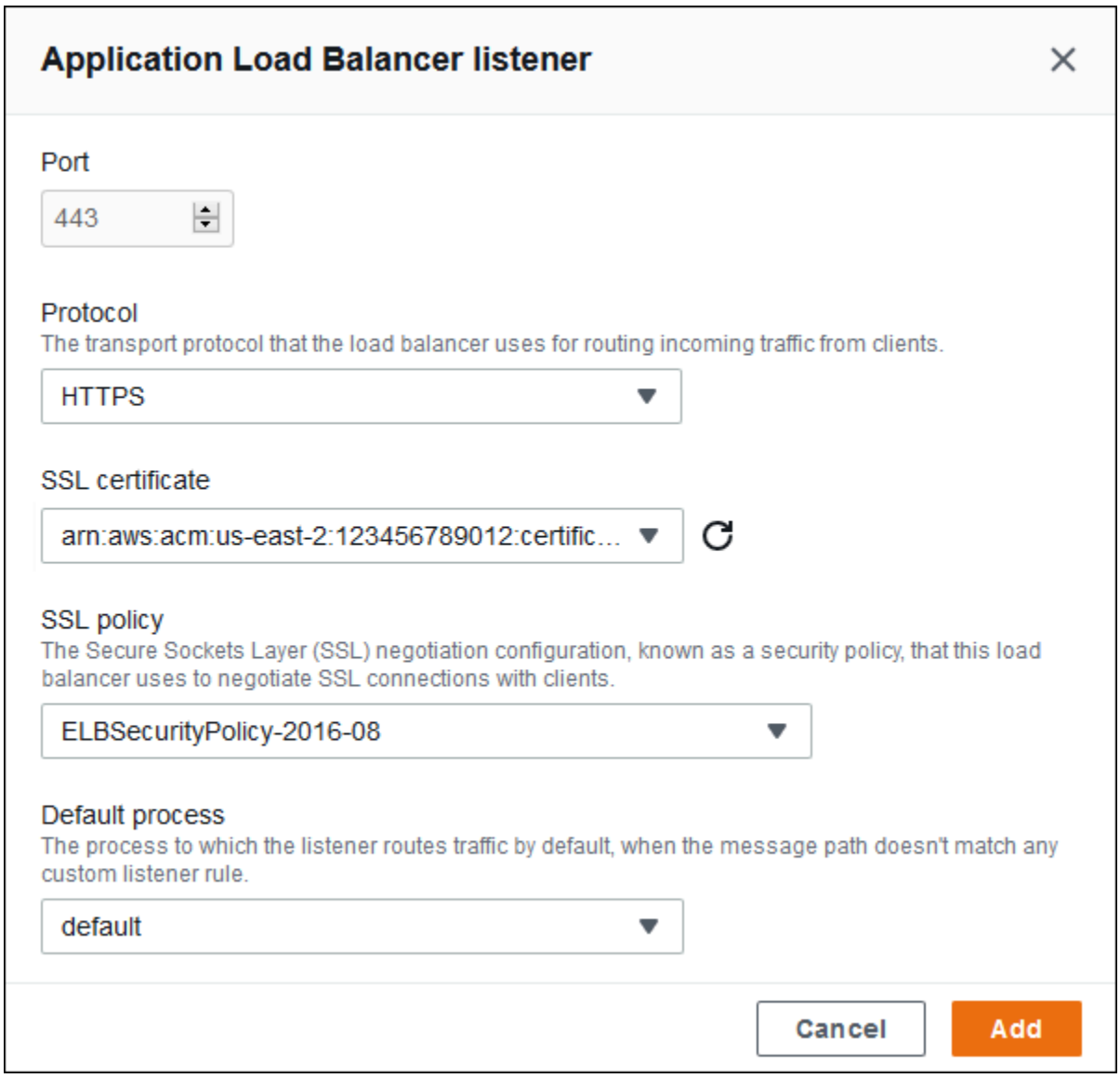
In diesem Beispiel erfordert Ihre Anwendung eine Verschlüsselung des end-to-end Datenverkehrs und einen separaten Prozess für die Bearbeitung administrativer Anfragen.

Um den Application Load Balancer Ihrer Umgebung so zu konfigurieren, dass er diese Anforderungen erfüllt, entfernen Sie den Standard-Listener, fügen einen HTTPS-Listener hinzu, geben an, dass der Standardprozess Port 443 auf HTTPS überwacht, und fügen einen Prozess und eine Listener-Regel für Administratordatenverkehr auf einem anderen Pfad hinzu.

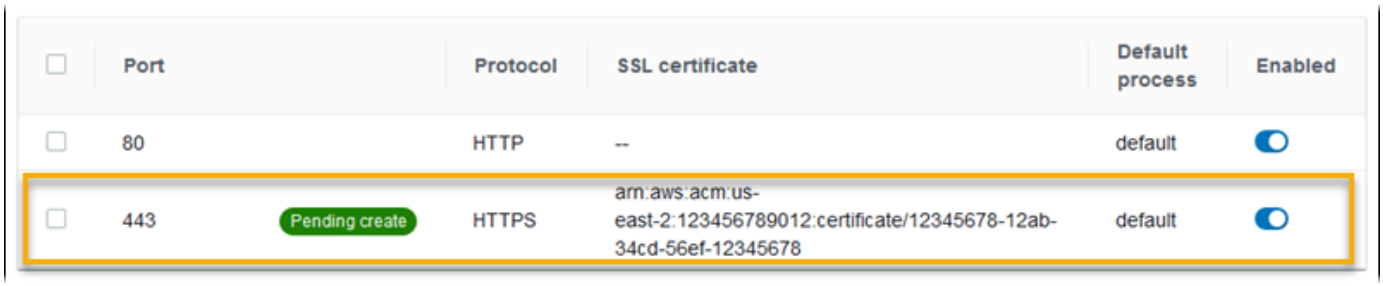
So konfigurieren Sie den Load Balancer für dieses Beispiel

1. Fügen Sie einen sicheren Listener hinzu. Geben Sie in Port **`443`** ein. Wählen Sie in Protocol (Protokoll) die Option **`HTTPS`** aus. Wählen Sie in SSL certificate (SSL-Zertifikat) den ARN Ihres SSL-Zertifikats aus. Beispiel: **`arn:aws:iam::123456789012:server-certificate/abc/certs/build`** oder **`arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`**.

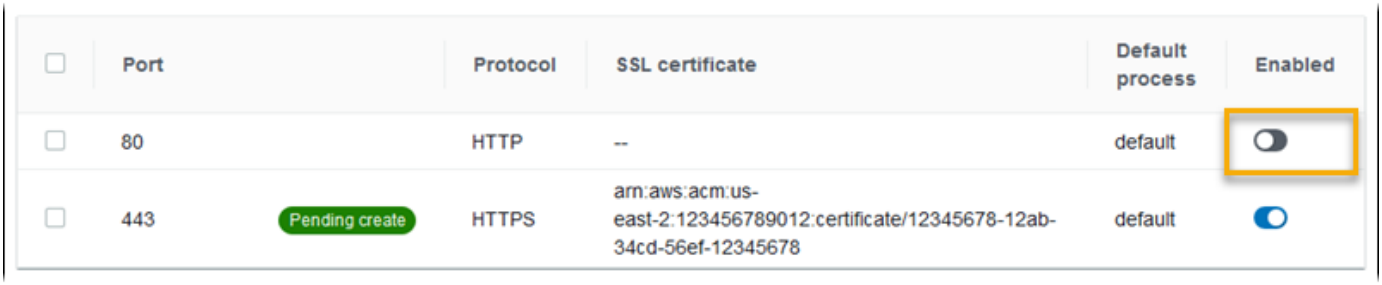
   Lassen Sie in Default process (Standardprozess) **`default`** ausgewählt.



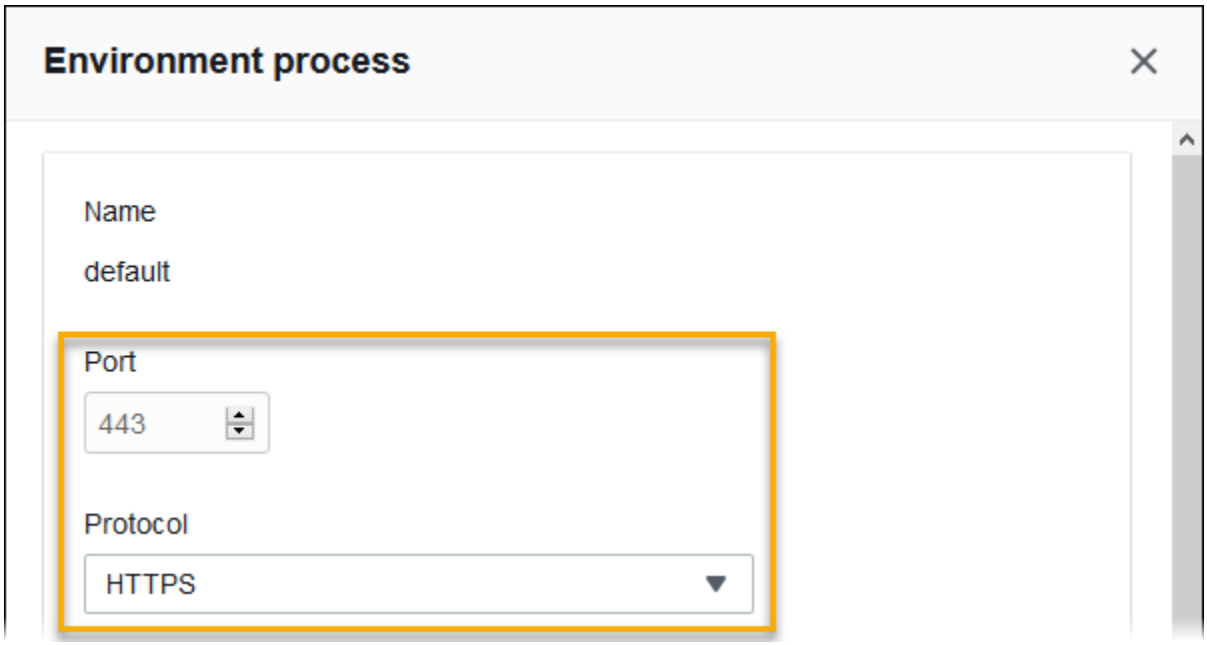
Sie können nun Ihre zusätzlichen Listener auf der Liste sehen.



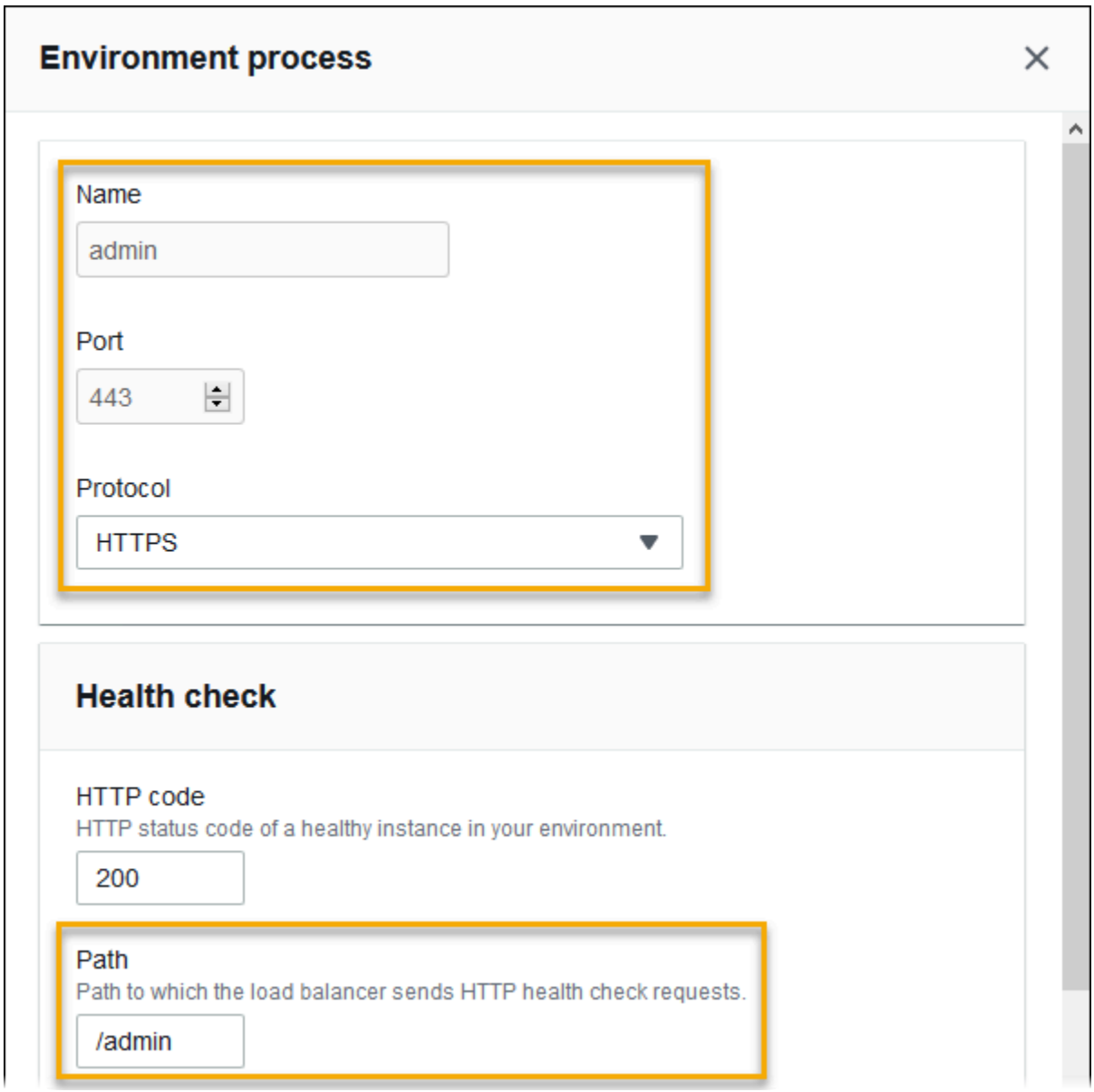
2. Deaktivieren Sie den HTTP-Listener am Standardport 80. Deaktivieren Sie für den Standard-Listener die Option Enabled (Aktiviert).



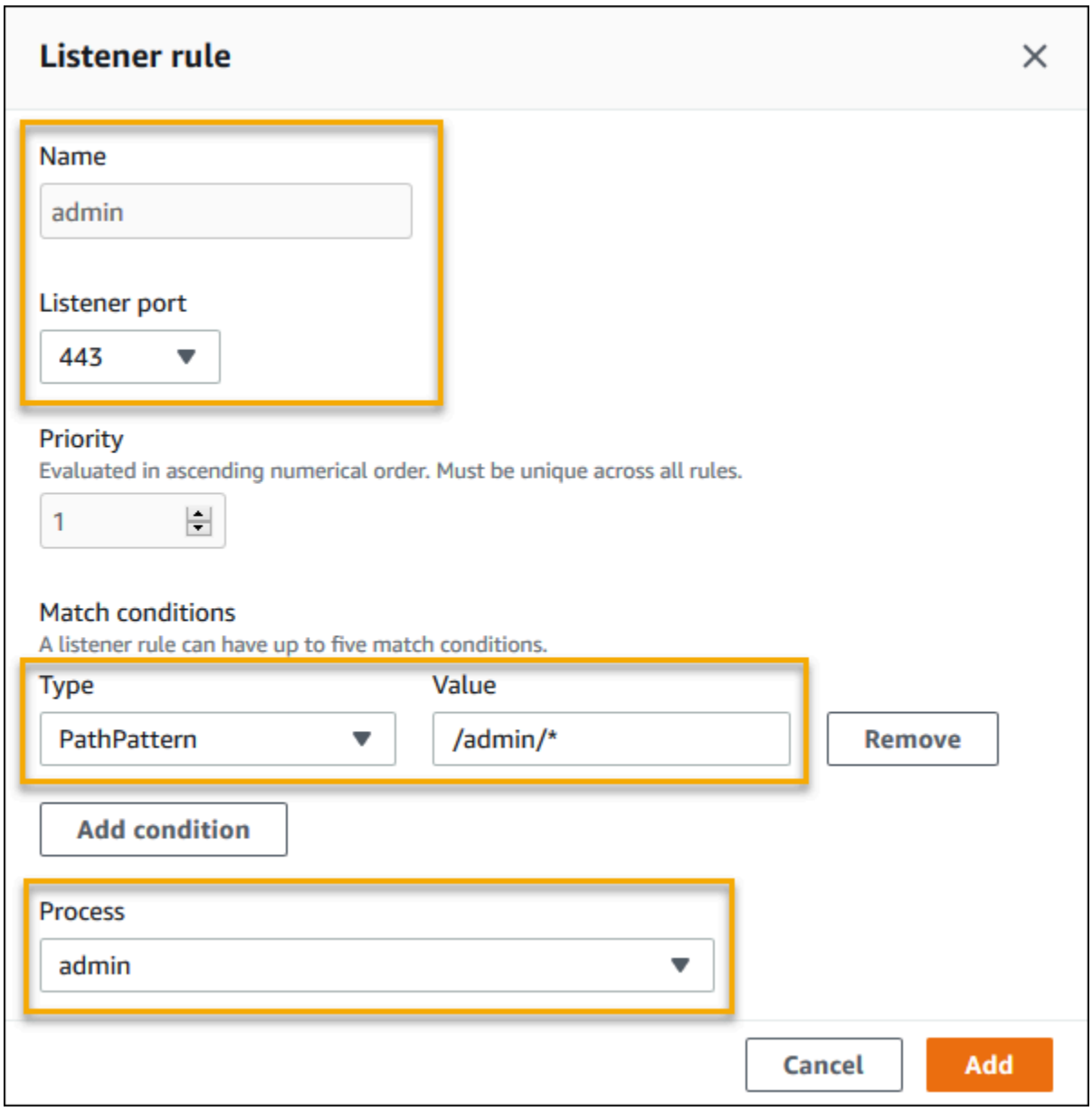
3. Konfigurieren Sie den Standardprozess für HTTPS. Wählen Sie den Standardprozess aus und wählen Sie dann für Actions (Aktionen) die Option Edit (Bearbeiten) aus. Geben Sie in Port **`443`** ein. Wählen Sie in Protocol (Protokoll) die Option **`HTTPS`** aus.



4. Fügen Sie einen Administratorprozess hinzu. Geben Sie bei Name **`admin`** ein. Geben Sie in Port **`443`** ein. Wählen Sie in Protocol (Protokoll) die Option **`HTTPS`** aus. Geben Sie in Health check (Zustandsprüfung) in Path (Pfad) **`/admin`** ein.



5. Fügen Sie eine Regel für Administratordatenverkehr hinzu. Geben Sie bei Name **`admin`** ein. Geben Sie für Listener port (Listener-Port) den Wert **`443`** ein. Fügen Sie für Match-Bedingungen a PathPatternmit dem Wert hinzu**`/admin/*`**. Wählen Sie in Process (Prozess) **`admin`** aus.



Application Load Balancer mit der EB CLI konfigurieren

Sie werden von der EB CLI aufgefordert, einen Load Balancer-Typ auszuwählen, wenn Sie eb create ausführen.

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF
```

```
Select a load balancer type
1) classic
2) application
3) network
(default is 2):
```

Sie können auch einen Load Balancer-Typ mit der `--elb-type`-Option angeben.

```
$ eb create test-env --elb-type application
```

Application Load Balancer-Namespaces

Einstellungen im Zusammenhang mit Application Load Balancers finden Sie in den folgenden Namespaces:

- `aws:elasticbeanstalk:environment` – Wählen Sie den Load Balancer-Typ für die Umgebung aus. Der Wert für einen Application Load Balancer ist `application`.

  Sie können diese Option nicht in Konfigurationsdateien (.Ebextensions) festlegen.
- `aws:elbv2:loadbalancer` – Konfigurieren Sie Zugriffsprotokolle und andere Einstellungen, die für den Application Load Balancer als Ganzes gelten.
- `aws:elbv2:listener` – Konfigurieren Sie Listener für den Application Load Balancer. Diese Einstellungen entsprechen den Einstellungen in `aws:elb:listener` für Classic Load Balancer.
- `aws:elbv2:listenerrule` – Konfigurieren Sie Regeln, die den Datenverkehr an verschiedene Prozesse weiterleiten, je nach Anfragepfad. Die Regeln gelten nur für Application Load Balancer.
- `aws:elasticbeanstalk:environment:process` – Konfigurieren Sie Zustandsprüfungen und geben Sie den Port und das Protokoll für die Prozesse an, die auf den Instances Ihrer Umgebung ausgeführt werden. Die Port- und Protokolleinstellungen entsprechen den Instance-Port und Instance-Protokoll-Einstellungen in `aws:elb:listener` für einen Listener auf einem Classic Load Balancer. Zustandsprüfungseinstellungen entsprechen den Einstellungen in den Namespaces `aws:elb:healthcheck` und `aws:elasticbeanstalk:application`.

Example.ebeextensions/ .config alb-access-logs

Die folgende Konfigurationsdatei ermöglicht Zugriffsprotokoll-Uploads für eine Umgebung mit einem Application Load Balancer.

```
option_settings:
```

```
  aws:elbv2:loadbalancer:
    AccessLogsS3Bucket: amzn-s3-demo-bucket
    AccessLogsS3Enabled: 'true'
    AccessLogsS3Prefix: beanstalk-alb
```

Example.ebextensions/ alb-default-process .config

Die folgende Konfigurationsdatei ändert Zustandsprüfungs- und "Sticky"-Einstellungen auf dem Standardprozess.

```
option_settings:
  aws:elasticbeanstalk:environment:process:default:
    DeregistrationDelay: '20'
    HealthCheckInterval: '15'
    HealthCheckPath: /
    HealthCheckTimeout: '5'
    HealthyThresholdCount: '3'
    UnhealthyThresholdCount: '5'
    Port: '80'
    Protocol: HTTP
    StickinessEnabled: 'true'
    StickinessLBCookieDuration: '43200'
```

Example.ebextensions/ alb-secure-listener .config

Die folgende Konfigurationsdatei fügt einen sicheren Listener und einen entsprechenden Prozess auf Port 443 hinzu.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
    SSLCertificateArns: arn:aws:acm:us-
east-2:123456789012:certificate/21324896-0fa4-412b-bf6f-f362d6eb6dd7
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
    Protocol: HTTPS
```

Example.ebextensions/ alb-admin-rule .config

Die folgende Konfigurationsdatei fügt einen sicheren Listener mit einer Regel hinzu, die Datenverkehr mit einem Anfragepfad von `/admin` an einen Prozess namens `admin`, der auf Port 4443 überwacht.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
    Rules: admin
    SSLCertificateArns: arn:aws:acm:us-
east-2:123456789012:certificate/21324896-0fa4-412b-bf6f-f362d6eb6dd7
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
    Protocol: HTTPS
  aws:elasticbeanstalk:environment:process:admin:
    HealthCheckPath: /admin
    Port: '4443'
    Protocol: HTTPS
  aws:elbv2:listenerrule:admin:
    PathPatterns: /admin/*
    Priority: 1
    Process: admin
```

Gemeinsamen Application Load Balancer konfigurieren

Wenn Sie Load Balancing aktivieren, ist Ihre AWS Elastic Beanstalk Umgebung mit einem Elastic Load Balancing Load Balancer ausgestattet, der den Traffic auf die Instances in Ihrer Umgebung verteilt. Elastic Load Balancing unterstützt verschiedene Load Balancer-Typen. Weitere Informationen dazu finden Sie im Elastic Load Balancing-Benutzerhandbuch. Elastic Beanstalk kann einen Load Balancer für Sie erstellen oder Sie einen von Ihnen erstellten freigegebenen Load Balancer angeben lassen.

In diesem Thema wird die Konfiguration eines freigegebenen Application Load Balancer beschrieben, das Sie erstellen und mit Ihrer Umgebung verknüpfen. Siehe auch the section called “Application Load Balancer”. Weitere Informationen zur Konfiguration aller Load Balancer-Typen, die Elastic Beanstalk unterstützt, finden Sie unter Load Balancer Ihrer Elastic-Beanstalk-Umgebung.

> **Note**
>
> Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern. Sie können auch nicht von einem dedizierten zu einem freigegebenen Load Balancer wechseln und umgekehrt.

Einführung

Ein Shared Load Balancer ist ein Load Balancer, den Sie mithilfe des Amazon Elastic Compute Cloud (Amazon EC2) -Service selbst erstellen und verwalten und dann in mehreren Elastic Beanstalk Beanstalk-Umgebungen verwenden.

Wenn Sie eine Umgebung mit Load-Balancing und Skalierung erstellen und sich für die Verwendung eines Application Load Balancers entscheiden, erstellt Elastic Beanstalk standardmäßig einen Load Balancer für Ihre Umgebung. Was ein Application Load Balancer ist und wie er in einer Elastic Beanstalk-Umgebung funktioniert, erfahren Sie unter Einführung zur Konfiguration eines Application Load Balancers für Elastic Beanstalk.

In einigen Situationen möchten Sie möglicherweise die Kosten für mehrere dedizierte Load Balancer sparen. Dies kann bei mehreren Umgebungen hilfreich sein, z. B. wenn Ihre Anwendung eine Suite von Mikroservices statt ein monolithischer Service ist. In solchen Fällen können Sie einen freigegebenen Load Balancer verwenden.

Um einen Shared Load Balancer zu verwenden, erstellen Sie ihn zunächst in Amazon EC2 und fügen Sie einen oder mehrere Listener hinzu. Während der Erstellung einer Elastic Beanstalk-Umgebung stellen Sie dann den Load Balancer bereit und wählen einen Listener-Port aus. Elastic Beanstalk ordnet den Listener dem Standardprozess in Ihrer Umgebung zu. Sie können benutzerdefinierte Listener-Regeln hinzufügen, um den Datenverkehr von bestimmten Host-Headern und -pfaden zu anderen Umgebungsprozessen weiterzuleiten.

Elastic Beanstalk fügt dem gemeinsamen Load Balancer ein Tag hinzu. Der Tag-Name lautet `elasticbeanstalk:shared-elb-environment-count`, und sein Wert ist die Anzahl der Umgebungen, für die dieser Load Balancer freigegeben ist.

Die Verwendung eines freigegebenen Load Balancer unterscheidet sich von der Verwendung eines dedizierten Load Balancer auf verschiedene Arten

Regarding	Dedizierter Application Load Balancer	Geteile Application Load Balancer
Verwaltung	Elastic Beanstalk erstellt und verwaltet den Load Balancer, Listener, Listener-Regeln und Prozesse (Zielgruppen). Elastic Beanstalk entfernt sie auch, wenn Sie Ihre Umgebung terminieren. Elastic Beanstalk kann die Erfassung des Zugriffsprotokolls des Load Balancers festlegen, wenn Sie diese Option auswählen.	Sie erstellen und verwalten den Load Balancer und die Listener außerhalb von Elastic Beanstalk. Elastic Beanstalk erstellt und verwaltet eine Standardregel und einen Standardprozess, und Sie können Regeln und Prozesse hinzufügen. Elastic Beanstalk entfernt die Listener-Regeln und -Prozesse, die während der Erstellung der Umgebung hinzugefügt wurden.
Listener-Regeln	Elastic Beanstalk erstellt eine Standardregel für jeden Listener, um den gesamten Datenverkehr an den Standardprozess des Listeners weiterzuleiten.	Elastic Beanstalk ordnet eine Standardregel nur einem Port 80-Listener zu, sofern einer vorhanden ist. Wenn Sie einen anderen Standard-Listener-Port wählen, müssen Sie ihm die Standardregel zuordnen (die Elastic Beanstalk-Konsole und die EB CLI erledigen dies für Sie). Um Konflikte mit Listener-Regelkonflikten in Umgebungen zu lösen, die den freigegebenen Load Balancer nutzen, fügt Elastic Beanstalk der Listener-Regel den CNAME der Umgebung als Host-Header-Bedingung hinzu. Elastic Beanstalk behandelt Regelprioritätseinstellungen als relativ über Umgebungen hinweg, die den freigegebenen Load Balancer nutzen, und ordnet sie bei der Erstellung absoluten Prioritäten zu.
Sicherheitsgruppen	Elastic Beanstalk erstellt eine Standardsicherheitsgruppe und fügt an den Load Balancer an.	Sie können eine oder mehrere Sicherheitsgruppen für den Load Balancer konfigurieren. Wenn dies nicht der Fall ist, prüft Elastic

Regarding	Dedizierter Application Load Balancer	Geteile Application Load Balancer
		Beanstalk, ob eine vorhandene Sicherhei tsgruppe, die Elastic Beanstalk verwaltet , bereits an den Load Balancer angehängt ist. Falls nicht, erstellt Elastic Beanstalk eine Sicherheitsgruppe und hängt sie an den Load Balancer an. Elastic Beanstalk löscht diese Sicherheitsgruppe, wenn die letzte Umgebung, die den Load Balancer teilt, beendet wird.
Aktualisi erungen	Sie können Ihren Application Load Balancer nach der Erstellun g der Umgebung aktualisieren. Sie können Listener, Listener-Regeln und Prozesse bearbeiten. Sie können die Zugriffsprotokolle rfassung des Load Balancer konfigurieren.	Sie können Elastic Beanstalk nicht zur Konfiguration der Zugriffsprotokollierung in Ihrem Application Load Balancer verwenden und Sie können die Listener und Listener-Regeln nach der Erstellung der Umgebung nicht aktualisieren. Sie können nur Prozesse (Zielgruppen) aktualisieren. Verwenden Sie Amazon, um die Erfassung von Zugriffsp rotokollen zu konfigurieren und Listener und Listener-Regeln zu aktualisieren. EC2

Gemeinsamen Application Load Balancer mit der Elastic Beanstalk-Konsole konfigurieren

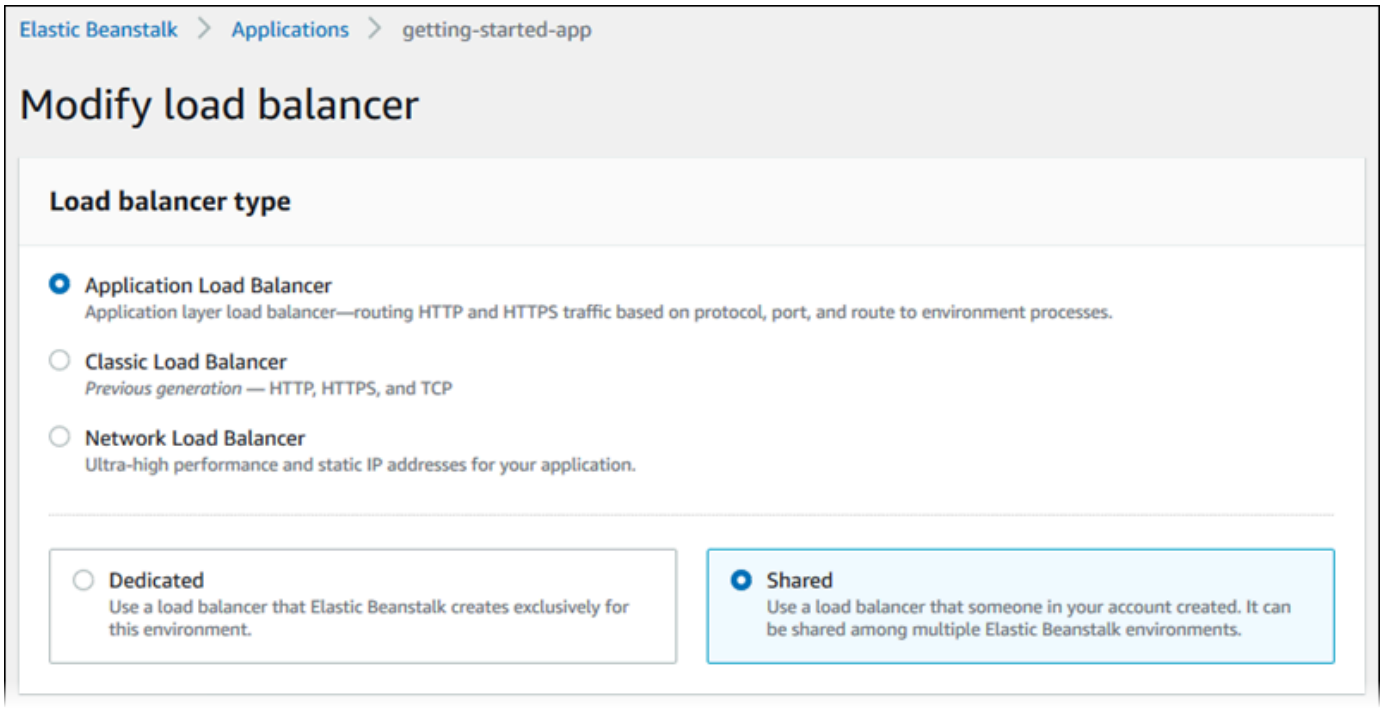
Sie können die Elastic Beanstalk-Konsole verwenden, um während der Erstellung der Umgebung einen gemeinsamen Application Load Balancer zu konfigurieren. Sie können einen der freigabefähigen Load Balancer Ihres Kontos für die Verwendung in der Umgebung auswählen, den Standard-Listener-Port auswählen und zusätzliche Prozesse und Listener-Regeln konfigurieren.

Sie können Ihre gemeinsame Application Load Balancer-Konfiguration nicht mehr in der Application Load Balancer-Konsole bearbeiten, nachdem Ihre Umgebung erstellt wurde. Verwenden Sie Amazon, um Listener, Listener-Regeln, Prozesse (Zielgruppen) und die Erfassung von Zugriffsprotokollen zu konfigurieren. EC2

So konfigurieren Sie einen Application Load Balancer in der Elastic Beanstalk-Konsole während der Erstellung der Umgebung:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus.
3. Wählen Sie Create a new environment (Neue Umgebung erstellen), um mit der Erstellung Ihrer Umgebung zu beginnen.
4. Wählen Sie auf der Hauptseite des Assistenten zuerst Configure more options (Weitere Optionen konfigurieren) und dann Create environment (Umgebung erstellen) aus.
5. Wählen Sie die Konfigurationsvoreinstellung High availability (Hohe Verfügbarkeit) aus.

   Alternativ können Sie in der Konfigurationskategorie Capacity (Kapazität) einen Umgebungstyp mit Load balanced (Lastenverteilung) konfigurieren. Details hierzu finden Sie unter Capacity (Kapazität).
6. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.
7. Wählen Sie die Option Application Load Balancer aus, falls sie nicht bereits ausgewählt ist. Wählen Sie dann die Option Shared (Gemeinsam) aus.



8. Nehmen Sie alle Konfigurationsänderungen am gemeinsamen Application Load Balancer vor, die für Ihre Umgebung erforderlich sind.

9. Wählen Sie Save (Speichern) aus und nehmen Sie dann alle weiteren Konfigurationsänderungen für Ihre Umgebung vor.
10. Wählen Sie Create environment (Umgebung erstellen) aus.

Einstellungen für gemeinsame Application Load Balancer

- Geteile Application Load Balancer
- Prozesse
- Regeln

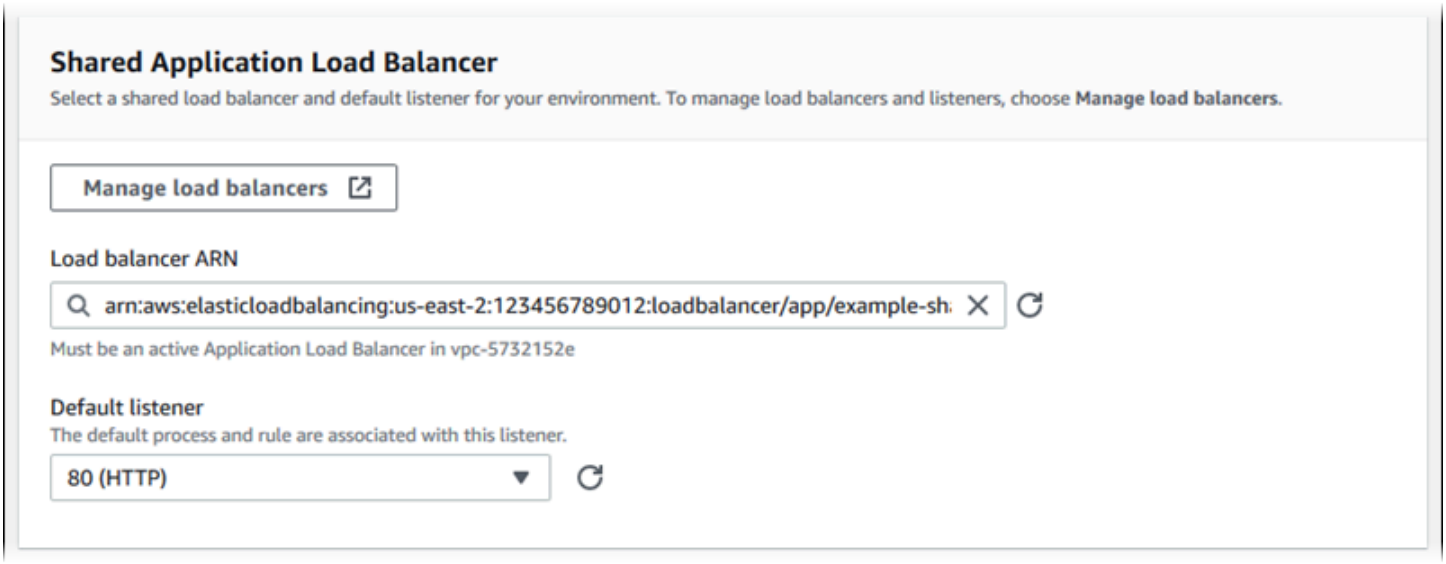
Geteile Application Load Balancer

In diesem Abschnitt wählen Sie einen gemeinsamen Application Load Balancer für Ihre Umgebung aus und konfigurieren das Standard-Datenverkehrs-Routing.

Bevor Sie hier einen gemeinsam genutzten Application Load Balancer konfigurieren können, verwenden Sie Amazon, EC2 um mindestens einen Application Load Balancer für die gemeinsame Nutzung mit mindestens einem Listener in Ihrem Konto zu definieren. Falls Sie dies noch nicht getan haben, können Sie Manage load balancers (Load Balancer verwalten) auswählen. Elastic Beanstalk öffnet die EC2 Amazon-Konsole in einem neuen Browser-Tab.

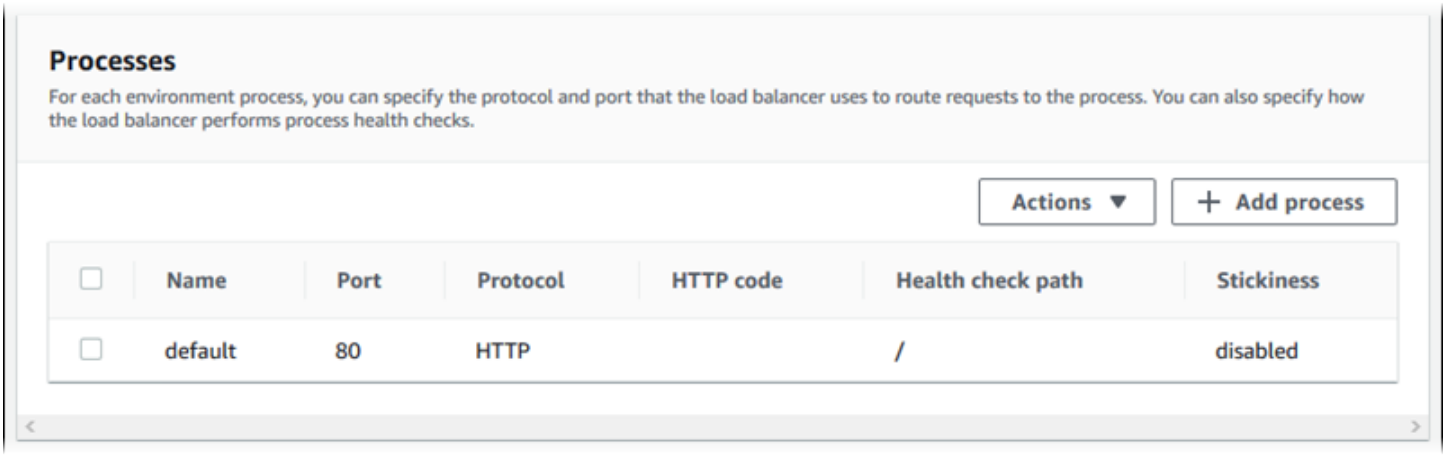
Wenn Sie mit der Konfiguration freigegebener Load Balancer außerhalb von Elastic Beanstalk fertig sind, konfigurieren Sie die folgenden Einstellungen in diesem Konsolenabschnitt:

- Load Balancer ARN – Der freigegebene Load Balancer, der in dieser Umgebung verwendet werden soll. Wählen Sie aus einer Liste von Load Balancern aus oder geben Sie den Amazon-Ressourcennamen (ARN) eines Load Balancers ein.
- Default listener port (Standard-Listener-Port) – Ein Listener-Port, den der freigegebene Load Balancer überwacht. Wählen Sie aus einer Liste vorhandener Listener-Ports aus. Der Datenverkehr von diesem Listener mit dem CNAME der Umgebung im Host-Header wird an einen Standardprozess in dieser Umgebung weitergeleitet.



Prozesse

Verwenden Sie diese Liste, um Prozesse für Ihren freigegebenen Load Balancer anzugeben. Ein Prozess ist ein Ziel für Listener, an das Datenverkehr weitergeleitet wird. Anfangs zeigt die Liste den Standardprozess an, der Datenverkehr vom Standard-Listener empfängt.



So konfigurieren Sie einen vorhandenen Prozess

1. Aktivieren Sie das Kontrollkästchen neben seinem Tabelleneintrag und wählen Sie dann Actions (Aktionen), Edit (Bearbeiten) aus.
2. Verwenden Sie das Dialogfeld Environment process (Umgebungsprozess), um Einstellungen zu bearbeiten, und wählen Sie dann Save (Speichern).

So fügen Sie einen Prozess hinzu

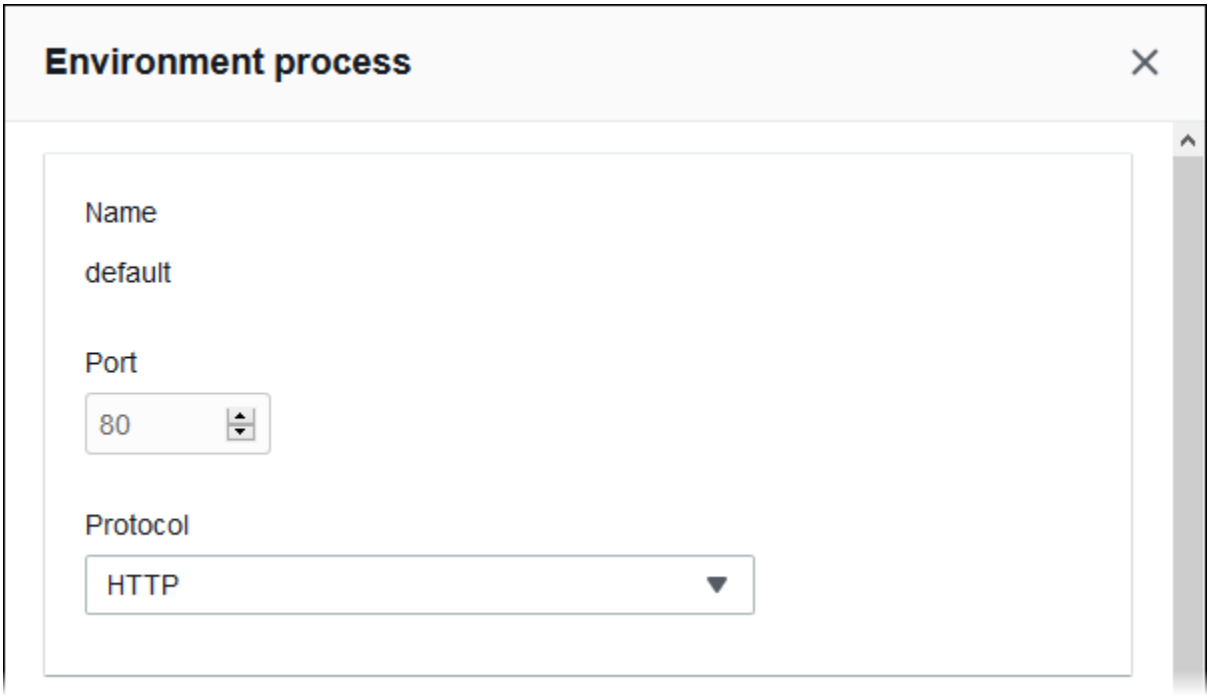
1. Wählen Sie Add process (Prozess hinzufügen).
2. Konfigurieren Sie im Dialogfeld Environment process (Umgebungsprozess) die gewünschten Einstellungen, und wählen Sie dann Add (Hinzufügen).

Prozesseinstellungen im Dialogfeld für die Umgebung des Application Load Balancers

- Definition
- Gesundheitscheck
- Sitzungen

Definition

Verwenden Sie diese Einstellungen zum Definieren des Prozesses: Name, Port und Protocol (Protokoll) für die Überwachung auf Anforderungen.

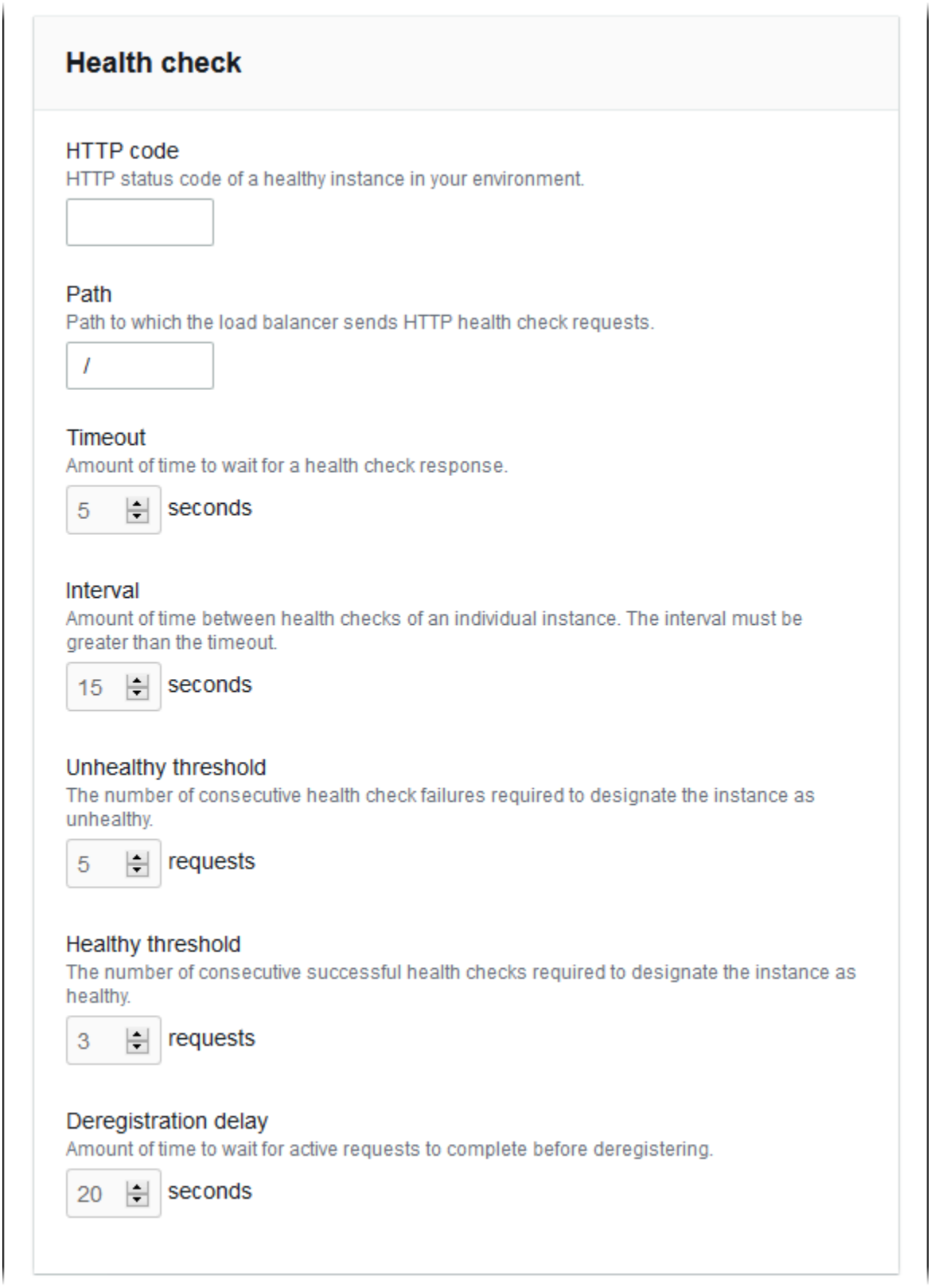


Gesundheitscheck

Verwenden Sie die folgenden Einstellungen zum Konfigurieren von Zustandsprüfungen für den Prozess:

- HTTP code (HTTP-Code) – Der HTTP-Statuscode, der einen funktionsfähigen Prozess bestimmt

- Path (Pfad) – Der Anforderungspfad der Zustandsprüfung für den Prozess
- Timeout – Die Wartezeit in Sekunden, bis eine Antwort von der Zustandsprüfung eingeht
- Interval (Intervall) – Der Zeitraum in Sekunden zwischen Zustandsprüfungen einer einzelnen Instance Der Intervall-Wert muss größer als der Timeout-Wert sein.
- Unhealthy threshold, Healthy threshold – Die Anzahl der Zustandsprüfungen, die nicht bestanden bzw. bestanden sein müssen, bevor Elastic Load Balancing den Zustand der Instance ändert.
- Deregistration delay (Verzögerung der Registrierungsaufhebung) – Die Zeit in Sekunden, die gewartet werden muss, bis aktive Anforderungen abgeschlossen sind, ehe eine Instance abgemeldet wird

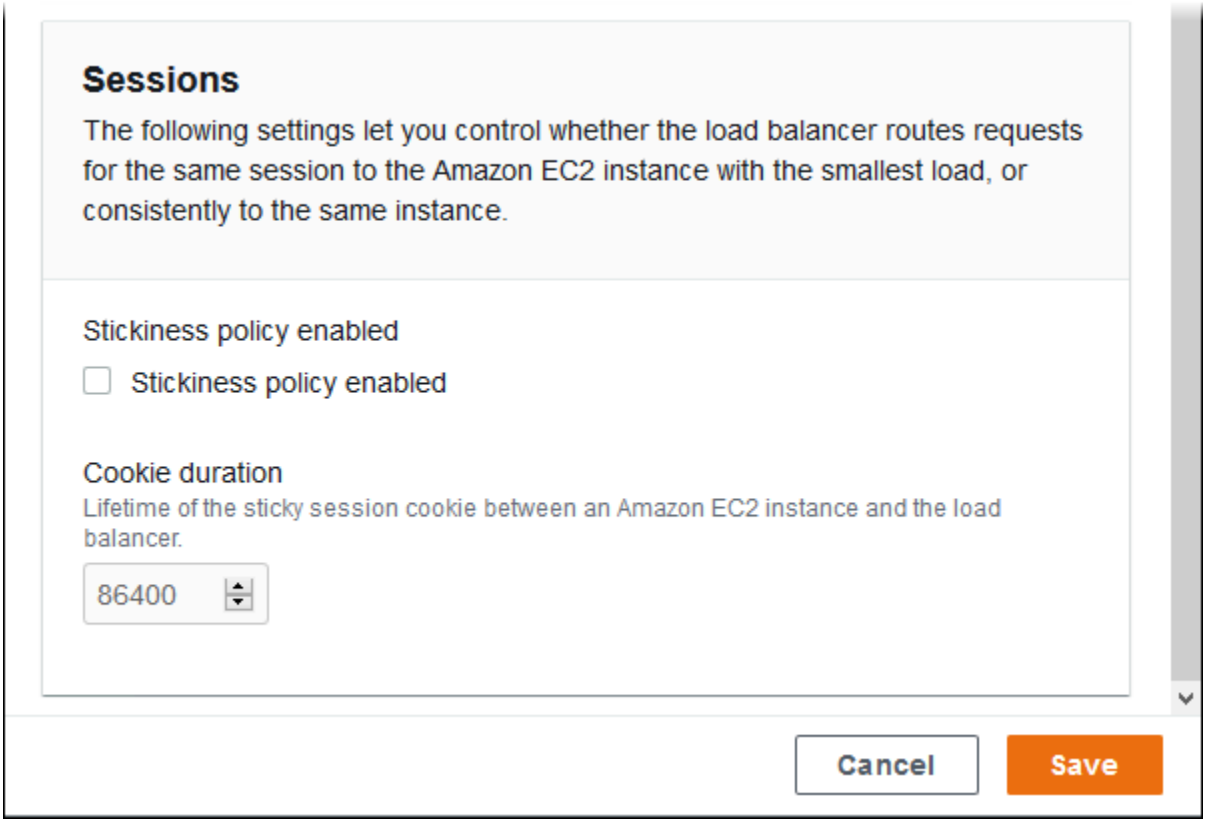

Health check
HTTP code
HTTP status code of a healthy instance in your environment.
Path
Path to which the load balancer sends HTTP health check requests.
/
Timeout
Amount of time to wait for a health check response.
5 seconds
Interval
Amount of time between health checks of an individual instance. The interval must be greater than the timeout.
15 seconds
Unhealthy threshold
The number of consecutive health check failures required to designate the instance as unhealthy.
5 requests
Healthy threshold
The number of consecutive successful health checks required to designate the instance as healthy.
3 requests
Deregistration delay
Amount of time to wait for active requests to complete before deregistering.
20 seconds

> **Note**
>
> Die Zustandsprüfung für Elastic Load Balancing hat keinen Einfluss auf das Verhalten der Zustandsprüfung der Auto Scaling-Gruppe einer Umgebung. Instances, die eine Elastic Load Balancing Balancing-Zustandsprüfung nicht bestehen, werden nicht automatisch durch Amazon EC2 Auto Scaling ersetzt, es sei denn, Sie konfigurieren Amazon EC2 Auto Scaling manuell entsprechend. Details dazu finden Sie unter Einstellung für die Auto Scaling Scaling-Integritätsprüfung für Ihre Elastic Beanstalk Beanstalk-Umgebung.

Weitere Informationen zu Zustandsprüfungen und deren Auswirkungen auf den Gesamtstatus der Umgebung finden Sie unter Grundlegende Zustandsberichte.

Sitzungen

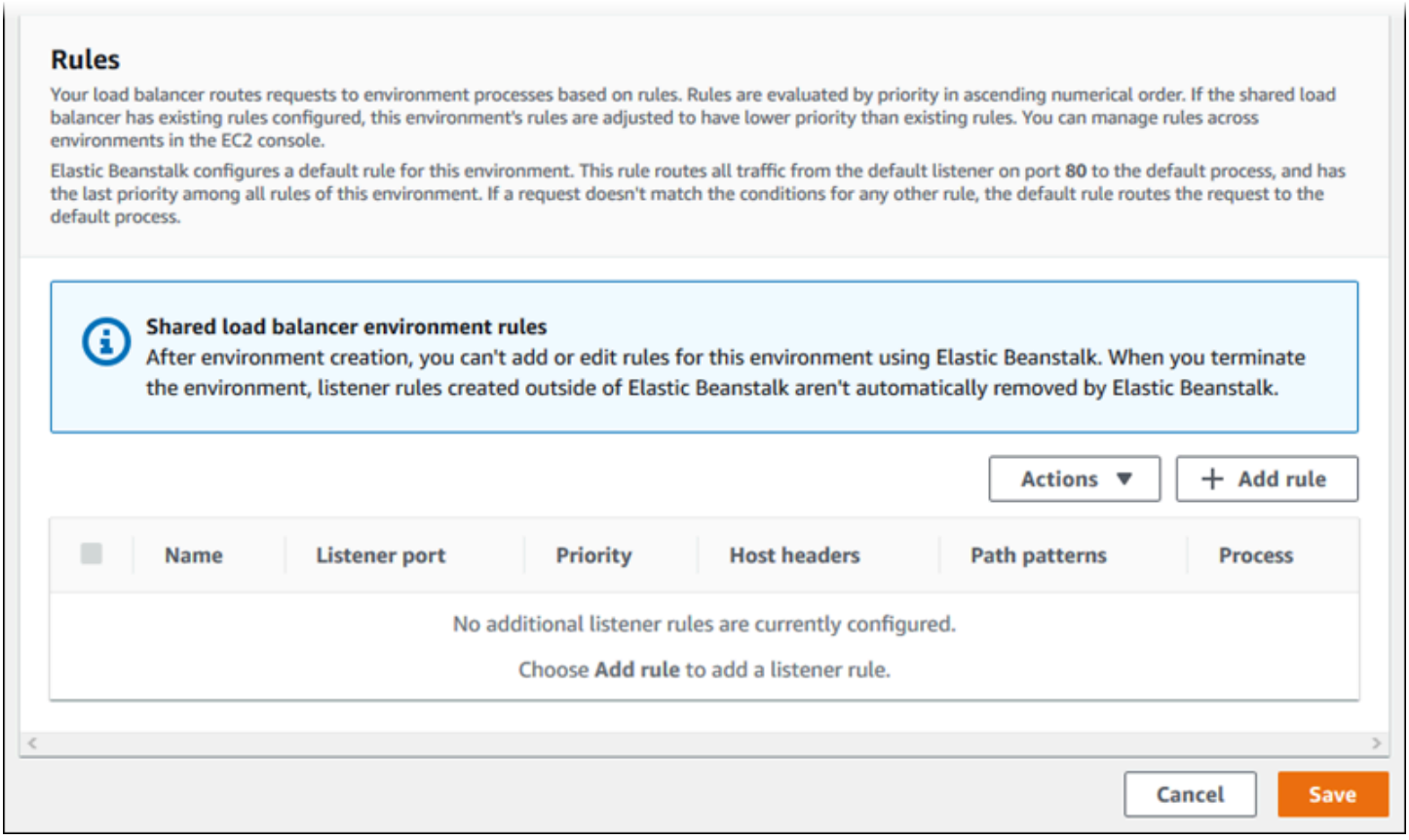
Aktivieren oder deaktivieren Sie das Kontrollkästchen Stickiness policy enabled (Sticky-Richtlinie aktiviert) zum Aktivieren oder Deaktivieren von Sticky Sessions. Verwenden Sie Cookie duration (Cookie-Dauer) zum Konfigurieren der Dauer einer Sticky Session (auf bis zu **604800** Sekunden).



Regeln

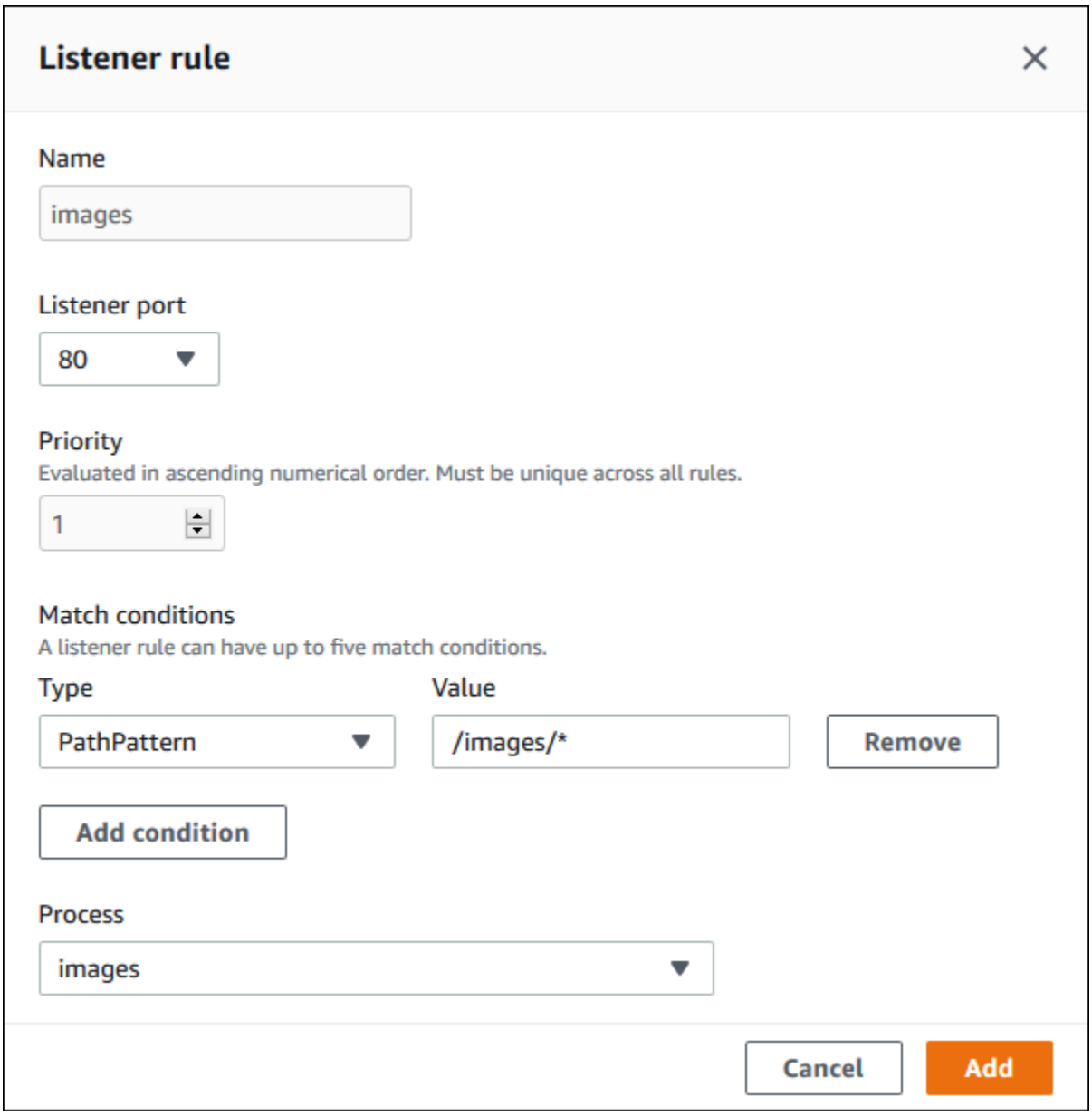
In dieser Liste können Sie benutzerdefinierte Listener-Regeln für Ihren freigegebenen Load Balancer angeben. Eine Regel ordnet Anforderungen, die der Listener unter einem spezifischen Pfadmuster erhält, einem Zielprozess zu. Jeder Listener kann mehrere Regeln aufweisen und Anforderungen auf verschiedenen Pfaden an verschiedene Prozesse auf Instances der verschiedenen Umgebungen, die den freigegebenen Listener nutzen, weiterleiten.

Regeln weisen numerische Prioritäten auf, die die Priorität ihrer Anwendung auf eingehende Anforderungen bestimmen. Elastic Beanstalk fügt eine Standardregel hinzu, die den gesamten Datenverkehr des Standard-Listeners an den Standardprozess Ihrer neuen Umgebung weiterleitet. Die Standardregel weist die geringste Priorität auf. Sie wird nur angewendet, wenn für denselben Listener keine weitere Regel der eingehenden Anforderung entspricht. Wenn Sie keine benutzerdefinierten Regeln hinzugefügt haben, ist die Liste zunächst leer. Die Standardregel wird nicht angezeigt.



Sie können die Einstellungen einer vorhandenen Regel bearbeiten oder eine neue Regel hinzufügen. Um mit dem Bearbeiten einer Regel auf der Liste zu beginnen oder die Regel zur Liste hinzuzufügen, verwenden Sie dieselben Schritte, die für die Prozessliste aufgelistet sind. Das Dialogfeld Listener rule (Listener-Regel) mit den folgenden Einstellungen wird geöffnet:

- Name – Der Name der Regel
- Listener port (Listener-Port) – Der Port des Listener, für den die Regel gilt
- Priority (Priorität) – Die Priorität der Regel Je niedriger die Zahl für die Priorität, desto höher die Priorität. Prioritäten der Regeln eines Listener müssen eindeutig sein. Im Fall eines freigegebenen Load Balancer behandelt Elastic Beanstalk Regelprioritäten in freigebenden Umgebungen als relativ und ordnet sie während der Erstellung absoluten Prioritäten zu.
- Match conditions (Übereinstimmungsbedingungen) – Eine Liste von Anforderungs-URL-Bedingungen, für die die Regel gilt. Es gibt zwei Arten von Bedingungen: HostHeader(den Domain-Teil der URL) und PathPattern(den Pfadteil der URL). Eine Bedingung ist für die Subdomäne der Umgebung reserviert, und Sie können bis zu vier Bedingungen hinzufügen. Jeder Bedingungswert darf bis zu 128 Zeichen enthalten, darunter auch Platzhalterzeichen.
- Process (Prozess) – Der Prozess, in dem der Load Balancer Anforderungen weiterleitet, die der Regel entsprechen



Beispiel: Verwenden Sie einen gemeinsam genutzten Application Load Balancer für eine sichere micro-service-based Anwendung

In diesem Beispiel besteht Ihre Anwendung aus mehreren Micro-Services, die jeweils als Elastic Beanstalk-Umgebung implementiert sind. Darüber hinaus benötigen Sie eine Verschlüsselung des end-to-end Datenverkehrs. Wir zeigen eine der Micro-Service-Umgebungen, die einen Hauptprozess für Benutzeranfragen und einen separaten Prozess für die Bearbeitung von administrativen Anforderungen hat.

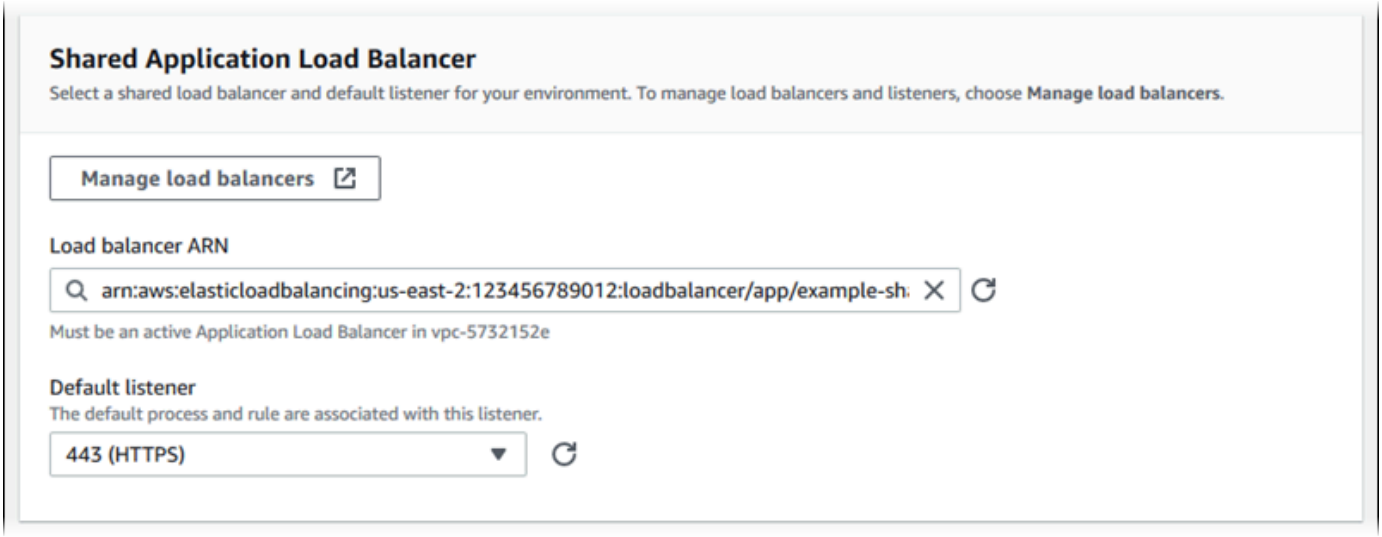
Um diese Anforderungen zu erfüllen, verwenden Sie Amazon, EC2 um einen Application Load Balancer zu erstellen, den Sie unter Ihren Microservices gemeinsam nutzen. Fügen Sie einen

sicheren Listener auf Port 443 und das HTTPS-Protokoll hinzu. Fügen Sie dann dem Listener mehrere SSL-Zertifikate hinzu, eins pro Micro-Service-Domäne. Einzelheiten zur Erstellung des Application Load Balancers und des sicheren Listeners finden Sie unter Application Load Balancer erstellen und HTTPS-Listener für Application Load Balancer erstellen im Application Load Balancer-Benutzerhandbuch.

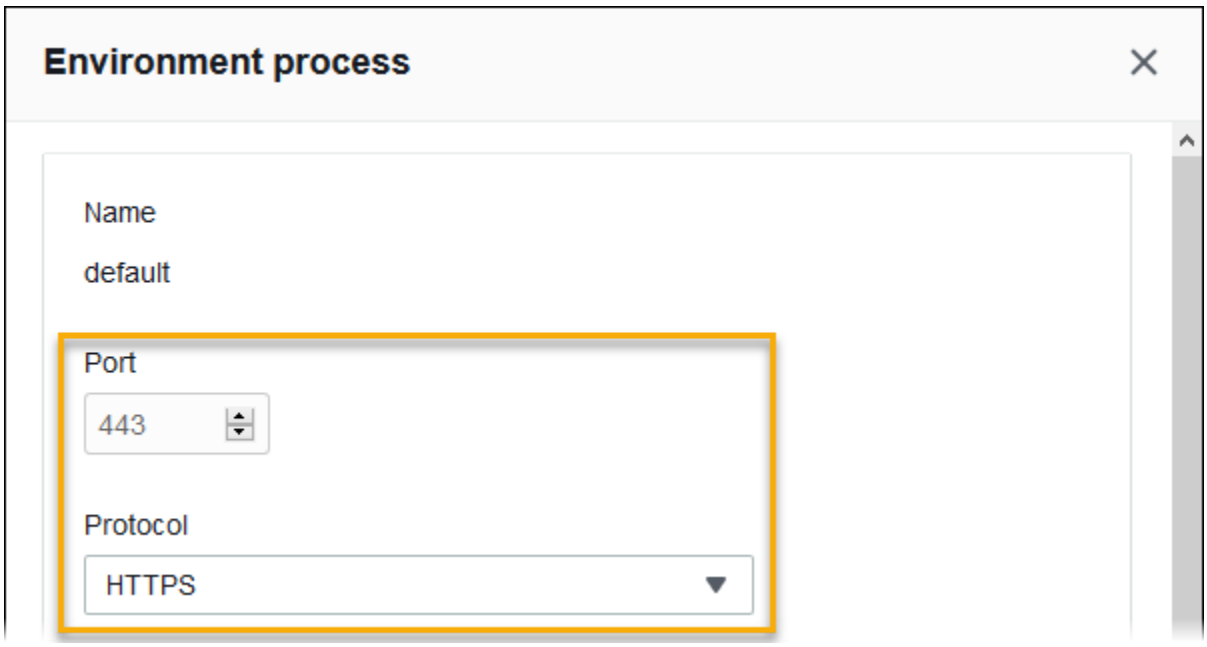
Konfigurieren Sie in Elastic Beanstalk jede Micro-Service-Umgebung für die Verwendung des gemeinsamen Application Load Balancers und setzen Sie den Standard-Listener-Port auf 443. Im Fall der bestimmten Umgebung, die wir hier demonstrieren, geben Sie an, dass der Standardprozess Port 443 auf HTTPS überwacht, und fügen Sie einen Prozess und eine Listener-Regel für den Admin-Datenverkehr auf einem anderen Pfad hinzu.

So konfigurieren Sie den freigegebenen Load Balancer für dieses Beispiel

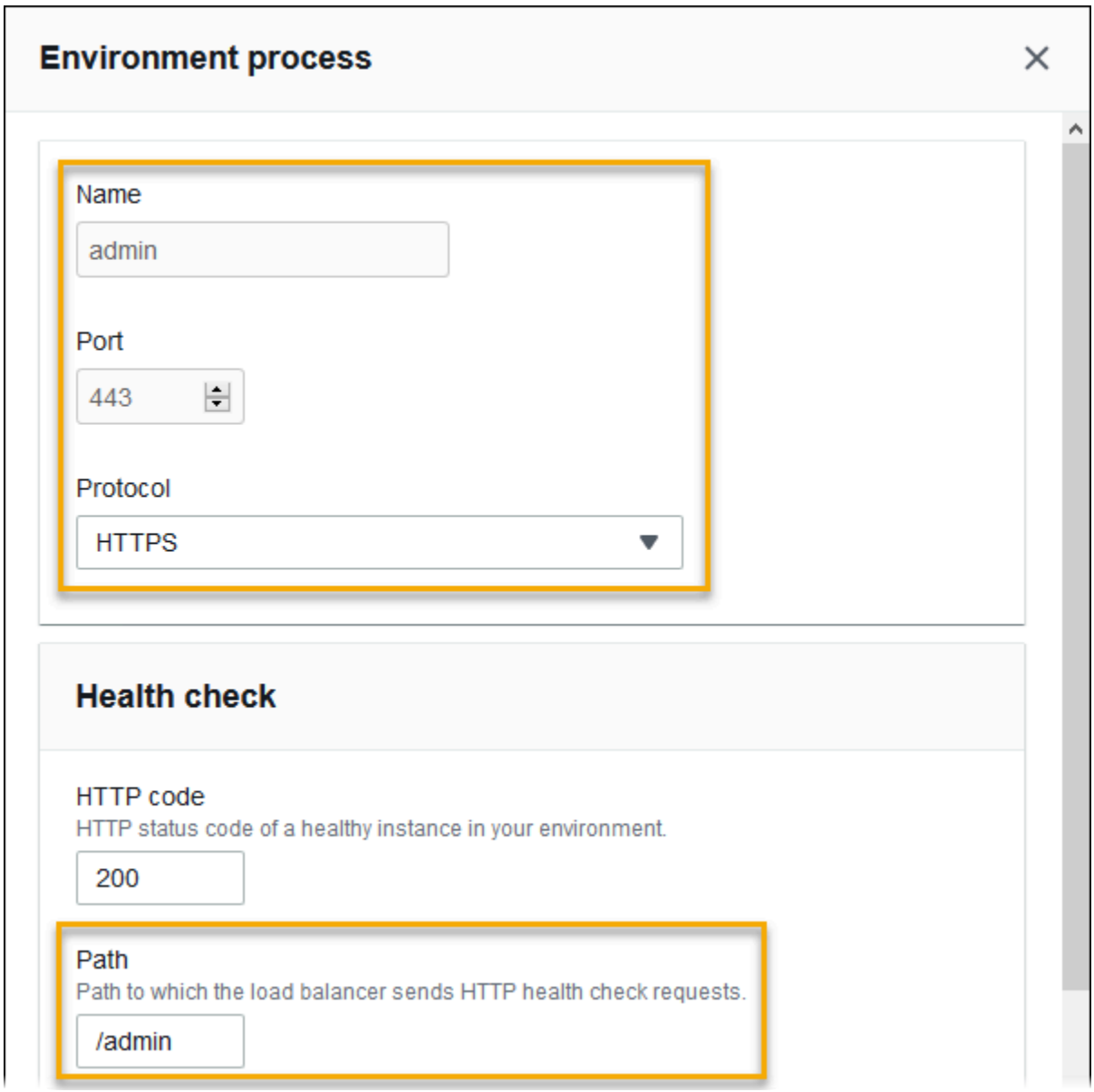
1. Wählen Sie im Abschnitt Shared Application Load Balancer (Gemeinsamer Application Load Balancer) Ihren Load Balancer aus. Wählen Sie dann für Default listener port (Standard-Listener-Port) **443** aus. Der Listener-Port sollte bereits ausgewählt sein, wenn er der einzige Listener ist, den der Load Balancer hat.



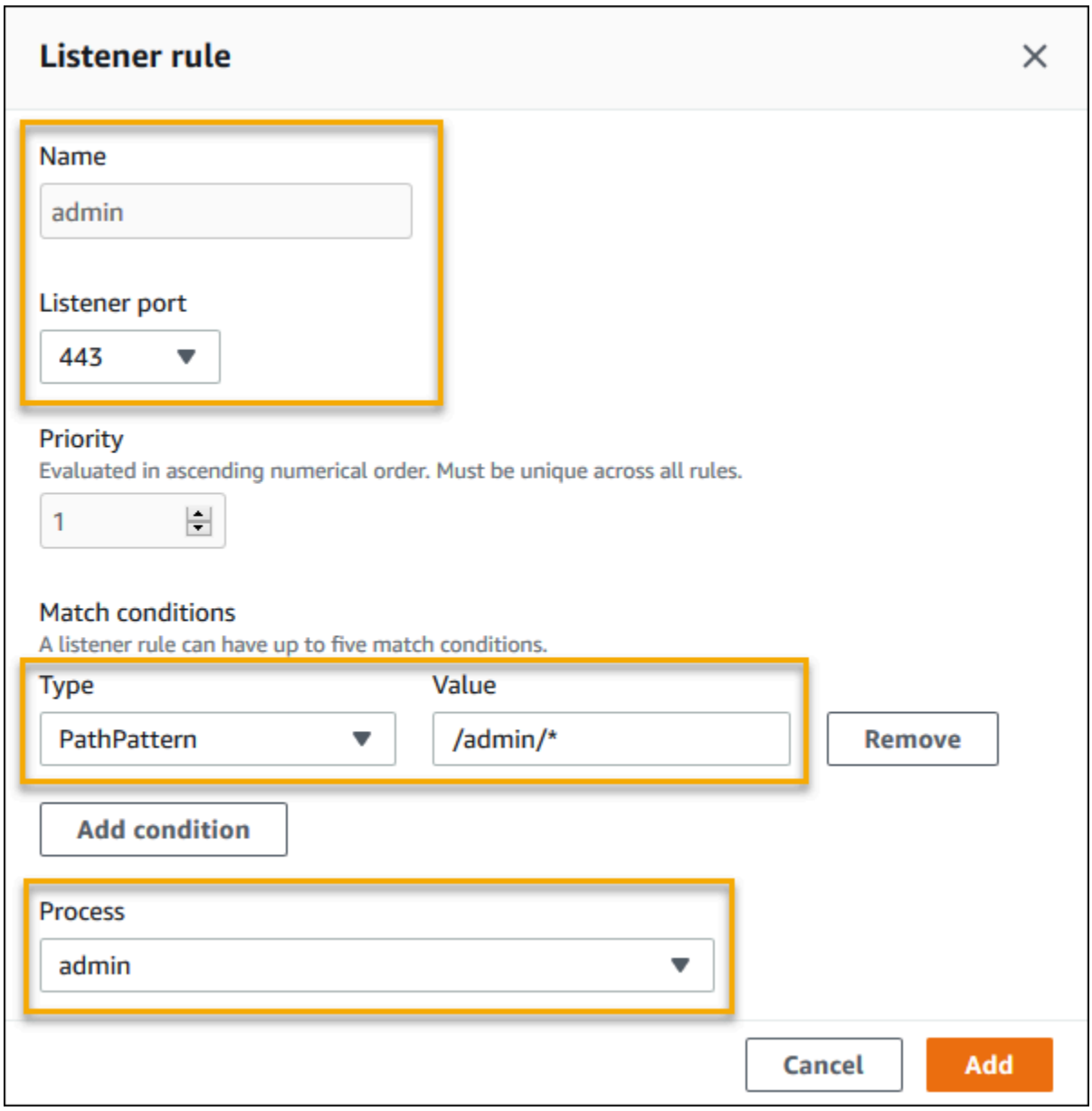
2. Konfigurieren Sie den Standardprozess für HTTPS. Wählen Sie den Standardprozess aus und wählen Sie dann für Actions (Aktionen) die Option Edit (Bearbeiten) aus. Geben Sie im Feld Port **443** ein. Wählen Sie in Protocol (Protokoll) die Option **HTTPS** aus.



3. Fügen Sie einen Administratorprozess hinzu. Geben Sie unter Name **`admin`** ein. Geben Sie im Feld Port **`443`** ein. Wählen Sie in Protocol (Protokoll) die Option **`HTTPS`** aus. Geben Sie in Health check (Zustandsprüfung) in Path (Pfad) **`/admin`** ein.



4. Fügen Sie eine Regel für Administratordatenverkehr hinzu. Geben Sie unter Name **`admin`** ein. Geben Sie in Listener port (Listener-Port) **`443`** ein. Fügen Sie für Match-Bedingungen a PathPatternmit dem Wert **`/admin/*`** hinzu. Wählen Sie in Process (Prozess) **`admin`** aus.



Gemeinsamen Application Load Balancers mit der EB CLI konfigurieren

Sie werden von der EB CLI aufgefordert, einen Load Balancer-Typ auszuwählen, wenn Sie eb create ausführen. Wenn Sie `application` auswählen (die Standardeinstellung), und Ihr Konto über mindestens einen gemeinsamen Application Load Balancer verfügt, fragt die EB CLI Sie, ob Sie einen gemeinsamen Application Load Balancer verwenden möchten. Wenn Sie mit **y** antworten, werden Sie auch aufgefordert, den Load Balancer und den Standardport auszuwählen.

```
$ eb create
Enter Environment Name
```

```
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 2):

Your account has one or more sharable load balancers. Would you like your new
 environment to use a shared load balancer?(y/N) y

Select a shared load balancer
1)MySharedALB1 - arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/
MySharedALB1/6d69caa75b15d46e
2)MySharedALB2 - arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/
MySharedALB2/e574ea4c37ad2ec8
(default is 1): 2

Select a listener port for your shared load balancer
1) 80
2) 100
3) 443
(default is 1): 3
```

Sie können auch einen freigegebenen Load Balancer mit Befehlsoptionen angeben.

```
$ eb create test-env --elb-type application --shared-lb MySharedALB2 --shared-lb-
port 443
```

Gemeinsame Application Load Balancer-Namespaces

Einstellungen für gemeinsame Application Load Balancer finden Sie in den folgenden Namespaces:

- `aws:elasticbeanstalk:environment` – Wählen Sie den Load Balancer-Typ für die Umgebung aus, und teilen Sie Elastic Beanstalk mit dass Sie einen freigegebenen Load Balancer verwenden werden.

  Sie können diese beiden Optionen nicht in Konfigurationsdateien (.Ebextensions) festlegen.
- `aws:elbv2:loadbalancer` – Konfigurieren von ARN und Sicherheitsgruppen für den gemeinsamen Application Load Balancer.

- `aws:elbv2:listener` – Über die Auflistung von Listener-Regeln Listener des gemeinsamen Application Load Balancers den Prozessen der Umgebung zuordnen.
- `aws:elbv2:listenerrule` – Konfigurieren Sie Listener-Regeln, die den Datenverkehr an verschiedene Prozesse weiterleiten, je nach Anfragepfad. Die Regeln sind einzigartig für Application Load Balancer – sowohl für dedizierte als auch für gemeinsame.
- `aws:elasticbeanstalk:environment:process` – Konfigurieren Sie Zustandsprüfungen und geben Sie den Port und das Protokoll für die Prozesse an, die auf den Instances Ihrer Umgebung ausgeführt werden.

Example.ebeextensions/ .config application-load-balancer-shared

Um mit einem gemeinsamen Application Load Balancer zu beginnen, verwenden Sie die Elastic Beanstalk-Konsole, EB CLI oder API, um den Load Balancer-Typ auf `application` festzulegen und sich für die Verwendung eines gemeinsamen Load Balancers zu entscheiden. Verwenden Sie eine Konfigurationsdatei, um den freigegebenen Load Balancer zu konfigurieren.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
```

> **Note**
>
> Sie können diese Option nur während der Erstellung der Umgebung konfigurieren.

Example.ebextensions/ alb-shared-secure-listener .config

Die folgende Konfigurationsdatei wählt einen sicheren Standard-Listener auf Port 443 für den freigegebenen Load Balancer aus und legt fest, dass der Standardprozess Port 443 überwacht.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
  aws:elbv2:listener:443:
    rules: default
  aws:elasticbeanstalk:environment:process:default:
```

```
    Port: '443'
    Protocol: HTTPS
```

Example.ebextensions/ alb-shared-admin-rule .config

Die folgende Konfigurationsdatei baut auf dem vorherigen Beispiel auf und fügt eine Regel hinzu, die den Datenverkehr mit einem Anfragepfad von `/admin` an einen Prozess namens `admin` weiterleitet, der Port 4443 überwacht.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
  aws:elbv2:listener:443:
    rules: default,admin
  aws:elasticbeanstalk:environment:process:default:
    Port: '443'
    Protocol: HTTPS
  aws:elasticbeanstalk:environment:process:admin:
    HealthCheckPath: /admin
    Port: '4443'
    Protocol: HTTPS
  aws:elbv2:listenerrule:admin:
    PathPatterns: /admin/*
    Priority: 1
    Process: admin
```

Konfigurieren eines Network Load Balancers

Wenn Sie Load Balancing aktivieren, ist Ihre AWS Elastic Beanstalk Umgebung mit einem Elastic Load Balancing Load Balancer ausgestattet, der den Traffic auf die Instances in Ihrer Umgebung verteilt. Elastic Load Balancing unterstützt verschiedene Load Balancer-Typen. Weitere Informationen dazu finden Sie im Elastic Load Balancing-Benutzerhandbuch. Elastic Beanstalk kann einen Load Balancer für Sie erstellen oder Sie einen von Ihnen erstellten freigegebenen Load Balancer angeben lassen.

Dieses Thema beschreibt die Konfiguration eines Network Load Balancers, den Elastic Beanstalk erstellt und Ihrer Umgebung zuweist. Weitere Informationen zur Konfiguration aller Load Balancer-Typen, die Elastic Beanstalk unterstützt, finden Sie unter Load Balancer Ihrer Elastic-Beanstalk-Umgebung.

> **Note**
>
> Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern.

Einführung

Mit einem Network Load Balancer akzeptiert der standardmäßige Listener TCP-Anfragen auf Port 80 und leitet sie an die Instances in Ihrer Umgebung weiter. Sie können Zustandsprüfungsverhalten konfigurieren, den Listener-Port konfigurieren oder einen Listener zu einem anderen Port hinzufügen.

> **Note**
>
> Im Gegensatz zu einem Classic Load Balancer oder einem Application Load Balancer kann ein Network Load Balancer keine HTTP- oder HTTPS-Listener der Anwendungsschicht (Schicht 7) haben. Er unterstützt nur TCP-Listener auf Transportebene (Ebene 4). HTTP- und HTTPS-Datenverkehr kann über TCP an Ihre Umgebung weitergeleitet werden. Um sichere HTTPS-Verbindungen zwischen Web-Clients und Ihrer Umgebung aufzubauen, installieren Sie ein selbstsigniertes Zertifikat auf den Instances in der Umgebung und konfigurieren Sie die Instances so, dass sie auf dem entsprechenden Port (typischerweise Port 443) überwachen und HTTPS-Verbindungen beenden. Die Konfiguration variiert je nach Plattform. Detaillierte Anweisungen finden Sie unter Konfiguration der HTTPS-Terminierung auf der Instanz. Anschließend konfigurieren Sie Network Load Balancer so, dass ein Listener hinzugefügt wird, der einem Prozess zugeordnet wird, der auf dem entsprechenden Port überwacht.

Ein Network Load Balancer unterstützt aktive Zustandsprüfungen. Diese Prüfungen basieren auf Nachrichten an den Stamm-Pfad (/). Darüber hinaus unterstützt ein Network Load Balancer passive Zustandsprüfungen. Er erkennt automatisch fehlerhafte Backend-Instances und leitet den Datenverkehr nur an stabile Instances.

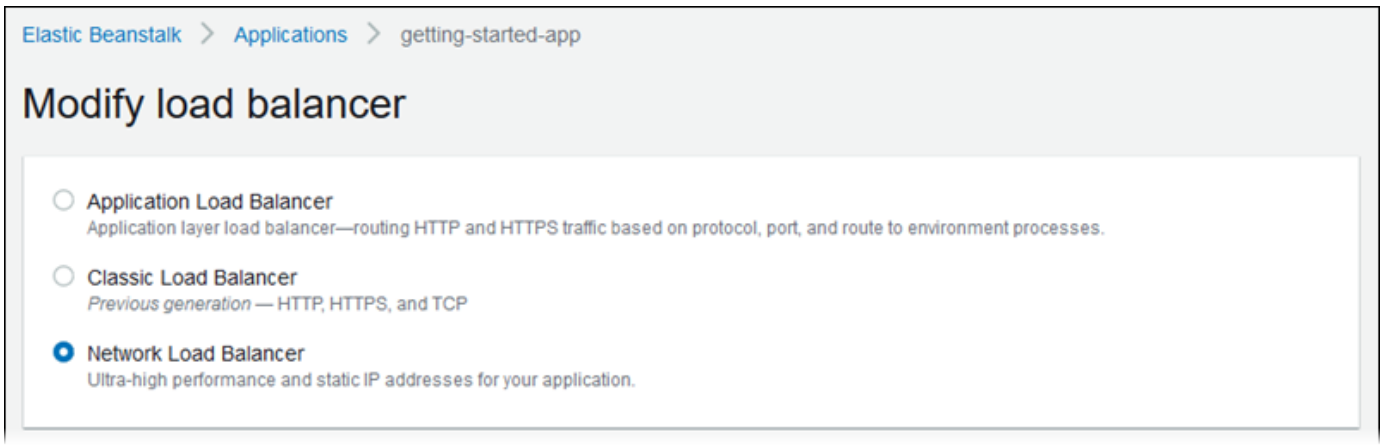
Network Load Balancer mit der Elastic Beanstalk-Konsole konfigurieren

Sie können die Elastic Beanstalk-Konsole verwenden, um die Listener und Prozesse eines Network Load Balancers während der Erstellung der Umgebung oder später (wenn Ihre Umgebung bereits ausgeführt wird) zu konfigurieren.

So konfigurieren Sie einen Network Load Balancer in der Elastic Beanstalk-Konsole während der Erstellung der Umgebung:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus.
3. Wählen Sie Create a new environment (Neue Umgebung erstellen), um mit der Erstellung Ihrer Umgebung zu beginnen.
4. Wählen Sie auf der Hauptseite des Assistenten zuerst Configure more options (Weitere Optionen konfigurieren) und dann Create environment (Umgebung erstellen) aus.
5. Wählen Sie die Konfigurationsvoreinstellung High availability (Hohe Verfügbarkeit) aus.

   Alternativ können Sie in der Konfigurationskategorie Capacity (Kapazität) einen Umgebungstyp mit Load balanced (Lastenverteilung) konfigurieren. Details hierzu finden Sie unter Capacity (Kapazität).
6. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.
7. Wählen Sie die Option Network Load Balancer aus, falls sie nicht bereits ausgewählt ist.



8. Nehmen Sie alle Konfigurationsänderungen am Network Load Balancer vor, die für Ihre Umgebung erforderlich sind.
9. Wählen Sie Save (Speichern) aus und nehmen Sie dann alle weiteren Konfigurationsänderungen für Ihre Umgebung vor.

10. Wählen Sie Create environment (Umgebung erstellen) aus.

So konfigurieren Sie den Network Load Balancer einer laufenden Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.

   > **Note**
   >
   > Ist in der Konfigurationskategorie Load balancer die Schaltfläche Edit (Bearbeiten) nicht vorhanden, verfügt Ihre Umgebung über keinen Load Balancer. Informationen zur Einrichtung finden Sie unter Ändern des Umgebungstyps.

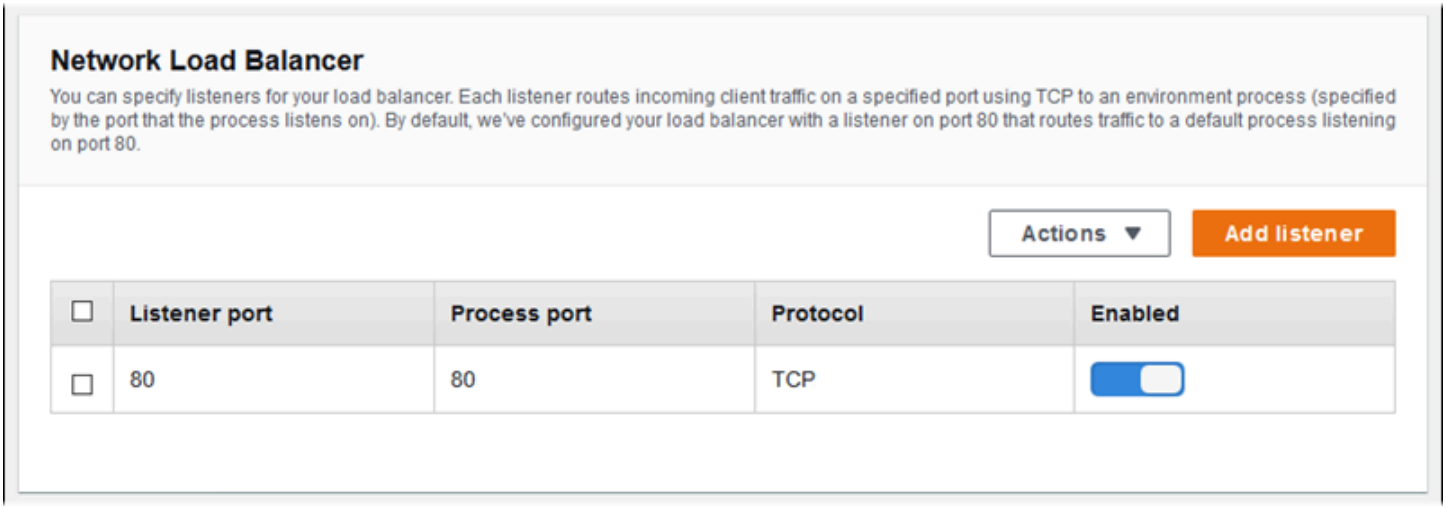
5. Nehmen Sie die Konfigurationsänderungen am Network Load Balancer vor, die Ihre Umgebung erfordert.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Einstellungen für Network Load Balancer

- Listener
- Prozesse

Listener

Verwenden Sie diese Liste, um Listener für Ihren Load Balancer anzugeben. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports an einen Prozess auf Ihren Instances weiter. Anfänglich wird in der Liste der Standard-Listener angezeigt, der eingehenden Datenverkehr auf Port 80 an einen Prozess mit der Bezeichnung default (Standard) weiterleitet, der Port 80 überwacht.



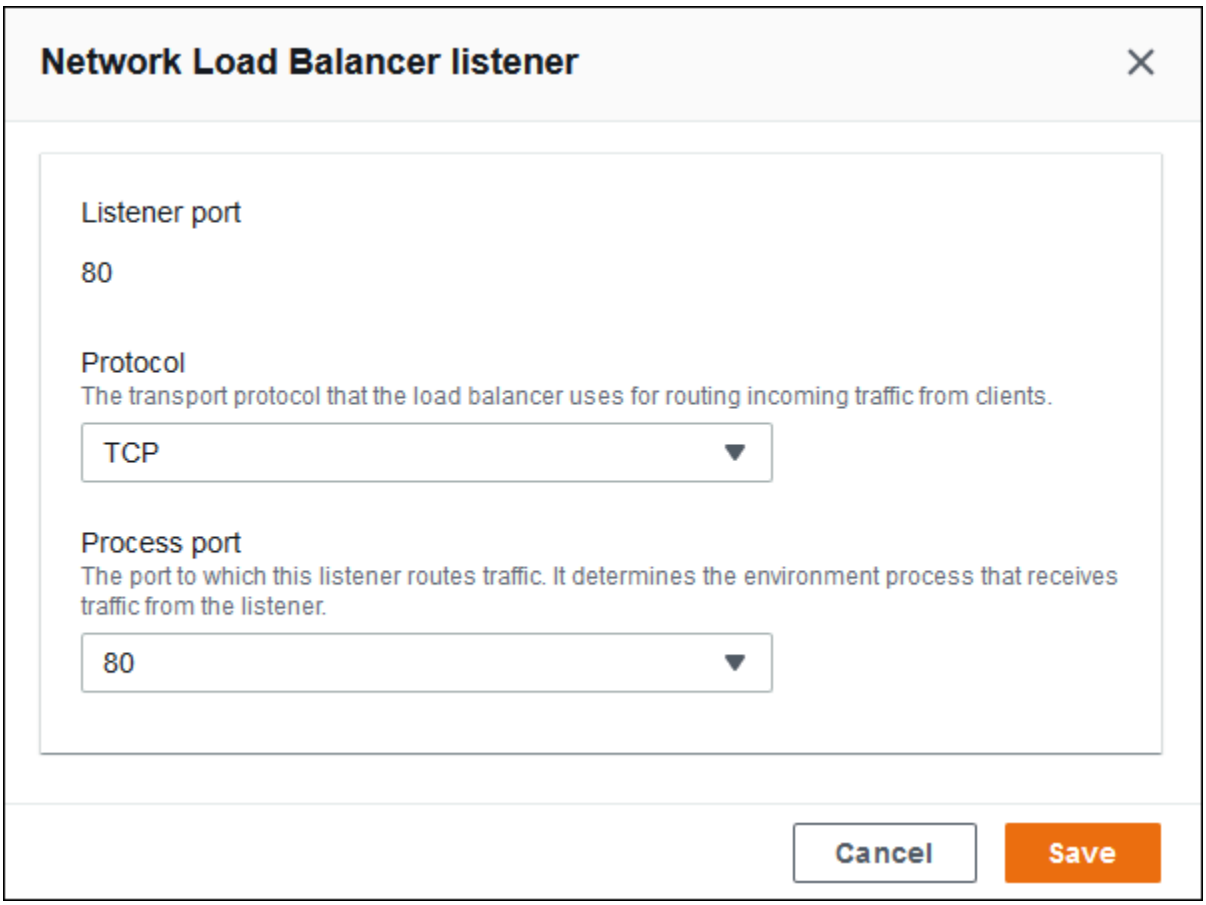
So konfigurieren Sie einen vorhandenen Listener

1. Aktivieren Sie das Kontrollkästchen neben seinem Tabelleneintrag und wählen Sie dann Actions (Aktionen), Edit (Bearbeiten) aus.
2. Verwenden Sie das Dialogfeld Network Load Balancer listener, um die Einstellungen zu bearbeiten, und wählen Sie dann Save (Speichern) aus.

So fügen Sie einen Listener hinzu

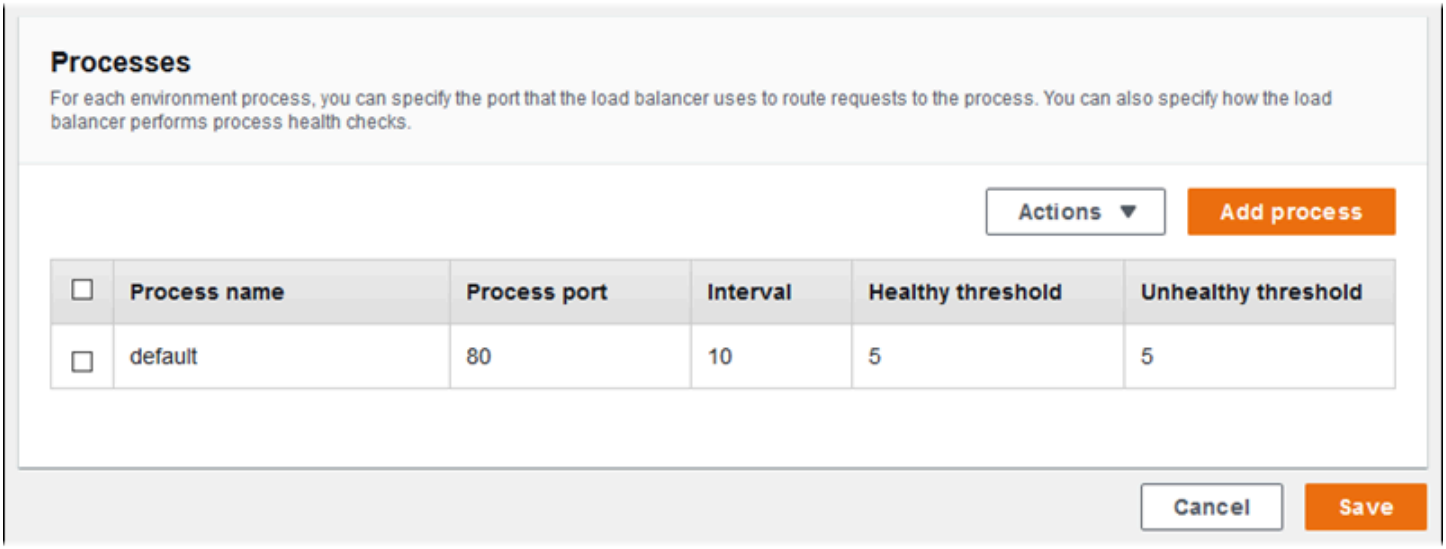
1. Wählen Sie Add listener (Listener hinzufügen) aus.
2. Konfigurieren Sie im Dialogfeld Network Load Balancer listener die erforderlichen Einstellungen und wählen Sie dann Add (Hinzufügen) aus.

Verwenden Sie das Dialogfeld Network Load Balancer listener, um den Port zu konfigurieren, dessen Datenverkehr der Listener überwachen soll, und um den Prozess auszuwählen, an den Sie den Datenverkehr weiterleiten möchten (der durch den Port angegeben wird, den der Prozess überwacht).



Prozesse

Verwenden Sie diese Liste, um Prozesse für Ihren Load Balancer anzugeben. Ein Prozess ist ein Ziel für Listener, an das Datenverkehr weitergeleitet wird. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports an einen Prozess auf Ihren Instances weiter. Anfänglich zeigt die Liste den Standardprozess, der eingehenden Datenverkehr auf Port 80 überwacht.



Sie können die Einstellungen eines vorhandenen Prozesses bearbeiten oder einen neuen Prozess hinzufügen. Um mit dem Bearbeiten eines Prozesses auf der Liste zu beginnen oder Prozesse zur Liste hinzuzufügen, verwenden Sie dieselben Schritte, die für die Liste der Listener aufgelistet sind. Das Dialogfeld Environment process (Umgebungsprozess) wird geöffnet.

Einstellungen des Prozess-Dialogfelds für Network Load Balancer-Umgebung

- Definition
- Gesundheitscheck

Definition

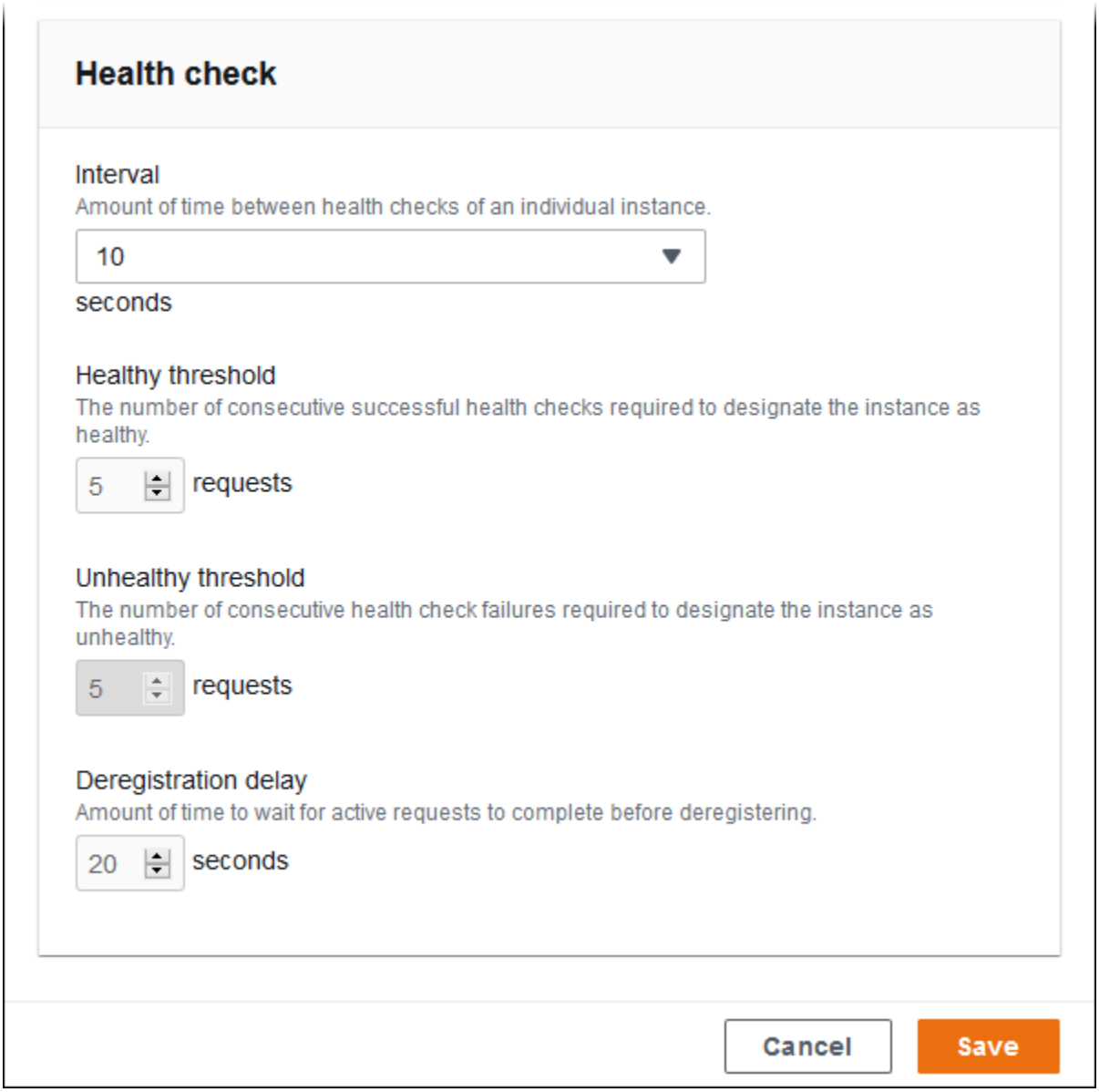
Verwenden Sie diese Einstellungen zum Definieren des Prozesses: Name und Process port (Prozess-Port) für die Überwachung auf Anforderungen.



Gesundheitscheck

Verwenden Sie die folgenden Einstellungen zum Konfigurieren von Zustandsprüfungen für den Prozess:

- Interval (Intervall) – Der Zeitraum in Sekunden zwischen Zustandsprüfungen einer einzelnen Instance
- Healthy threshold (Schwellenwert für fehlerfreien Zustand) – Die Anzahl der Zustandsprüfungen, die bestanden sein müssen, bevor Elastic Load Balancing den Zustand der Instance ändert. (Für Network Load Balancer ist Unhealthy threshold (Schwellenwert für fehlerhaften Zustand) eine schreibgeschützte Einstellung, die immer gleich dem Wert des Schwellenwerts für fehlerfreien Zustand ist.)
- Deregistration delay (Verzögerung der Registrierungsaufhebung) – Die Zeit in Sekunden, die gewartet werden muss, bis aktive Anforderungen abgeschlossen sind, ehe eine Instance abgemeldet wird



> **Note**
>
> Die Zustandsprüfung für Elastic Load Balancing hat keinen Einfluss auf das Verhalten der Zustandsprüfung der Auto Scaling-Gruppe einer Umgebung. Instances, die eine Elastic Load Balancing Balancing-Zustandsprüfung nicht bestehen, werden nicht automatisch durch Amazon EC2 Auto Scaling ersetzt, es sei denn, Sie konfigurieren Amazon EC2 Auto Scaling manuell entsprechend. Details dazu finden Sie unter Einstellung für die Auto Scaling Scaling-Integritätsprüfung für Ihre Elastic Beanstalk Beanstalk-Umgebung.

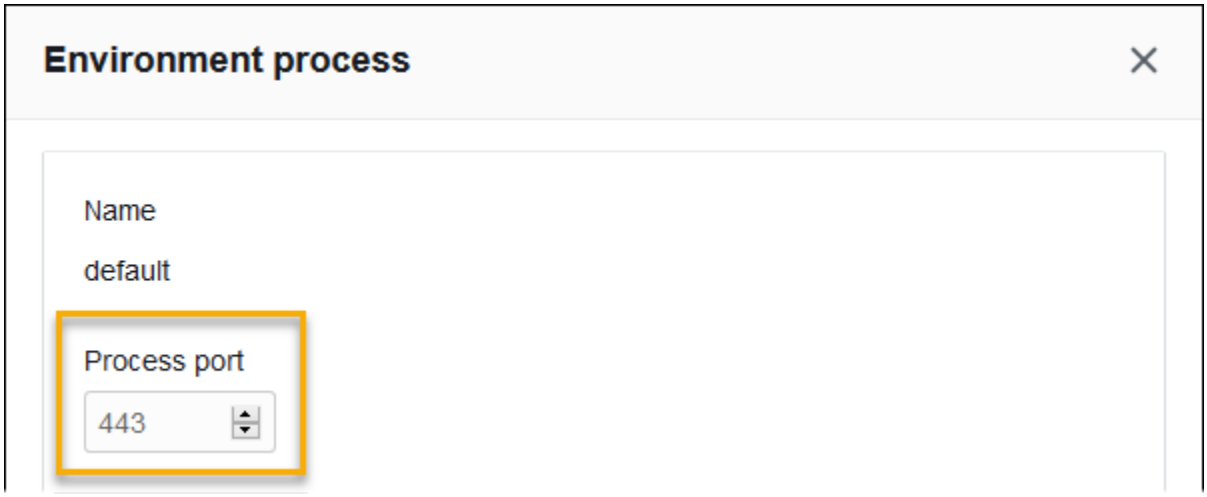
Weitere Informationen zu Zustandsprüfungen und deren Auswirkungen auf den Gesamtstatus der Umgebung finden Sie unter Grundlegende Zustandsberichte.

Beispiel: Network Load Balancer für eine Umgebung mit Verschlüsselung end-to-end

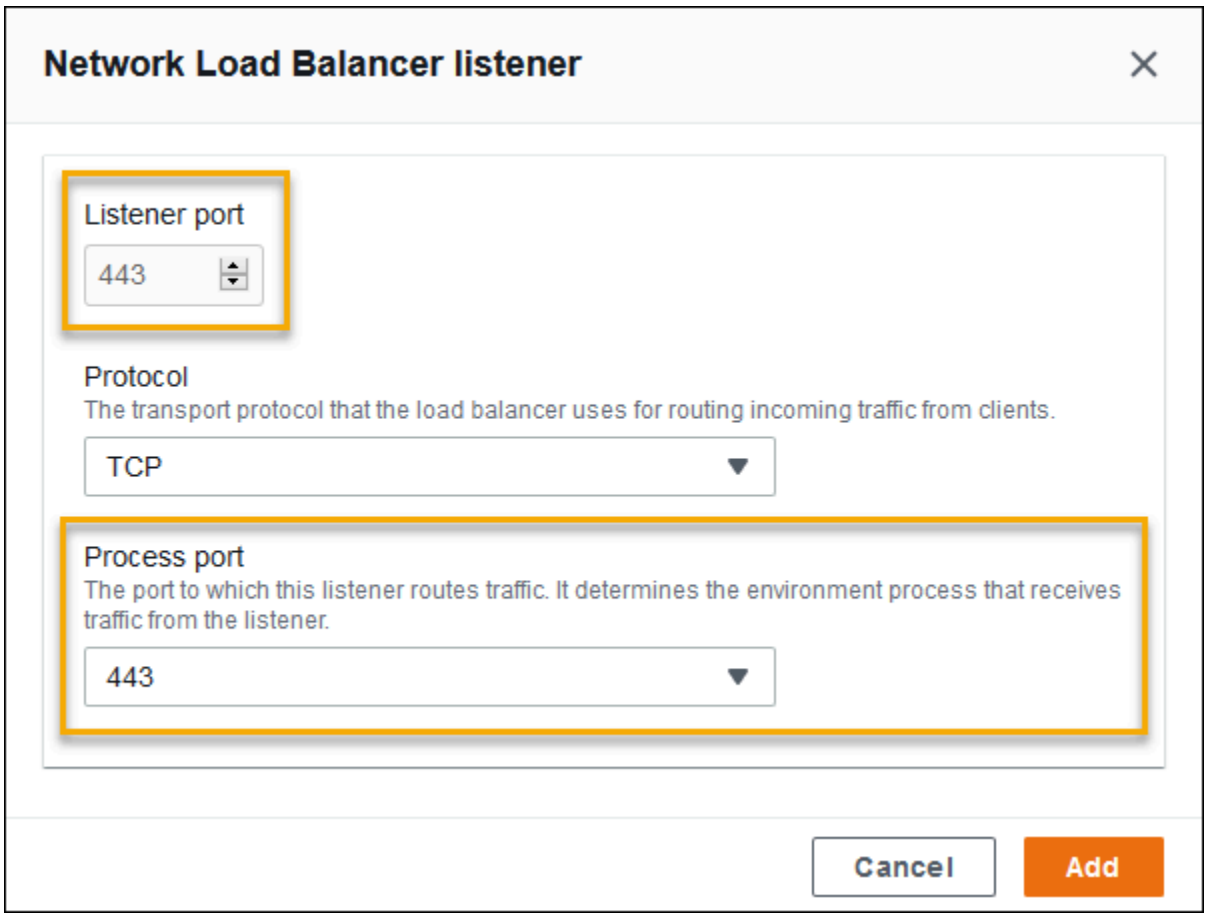
In diesem Beispiel erfordert Ihre Anwendung eine Verschlüsselung des end-to-end Datenverkehrs. Um Network Load Balancer Ihrer Umgebung so zu konfigurieren, dass er diese Anforderungen erfüllt, konfigurieren Sie den Standardprozess so, dass er Port 443 überwacht, fügen Sie einen Listener zu Port 443 hinzu, der den Datenverkehr an den Standardprozess weiterleitet, und deaktivieren Sie den Standard-Listener.

So konfigurieren Sie den Load Balancer für dieses Beispiel

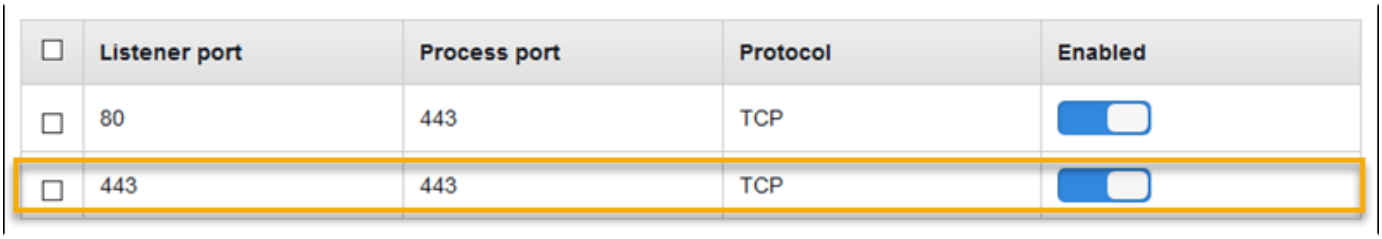
1. Konfigurieren Sie den Standardprozess. Wählen Sie den Standardprozess aus und wählen Sie dann für Actions (Aktionen) die Option Edit (Bearbeiten) aus. Für Process port (Prozess-Port) geben Sie `443` ein.



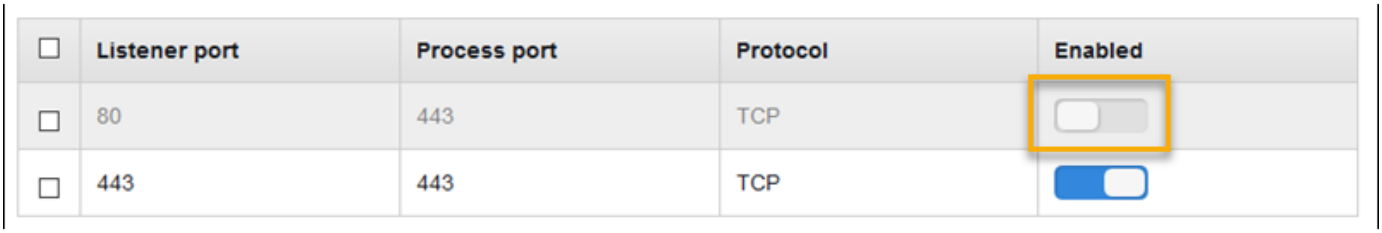
2. Fügen Sie einen Listener auf Port 443 hinzu. Fügen Sie einen neuen Listener hinzu. Geben Sie für Listener port (Listener-Port) den Wert `443` ein. Stellen Sie bei Process port (Prozess-Port) sicher, dass `443` ausgewählt ist.



Sie können nun Ihre zusätzlichen Listener auf der Liste sehen.



3. Deaktivieren Sie den Listener am Standardport 80. Deaktivieren Sie für den Standard-Listener die Option Enabled (Aktiviert).



Network Load Balancer mit der EB CLI konfigurieren

Sie werden von der EB CLI aufgefordert, einen Load Balancer-Typ auszuwählen, wenn Sie eb create ausführen.

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 1): 3
```

Sie können auch einen Load Balancer-Typ mit der `--elb-type`-Option angeben.

```
$ eb create test-env --elb-type network
```

Network Load Balancer-Namespaces

Sie finden Einstellungen im Zusammenhang mit Network Load Balancers in den folgenden Namespaces:

- `aws:elasticbeanstalk:environment` – Wählen Sie den Load Balancer-Typ für die Umgebung aus. Der Wert für einen Network Load Balancer ist `network`.
- `aws:elbv2:listener` – Konfigurieren Sie Listener für den Network Load Balancer. Diese Einstellungen entsprechen den Einstellungen in `aws:elb:listener` für Classic Load Balancer.
- `aws:elasticbeanstalk:environment:process` – Konfigurieren Sie Zustandsprüfungen und geben Sie den Port und das Protokoll für die Prozesse an, die auf den Instances Ihrer Umgebung ausgeführt werden. Die Port- und Protokolleinstellungen entsprechen den Instance-Port und Instance-Protokoll-Einstellungen in `aws:elb:listener` für einen Listener auf einem Classic Load Balancer. Zustandsprüfungseinstellungen entsprechen den Einstellungen in den Namespaces `aws:elb:healthcheck` und `aws:elasticbeanstalk:application`.

Example.ebeextensions/ .config network-load-balancer

Für die ersten Schritte mit einem Network Load Balancer verwenden Sie eine Konfigurationsdatei, um den Load Balancer-Typ auf `network` festzulegen.

```
option_settings:
  aws:elasticbeanstalk:environment:
    LoadBalancerType: network
```

> **Note**
>
> Sie können den Load Balancer-Typ nur während der Umgebungserstellung festlegen.

Example.ebextensions/ nlb-default-process .config

Die folgende Konfigurationsdatei ändert Zustandsprüfungseinstellungen auf dem Standardprozess.

```
option_settings:
  aws:elasticbeanstalk:environment:process:default:
    DeregistrationDelay: '20'
    HealthCheckInterval: '10'
    HealthyThresholdCount: '5'
    UnhealthyThresholdCount: '5'
    Port: '80'
    Protocol: TCP
```

Example.ebextensions/ nlb-secure-listener .config

Die folgende Konfigurationsdatei fügt einen Listener für sicheren Datenverkehr auf Port 443 sowie einen entsprechenden Zielprozess hinzu, der Port 443 überwacht.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
```

Die Option `DefaultProcess` hat diesen Namen wegen Application Load Balancers, die Nicht-Standard-Listener auf demselben Port für Datenverkehr auf bestimmten Pfaden haben können (siehe

Application Load Balancer, um weitere Informationen zu erhalten). Bei einem Network Load Balancer gibt die Option den einzigen Zielprozess für diesen Listener an.

In diesem Beispiel haben wir den Prozess `https` genannt, da er sicheren (HTTPS) Datenverkehr überwacht. Der Listener sendet den Datenverkehr über das TCP-Protokoll an den Prozess auf dem festgelegten Port, da ein Network Load Balancer nur mit TCP funktioniert. Das ist in Ordnung, da der Netzwerkdatenverkehr für HTTP und HTTPS zusätzlich zu TCP implementiert ist.

Konfiguration von Elastic Beanstalk Beanstalk-Loadbalancern mit zwei Stacks

Sie können Ihre Elastic Beanstalk Beanstalk-Umgebungen so einrichten, dass sie sowohl IPv4 Protokolle als auch IPv6 Protokolle mit Dual-Stack-konfigurierten Load Balancern bereitstellen. Wenn Sie eine Umgebung mit Lastenausgleich erstellen, ist die Infrastruktur standardmäßig auf. IPv4 Sie können wählen, ob Sie neue Umgebungen mit Dual-Stack-Konfiguration erstellen oder bestehende IPv4 Nur-Umgebungen auf Dual-Stack-Umgebungen aktualisieren möchten.

Damit die Load Balancer Ihrer Umgebung sowohl IPv6 den Netzwerkverkehr als auch den IPv4 Netzwerkverkehr bedienen können, setzen Sie die *`IpAddressType`* Option im Namespace auf Dual-Stack. aws:elbv2:loadbalancer

> **Note**
>
> Elastic Beanstalk unterstützt nur Dual-Stack-Protokolle für Application Load Balancer und Network Load Balancer. Dual-Stack wird nicht für Umgebungen unterstützt, die Classic Load Balancers oder Single-Instance-Umgebungen verwenden.

Voraussetzungen für Amazon VPC

Bevor Sie Ihren Load Balancer in Ihrer Elastic Beanstalk Beanstalk-Umgebung konfigurieren, müssen Sie zunächst einige Konfigurationsschritte mit Amazon VPC durchführen:

1. Ordnen Sie der VPC Ihrer Umgebung einen IPv6 CIDR-Block zu.
2. Ordnen Sie IPv6 CIDR-Blöcke allen Subnetzen der VPC zu.
3. (Optional) Wenn Ihre Umgebung Netzwerkverkehr mit Komponenten außerhalb der VPC austauscht, verfügt sie über Routing-Tabellen, die angeben, mit welchen Netzwerken Ihre VPC

kommunizieren kann. In diesem Fall müssen Sie die VPC-Routentabellen aktualisieren, um den IPv6 Verkehr zu aktivieren.

Sie können diese vorausgesetzten Konfigurationen entweder mit der Amazon VPC-Konsole oder mit AWS CLI Befehlen abschließen. Die folgenden Themen führen Sie als Leitfaden und leiten Sie zum Amazon VPC-Benutzerhandbuch und zur AWS CLI Befehlsreferenz weiter, wo Sie weitere Informationen finden.

> **Note**
>
> Warten Sie nach Abschluss der VPC-Konfiguration einige Minuten, bis die Änderungen wirksam werden, bevor Sie Dual-Stack für Ihren Load Balancer konfigurieren. Wenn beim Dual-Stack-Setup VPC- oder Subnetzkonfigurationsfehler auftreten, warten Sie einige Minuten, bis sich die VPC-Konfiguration verbreitet hat, und versuchen Sie es erneut mit der Dual-Stack-Konfiguration.

Erfüllen Sie die VPC-Voraussetzungen mithilfe der Konsole

Das Amazon VPC-Benutzerhandbuch enthält detaillierte Schritte zur Ausführung dieser erforderlichen Aufgaben.

1. Weitere Informationen finden Sie unter Schritt 1: Zuordnen eines IPv6 CIDR-Blocks zu Ihrer VPC und Ihren Subnetzen im Amazon VPC-Benutzerhandbuch.

   Dieser Schritt umfasst zwei Verfahren, die Sie ausführen müssen:

   - Ordnen Sie Ihrer VPC einen IPv6 CIDR-Block zu.
   - Ordnen Sie Ihren VPC-Subnetzen einen IPv6 CIDR-Block zu.

2. (Optional) Wenn Ihre Umgebung Netzwerkverkehr mit Komponenten außerhalb der VPC austauscht, verfügt sie über Routing-Tabellen, die angeben, mit welchen Netzwerken Ihre VPC kommunizieren kann. In diesem Fall müssen Sie die VPC-Routentabellen aktualisieren, um den IPv6 Verkehr zu aktivieren. Informationen zum Abschließen dieser Konfiguration finden Sie unter Schritt 2: Aktualisieren Sie Ihre Routing-Tabellen im Amazon VPC-Benutzerhandbuch.

Erfüllen Sie die VPC-Voraussetzungen mit dem AWS CLI

Sie können den verwenden AWS CLI , um die erforderlichen Konfigurationen abzuschließen und zu überprüfen.

1. Ordnen Sie der VPC Ihrer Umgebung einen IPv6 CIDR-Block zu.
   a. Verwenden Sie den associate-vpc-cidr-blockBefehl, um Ihrer VPC einen CIDR-Block zuzuordnen.
   b. Verwenden Sie den Befehl describe-vpcs, um Ihre VPC-Konfigurationen zu überprüfen.

   Example commands

```
# Associate an Amazon-provided IPv6 CIDR block with your VPC
aws ec2 associate-vpc-cidr-block \
    --vpc-id vpc-12345678 \
    --region us-east-1  \
    --amazon-provided-ipv6-cidr-block

# Verify the IPv6 CIDR block association
aws ec2 describe-vpcs \
    --vpc-ids vpc-12345678 \
    --region us-east-1  \
    --query 'Vpcs[0].Ipv6CidrBlockAssociationSet'
```

2. Ordnen Sie IPv6 CIDR-Blöcke allen Subnetzen der VPC zu.
   a. Verwenden Sie den associate-subnet-cidr-blockBefehl, um Ihrem Subnetz einen CIDR-Block zuzuordnen.
   b. Verwenden Sie den Befehl describe-subnets, um Ihre Subnetzkonfigurationen zu überprüfen.

   Example commands

```
# List all subnets in your VPC
aws ec2 describe-subnets \
    --region us-east-1  \
    --filters "Name=vpc-id,Values=vpc-12345678" \
    --query 'Subnets[].{SubnetId:SubnetId,AvailabilityZone:AvailabilityZone}'

# Associate IPv6 CIDR block with each subnet
```

```
aws ec2 associate-subnet-cidr-block \
    --subnet-id subnet-12345678 \
    --region us-east-1  \
    --ipv6-cidr-block 2001:db8::/64

# Verify IPv6 CIDR block association for all subnets
aws ec2 describe-subnets \
    --region us-east-1  \
    --filters "Name=vpc-id,Values=vpc-12345678" \
    --query 'Subnets[].
{SubnetId:SubnetId,Ipv6CidrBlock:Ipv6CidrBlockAssociationSet[0].Ipv6CidrBlock}'
```

3. (Optional) Wenn Ihre Umgebung Netzwerkverkehr mit Komponenten außerhalb der VPC austauscht, verfügt sie über Routing-Tabellen, die angeben, mit welchen Netzwerken Ihre VPC kommunizieren kann. In diesem Fall müssen Sie die VPC-Routentabellen aktualisieren, um den IPv6 Verkehr zu aktivieren.

   a. Verwenden Sie den Befehl create-route, um eine Route zu einer Routentabelle innerhalb der VPC hinzuzufügen.

   b. Verwenden Sie den describe-route-tablesBefehl, um Ihre Routentabellen zu überprüfen.

   Example commands

```
# Add IPv6 route to Internet Gateway for public subnets
aws ec2 create-route \
    --route-table-id rtb-12345678 \
    --destination-ipv6-cidr-block ::/0 \
    --gateway-id igw-12345678  \
    --region us-east-1

# Add IPv6 route to NAT Gateway for private subnets (if applicable)
aws ec2 create-route \
    --route-table-id rtb-87654321 \
    --destination-ipv6-cidr-block ::/0 \
    --nat-gateway-id nat-12345678 \
    --region us-east-1

# Verify routes
aws ec2 describe-route-tables \
    --route-table-ids rtb-12345678 \
    --region us-east-1 \
```

```
    --query 'RouteTables[0].Routes'
```

Konfiguration von Dual-Stack für Ihren Elastic Beanstalk Load Balancer

Nachdem Ihre VPC-Grundkonfiguration für Ihre Umgebung eingerichtet wurde, können Sie den Load Balancer mit der Dual-Stack-Option konfigurieren, sodass er sowohl die als auch die IPv4 Protokolle bedienen kann. IPv6 Sie können die Elastic Beanstalk Beanstalk-Konsole, die Konfigurationsdateien in und das AWS SDK verwenden AWS CLI, um den Load Balancer so zu konfigurieren `.ebextensions`, dass er Dual-Stack-Traffic bedient.

Verwenden der Konsole

Sie können die Elastic Beanstalk Beanstalk-Konsole verwenden, um Dual-Stack für den Load Balancer Ihrer Umgebung zu konfigurieren.

> **Note**
>
> Diese Konfiguration hängt vom Zeitpunkt der Datenweitergabe an mehreren Punkten ab. Beachten Sie die folgenden zeitlichen Anforderungen, wenn Sie Ihren Load Balancer konfigurieren und die Änderungen testen.
> VPC-Konfigurationszeit. Nachdem Sie die unter beschriebene VPC-Konfiguration abgeschlossen habenVoraussetzungen für Amazon VPC, warten Sie einige Minuten, bis die Änderungen übernommen wurden, bevor Sie diese Verfahren ausführen. Wenn beim Dual-Stack-Setup VPC- oder Subnetzkonfigurationsfehler auftreten, warten Sie einige Minuten und führen Sie die Konfigurationsschritte erneut aus.
> DNS-Übertragungszeit. Nachdem Sie die `IpAddressType` Option auf Dualstack gesetzt haben, warten Sie einige Minuten, bis die Änderungen wirksam werden, bevor Sie den Test durchführen. Es kann 1—2 Minuten dauern, bis die DNS-Propagierung für Route 53 abgeschlossen ist. Während dieser Zeit können beim Testen der Kommunikation zwischen einem Client und Ihrer Anwendung Probleme auftreten, wenn Sie Anfragen mithilfe des IPv6 Protokolls initiieren.

Neue Umgebung erstellen: Um Ihren Load Balancer für Dual-Stack-Unterstützung zu konfigurieren

1. Starten Sie die Elastic Beanstalk Beanstalk-Konsole und beginnen Sie mit den Schritten zum Erstellen einer neuen Umgebung. Nachdem Sie auf der Seite „Servicezugriff konfigurieren“ die erforderlichen Felder „Servicerolle“ und „EC2-Instanzprofil“ festgelegt haben, fahren Sie mit den

Schritten in diesem Verfahren fort, um Ihren Load Balancer auf die Dual-Stack-Konfiguration einzustellen. Weitere Informationen zu den ersten Schritten finden Sie unter. Erstellen einer Elastic Beanstalk-Umgebung

2. Wählen Sie unter Dienstzugriff konfigurieren die Option Weiter aus.
3. Die Seite Netzwerk, Datenbank und Tags einrichten wird angezeigt.

   Wenn Sie die im vorherigen Abschnitt Voraussetzungen für Amazon VPC beschriebenen Schritte abgeschlossen haben, haben Sie die erforderliche VPC und Subnetze bereits eingerichtet. In diesem Fall überspringen Sie diesen Schritt zusammen mit seinen Unterschritten, um mit der Auswahl der VPC fortzufahren.

   a. Um die VPC und die Subnetze zu konfigurieren, können Sie Create VPC auswählen, um zur VPC-Konsole zu navigieren. Führen Sie die Schritte unter Erfüllen Sie die VPC-Voraussetzungen mithilfe der Konsole aus.

   b. Warten Sie einige Minuten, bis sich die VPC-Updates verbreitet haben. Kehren Sie dann zur Elastic Beanstalk Beanstalk-Konsole zurück und wählen Sie Aktualisieren, um mit dem nächsten Schritt fortzufahren.

      Wenn Sie in den verbleibenden Schritten auf VPC- oder Subnetzkonfigurationsfehler stoßen, warten Sie einige Minuten, bis sich die VPC-Konfiguration verbreitet hat, und wiederholen Sie die Schritte.

4. Wählen Sie auf der Seite Netzwerk, Datenbank und Tags einrichten einen Wert aus der VPC-Dropdownliste aus, dem ein IPv6 CIDR-Block zugeordnet ist.

   Nachdem Sie eine VPC ausgewählt haben, werden die Instance-Subnetze mit den VPC-Subnetzen gefüllt.

5. Wählen Sie ein oder mehrere Instance-Subnetze aus und klicken Sie dann auf Weiter.
6. Die Seite Instance-Verkehr und Skalierung konfigurieren wird angezeigt.

   Wählen Sie in den Load Balancer-Netzwerkeinstellungen die Option Enable for Dualstack (IPv4 &) aus. IPv6

7. Wählen Sie den Load Balancer-Typ aus. Sowohl der Application Load Balancer als auch der Network Load Balancer unterstützen Dual-Stack.
8. Sie können weitere Load Balancer-Optionen auf der aktuellen Konsolenseite konfigurieren. Weitere Informationen zu den Optionen und der Konfiguration des Load Balancers finden Sie in den anderen Themen in diesem Kapitel.

9. Fahren Sie mit den Schritten fort, um die Konfiguration und Erstellung Ihrer Umgebung abzuschließen. Weitere Informationen finden Sie unter Erstellen einer Elastic Beanstalk-Umgebung.

Bestehende Umgebung: Um Ihren Load Balancer für Dual-Stack-Unterstützung zu konfigurieren

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Kategorie Netzwerk- und Datenbankkonfiguration die Option Bearbeiten aus.
5. Wenn Sie die im vorherigen Abschnitt Voraussetzungen für Amazon VPC beschriebenen Schritte abgeschlossen haben, haben Sie die erforderliche VPC und Subnetze bereits eingerichtet. In diesem Fall überspringen Sie diesen Schritt zusammen mit seinen Unterschritten, um mit der Auswahl der VPC fortzufahren.

   a. Um die VPC und die Subnetze zu konfigurieren, können Sie Create VPC auswählen, um zur VPC-Konsole zu navigieren. Führen Sie die Schritte unter Erfüllen Sie die VPC-Voraussetzungen mithilfe der Konsole aus.

   b. Warten Sie einige Minuten, bis sich die VPC-Updates verbreitet haben. Kehren Sie dann zur Elastic Beanstalk Beanstalk-Konsole zurück und wählen Sie Aktualisieren, um mit dem nächsten Schritt fortzufahren.

      Wenn Sie in den verbleibenden Schritten auf VPC- oder Subnetzkonfigurationsfehler stoßen, warten Sie einige Minuten, bis sich die VPC-Konfiguration verbreitet hat, und wiederholen Sie die Schritte.
6. Wählen Sie auf der Seite Netzwerk und Datenbank einen Wert aus der VPC-Dropdownliste aus, dem ein IPv6 CIDR-Block zugeordnet ist.

   Nachdem Sie eine VPC ausgewählt haben, werden die Instance-Subnetze mit den VPC-Subnetzen gefüllt.
7. Wählen Sie ein oder mehrere Instance-Subnetze aus.
8. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

9. Wählen Sie in der Konfigurationskategorie Instance traffic and scaling (Instance, Datenverkehr und Skalierung) die Option Edit (Bearbeiten) aus.
10. Wählen Sie in den Load Balancer-Netzwerkeinstellungen die Option Enable for Dualstack (&). IPv4 IPv6
11. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Verwenden von AWS CLI

Sie können die AWS Befehlszeilenschnittstelle (AWS CLI) verwenden, um die Load Balancer Ihrer Umgebung so zu konfigurieren, dass sie IPv6 sowohl den Netzwerkverkehr als auch den IPv4 Netzwerkverkehr bedienen. Dieser Abschnitt enthält Beispiele für die Befehle create-environment und update-environment mit dem Namespace. aws:elbv2:loadbalancer

> **Note**
>
> Diese Konfiguration hängt vom Zeitpunkt der Datenweitergabe an mehreren Punkten ab. Beachten Sie die folgenden zeitlichen Anforderungen, wenn Sie Ihren Load Balancer konfigurieren und die Änderungen testen.
> VPC-Konfigurationszeit. Nachdem Sie die unter beschriebene VPC-Konfiguration abgeschlossen habenVoraussetzungen für Amazon VPC, warten Sie einige Minuten, bis die Änderungen wirksam werden, bevor Sie diese Befehle ausführen, um Ihren Load Balancer für Dual-Stack zu konfigurieren. Wenn beim Dual-Stack-Setup VPC- oder Subnetzkonfigurationsfehler auftreten, warten Sie einige Minuten und versuchen Sie es erneut mit den Befehlen.
> DNS-Übertragungszeit. Nachdem Sie die `IpAddressType` Option auf Dualstack gesetzt haben, warten Sie einige Minuten, bis die Änderungen wirksam werden, bevor Sie den Test durchführen. Es kann 1—2 Minuten dauern, bis die DNS-Propagierung für Route 53 abgeschlossen ist. Während dieser Zeit können beim Testen der Kommunikation zwischen einem Client und Ihrer Anwendung Probleme auftreten, wenn Sie Anfragen mithilfe des IPv6 Protokolls initiieren.

Example von Create-Environment mit Dual-Stack-Konfiguration (Inline-Namespace-Optionen)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
```

```
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.0 running Python 3.8" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \
Namespace=aws:elbv2:loadbalancer,OptionName=IpAddressType,Value=dualstack
```

Verwenden Sie alternativ eine`options.json`-Datei, um die Namespace-Optionen anzugeben, anstatt sie einzubinden. Der folgende Beispielbefehl veranschaulicht den Befehl update-environment.

Example von update-environment mit Dual-Stack-Konfiguration (Namespace-Optionen in der Datei) **`options.json`**

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.0 running Python 3.8" \
--option-settings \ file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:elbv2:loadbalancer",
    "OptionName": "IpAddressType",
    "Value": "dualstack"
  }
]
```

Im folgenden Beispiel wird eine bestehende Umgebung aktualisiert, sodass die Option auf gesetzt wird. `IpAddressType` IPv4

> **Note**
>
> Dieser `update-environment` Beispielbefehl ist nützlich, wenn Sie Ihre Umgebungskonfiguration von Dual-Stack auf IPv4 zurücksetzen müssen.

Example der Update-Umgebung, auf die gesetzt werden soll IpAddressType IPv4

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.0 running Python 3.8" \
--option-settings \
Namespace=aws:elbv2:loadbalancer,OptionName=IpAddressType,Value=ipv4
```

Verwenden von Konfigurationsdateien von.ebextensions

Sie können Elastic Beanstalk Beanstalk-Konfigurationsdateien verwenden, damit die Load Balancer Ihrer Umgebung sowohl IPv6 den Netzwerkverkehr als auch den Netzwerkverkehr bedienen können. IPv4 Setzen Sie die *`IpAddressType`* Option im aws:elbv2:loadbalancer Namespace auf dualstack.

> **Note**
>
> Diese Konfiguration hängt vom Zeitpunkt der Datenweitergabe an mehreren Punkten ab. Beachten Sie die folgenden zeitlichen Anforderungen, wenn Sie Ihren Load Balancer konfigurieren und die Änderungen testen.
> VPC-Konfigurationszeit. Warten Sie nach Abschluss der unter beschriebenen VPC-Konfiguration einige MinutenVoraussetzungen für Amazon VPC, bis sich die VPC-Änderungen übertragen haben, bevor Sie die Konfigurationen anwenden. `.ebextensions` Wenn Sie während dieser Konfiguration auf VPC- oder Subnetzkonfigurationsfehler stoßen, warten Sie einige Minuten und versuchen Sie es erneut.
> DNS-Übertragungszeit. Nachdem Sie die `IpAddressType` Option auf Dualstack gesetzt haben, warten Sie einige Minuten, bis die Änderungen wirksam werden, bevor Sie den Test durchführen. Es kann 1—2 Minuten dauern, bis die DNS-Propagierung für Route 53 abgeschlossen ist. Während dieser Zeit können beim Testen der Kommunikation zwischen einem Client und Ihrer Anwendung Probleme auftreten, wenn Sie Anfragen mithilfe des IPv6 Protokolls initiieren.

Example.ebextensions/options.config für die Dual-Stack-Konfiguration des Load Balancers (Kurzsyntax)

```
option_settings:
  aws:elbv2:loadbalancer:
```

```
    IpAddressType: dualstack
```

Example.ebextensions/options.config für die Dual-Stack-Konfiguration des Load Balancers (Standardsyntax)

```
option_settings:
  - namespace: aws:elbv2:loadbalancer
    option_name: IpAddressType
    value: dualstack
```

Verwenden AWS des SDK

Sie können Dual-Stack mit dem AWS SDKskonfigurieren. Ähnlich wie bei den im vorherigen Abschnitt erwähnten `create-environment` AWS CLI Befehlen `update-environment` und können Sie die UpdateEnvironmentAPI-Aktionen CreateEnvironmentund verwenden. Verwenden Sie den `OptionSettings` Anforderungsparameter, um die Optionen des aws:elbv2:loadbalancer Namespaces anzugeben.

> **Note**
>
> Diese Konfiguration hängt vom Zeitpunkt der Datenweitergabe an mehreren Punkten ab. Beachten Sie die folgenden zeitlichen Anforderungen, wenn Sie Ihren Load Balancer konfigurieren und die Änderungen testen.
> VPC-Konfigurationszeit. Nachdem Sie die unter beschriebene VPC-Konfiguration abgeschlossen haben, warten Sie einige MinutenVoraussetzungen für Amazon VPC, bis sich die Änderungen übertragen haben, bevor Sie Ihre Programme ausführen, um den Load Balancer für Dual-Stack zu konfigurieren. Wenn beim Dual-Stack-Setup VPC- oder Subnetzkonfigurationsfehler auftreten, warten Sie einige Minuten und versuchen Sie erneut, die Programme für die Dual-Stack-Konfiguration auszuführen.
> DNS-Übertragungszeit. Nachdem Sie die `IpAddressType` Option auf Dualstack gesetzt haben, warten Sie einige Minuten, bis die Änderungen wirksam werden, bevor Sie den Test durchführen. Es kann 1—2 Minuten dauern, bis die DNS-Propagierung für Route 53 abgeschlossen ist. Während dieser Zeit können beim Testen der Kommunikation zwischen einem Client und Ihrer Anwendung Probleme auftreten, wenn Sie Anfragen mithilfe des IPv6 Protokolls initiieren.

Fehlerbehebung

> **Testen Sie Amazon Q Developer CLI für KI-gestützte Problembehebung**
>
> Amazon Q Developer CLI kann Ihnen helfen, Umgebungsprobleme schnell zu beheben. Die Q CLI bietet Lösungen, indem sie den Umgebungsstatus überprüft, Ereignisse überprüft, Protokolle analysiert und klärende Fragen stellt. Weitere Informationen und detaillierte Anleitungen finden Sie in den Blogs unter Troubleshooting Elastic Beanstalk Environments with Amazon Q Developer CLI. AWS

Dieser Abschnitt enthält Anleitungen zur Behebung von Problemen mit Ihrer Load Balancer-Dual-Stack-Konfiguration.

Ereignis: In der VPC sind *vpc_id* keine IPv6 CIDR-Blöcke konfiguriert. IPv6 CIDR-Blöcke sind für den Dual-Stack-Load Balancer erforderlich. Bitte ordnen Sie Ihrer VPC einen IPv6 CIDR-Block zu, bevor Sie den Dual-Stack-Modus verwenden.

Ihrer VPC und allen Subnetzen müssen IPv6 CIDR-Blöcke zugeordnet sein. Dies ist eine der VPC-Voraussetzungen, die Sie erfüllen müssen, bevor Sie Ihren Load Balancer für die Dual-Stack-Unterstützung konfigurieren können. Weitere Informationen zur Ausführung dieser Aufgabe finden Sie weiter Voraussetzungen für Amazon VPC oben in diesem Thema.

Ereignis: In einem oder mehreren Subnetzen für VPC sind *vpc_id* keine IPv6 CIDR-Blöcke konfiguriert. IPv6 CIDR-Blöcke sind für die Subnetze erforderlich, die mit dem Dualstack Load Balancer verwendet werden. Bitte ordnen Sie IPv6 CIDR-Blöcke allen erforderlichen Subnetzen zu, bevor Sie den Dual-Stack-Modus verwenden.

Allen Subnetzen für Ihre VPC müssen IPv6 CIDR-Blöcke zugeordnet sein. Dies ist eine der VPC-Voraussetzungen, die Sie erfüllen müssen, bevor Sie Ihren Load Balancer für die Dual-Stack-Unterstützung konfigurieren können. Weitere Informationen zur Ausführung dieser Aufgabe finden Sie weiter Voraussetzungen für Amazon VPC oben in diesem Thema.

Fehler: Die *IpAddressType* Option kann nur auf Elastic Beanstalk Beanstalk-Umgebungen angewendet werden, die mit einem Application Load Balancer oder Network Load Balancer konfiguriert sind.

Diese Meldung weist darauf hin, dass es sich bei Ihrer Elastic Beanstalk Beanstalk-Umgebung möglicherweise um eine Single-Instance-Umgebung handelt oder dass sie einen Classic Load

Balancer verwendet. Nur Umgebungen, die mit einem Application Load Balancer oder Network Load Balancer konfiguriert sind, können konfiguriert werden. `IpAddressType`

Konfigurieren von Zugriffsprotokollen

Mithilfe der Konfigurationsdateien können Sie den Load Balancer in der Umgebung für den Upload der Zugriffsprotokolle in einen Amazon S3-Bucket konfigurieren. Anweisungen finden Sie in den folgenden Beispielkonfigurationsdateien unter GitHub :

- `loadbalancer-accesslogs-existingbucket.config` – Konfigurieren Sie den Load Balancer für den Upload der Zugriffsprotokolle in einen vorhandenen Amazon S3-Bucket.
- `loadbalancer-accesslogs-newbucket.config` – Konfigurieren Sie den Load Balancer für den Upload der Zugriffsprotokolle in einen neuen Bucket.

Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung

Elastic Beanstalk bietet eine gekoppelte Datenbankintegration mit Amazon Relational Database Service (Amazon RDS). Sie können Elastic Beanstalk verwenden, um eine MySQL-, PostgreSQL-, Oracle- oder SQL Server-Datenbank zu einer bestehenden oder einer neuen Umgebung hinzuzufügen, wenn Sie diese erstellen. Wenn Sie eine Datenbank-Instance hinzufügen, die an eine Umgebung gekoppelt ist, stellt Elastic Beanstalk die Verbindungsinformationen für Ihre Anwendung bereit. Dadurch werden die Umgebungseigenschaften für den Datenbank-Hostnamen, den Port, Benutzernamen und das Passwort sowie den Datenbanknamen festgelegt.

Vorteile der Verwendung einer gekoppelten Datenbank

Wenn Sie zuvor keine Datenbank-Instance mit Ihrer Anwendung verwendet haben, empfehlen wir Ihnen, zunächst den in diesem Thema beschriebenen Prozess zu verwenden, um zu einer Testumgebung mithilfe eines Elastic-Beanstalk-Services eine Datenbank hinzuzufügen. Auf diese Weise können Sie überprüfen, ob Ihre Anwendung die Umgebungseigenschaften lesen, eine Verbindungszeichenfolge erstellen und eine Verbindung zu einer Datenbank-Instance herstellen kann, ohne die zusätzliche Konfigurationsarbeit, die für eine Datenbank außerhalb von Elastic Beanstalk erforderlich ist.

Überlegungen beim Produktionsstart

Nachdem Sie überprüft haben, dass Ihre Anwendung ordnungsgemäß mit der Datenbank funktioniert, erwägen Sie möglicherweise, sich in eine Produktionsumgebung zu bewegen. An dieser Stelle haben

Sie die Möglichkeit, die Datenbank von Ihrer Elastic-Beanstalk-Umgebung zu entkoppeln, um zu einer Konfiguration zu gelangen, die mehr Flexibilität bietet. Die entkoppelte Datenbank kann als externe Amazon-RDS-Datenbank-Instance betriebsbereit bleiben. Der Zustand der Umgebung wird durch die Entkopplung der Datenbank nicht beeinflusst. Wenn Sie die Umgebung beenden müssen, können Sie dies tun und auch die Option auswählen, um die Datenbank außerhalb von Elastic Beanstalk verfügbar und betriebsbereit zu halten.

Vorteile der Umstellung auf eine entkoppelte Datenbank

Die Verwendung einer externen Datenbank hat mehrere Vorteile. Sie können von mehreren Umgebungen aus eine Verbindung mit der externen Datenbank herstellen, Datenbanktypen verwenden, die von integrierten Datenbanken nicht unterstützt werden, und Bereitstellungen durchführen blue/green . Alternativ zur Verwendung einer entkoppelten Datenbank, die Elastic Beanstalk erstellt hat, können Sie auch eine Datenbank-Instance außerhalb Ihrer Elastic-Beanstalk-Umgebung erstellen. Beide Optionen führen zu einer Datenbank-Instance, die sich außerhalb Ihrer Elastic-Beanstalk-Umgebung befindet und eine zusätzliche Konfiguration von Sicherheitsgruppen und Verbindungszeichenfolgen erfordert. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon RDS.

Sections

- Datenbank-Lebenszyklus
- Hinzufügen einer Amazon-RDS-DB-Instance zur Umgebung mithilfe der Konsole
- Verbinden mit der Datenbank
- Konfigurieren einer integrierten RDS-DB-Instance mithilfe der Konsole
- Konfigurieren einer integrierten RDS-DB-Instance mithilfe von Konfigurationsdateien
- Entkoppeln einer RDS-DB-Instance über die Konsole
- Entkoppeln einer RDS-DB-Instance mithilfe von Konfigurationsdateien

Datenbank-Lebenszyklus

Sie können auswählen, was mit der Datenbank geschehen soll, nachdem Sie sie von Ihrer Elastic-Beanstalk-Umgebung entkoppelt haben. Die Optionen, aus denen Sie wählen können, werden zusammenfassend als Löschrichtlinien bezeichnet. Die folgenden Löschrichtlinien gelten für eine Datenbank, nachdem Sie sie von einer Elastic-Beanstalk-Umgebung entkoppelt oder die Elastic-Beanstalk-Umgebung beendet haben.

- Snapshot – Bevor Elastic Beanstalk die Datenbank beendet, speichert es einen Snapshot davon. Sie können eine Datenbank aus einem Snapshot wiederherstellen, wenn Sie eine DB-Instance zu einer Elastic-Beanstalk-Umgebung hinzufügen oder wenn Sie eine eigenständige Datenbank erstellen. Weitere Informationen zum Erstellen einer neuen eigenständigen DB-Instance aus einem Snapshot finden Sie unter Wiederherstellen aus einem DB-Snapshot im Amazon-RDS-Benutzerhandbuch. Für das Speichern von Datenbank-Snapshots fallen möglicherweise Kosten an. Weitere Informationen finden Sie im Abschnitt Backup-Speicher unter Amazon RDS – Preise.
- Löschen – Elastic Beanstalk beendet die Datenbank. Nach dem Beenden ist die Datenbank-Instance für keinen Vorgang mehr verfügbar.
- Beibehalten – Die Datenbank-Instance wird nicht beendet. Sie bleibt verfügbar und betriebsbereit, obwohl sie von Elastic Beanstalk entkoppelt ist. Sie können dann eine oder mehrere Umgebungen für die Verbindung mit der Datenbank als externe Amazon-RDS-Datenbank-Instance konfigurieren. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon RDS.

Hinzufügen einer Amazon-RDS-DB-Instance zur Umgebung mithilfe der Konsole

Sie können der Umgebung mit der Elastic Beanstalk-Konsole eine DB-Instance hinzufügen.

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Sie können die folgenden Optionen konfigurieren:

- Snapshot – Wählen Sie einen Datenbank-Snapshot aus. Elastic Beanstalk stellt den Snapshot wieder her und fügt ihn der Umgebung hinzu. Der Standardwert ist None (Kein). Wenn der Wert

Keiner ist, können Sie eine neue Datenbank mit den anderen Einstellungen auf dieser Seite konfigurieren.

- Engine – Wählen Sie eine Datenbank-Engine aus.
- Engine version (Engine-Version) – Wählen Sie eine bestimmte Version der Datenbank-Engine aus.
- Instance class (Instance-Klasse) – Wählen Sie eine DB-Instance-Klasse aus. Weitere Informationen zu DB-Instance-Klassen finden Sie unter https://aws.amazon.com/rds/.
- Storage (Speicher) – Wählen Sie die Speicherkapazität für Ihre Datenbank aus. Sie können zugewiesenen Speicher zu einem späteren Zeitpunkt erhöhen, jedoch nicht verringern. Weitere Informationen zur Speicherzuweisung finden Sie unter Funktionen.
- Benutzername: – Geben Sie einen beliebigen Benutzernamen mit einer Kombination aus Zahlen und Buchstaben ein.
- Password (Passwort) – Geben Sie ein beliebiges Passwort aus 8–16 druckbaren ASCII-Zeichen (ohne `/`, `\` und `@`) ein.
- Availability (Verfügbarkeit) – Wählen Sie High (Multi-AZ) (Hoch (Multi-AZ) aus, um ein Warm-Backup in einer zweiten Availability Zone für hohe Verfügbarkeit durchzuführen.
- Richtlinie zum Löschen von Datenbanken – Die Richtlinie zum Löschen bestimmt, was mit der Datenbank passiert, nachdem sie von Ihrer Umgebung entkoppelt wurde. Werte können einer der folgenden sein: `Create Snapshot`, `Retain` oder `Delete`. Diese Werte sind unter Datenbank-Lebenszyklus in demselben Thema beschrieben.

> **Note**
>
> Elastic Beanstalk erstellt einen Master-Benutzer für die Datenbank unter Verwendung des Benutzernamens und des Passworts, die Sie festgelegt haben. Weitere Informationen zum Master-Benutzer und dessen Berechtigungen finden Sie unter Berechtigungen von Hauptbenutzerkonten.

Das Hinzufügen einer DB-Instance dauert etwa 10 Minuten. Wenn das Update abgeschlossen ist, wird die neue Datenbank mit Ihrer Umgebung gekoppelt. Der Hostname und andere Verbindungsinformationen für die DB-Instance stehen Ihrer Anwendung über die folgenden Umgebungseigenschaften zur Verfügung.

Name der Eigenschaft	Beschreibung	Eigenschaftenwert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.

Verbinden mit der Datenbank

Verwenden Sie die Verbindungsinformationen für die Verbindung mit Ihrer Datenbank von innerhalb Ihrer Anwendung über Umgebungsvariablen. Weitere Informationen zur Verwendung von Amazon RDS mit Ihren Anwendungen finden Sie in den folgenden Themen.

- Java SE – Verbinden mit einer Datenbank (Java SE-Plattformen)
- Java mit Tomcat – Verbinden mit einer Datenbank (Tomcat-Plattformen)

- Node.js – Herstellen einer Verbindung zu einer Datenbank
- .NET – Verbinden mit einer Datenbank
- PHP – Mit einem PDO oder My verbinden Sie sich mit einer Datenbank SQLi
- Python – Herstellen einer Verbindung zu einer Datenbank
- Ruby – Herstellen einer Verbindung zu einer Datenbank

Konfigurieren einer integrierten RDS-DB-Instance mithilfe der Konsole

Sie können Konfigurationseinstellungen für die Datenbank-Instance im Abschnitt Database (Datenbank) auf der Seite Configuration (Konfiguration) der Umgebung in der Elastic-Beanstalk-Konsole anzeigen und ändern.

So konfigurieren Sie die DB-Instance Ihrer Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten) aus.

Sie können die Richtlinieneinstellungen für Instance-Klasse, Speicher, Passwort, Verfügbarkeit und Richtlinie zum Löschen von Datenbanken nach der Datenbankerstellung ändern. Wenn Sie die Instance-Klasse ändern, stellt Elastic Beanstalk die DB-Instance erneut bereit.

Wenn Sie Elastic Beanstalk nicht mehr benötigen, um die Datenbank mit der Umgebung zu verknüpfen, können Sie sie entkoppeln, indem Sie Datenbank entkoppeln auswählen. Es ist wichtig, die Optionen und Überlegungen zu diesem Vorgang zu verstehen. Weitere Informationen finden Sie unter the section called “Entkoppeln einer RDS-DB-Instance über die Konsole”.

> **Warnung**
>
> Ändern Sie keine Einstellungen der gekoppelten Datenbank-Instance außerhalb der von Elastic Beanstalk bereitgestellten Funktionen (z. B. in der Amazon-RDS-Konsole). Wenn Sie dies tun, ist Ihre Amazon RDS-DB-Konfiguration möglicherweise nicht synchron mit

der Definition Ihrer Umgebung. Wenn Sie Ihre Umgebung aktualisieren oder neu starten, überschreiben die Einstellungen in der Umgebung alle Einstellungen, die Sie außerhalb von Elastic Beanstalk vornehmen.
Wenn Sie Einstellungen ändern müssen, die Elastic Beanstalk nicht direkt unterstützt, verwenden Sie die Elastic Beanstalk-Konfigurationsdateien.

Konfigurieren einer integrierten RDS-DB-Instance mithilfe von Konfigurationsdateien

Sie können die Datenbank-Instance Ihrer Umgebung mithilfe von Konfigurationsdateien konfigurieren. Verwenden Sie die Optionen im Namespace `aws:rds:dbinstance`. Das folgende Beispiel ändert die Größe des zugewiesenen Datenbankspeichers auf 100 GB.

Example.ebeextensions/ .config db-instance-options

```
option_settings:
  aws:rds:dbinstance:
    DBAllocatedStorage: 100
```

Wenn Sie DB-Instance-Eigenschaften konfigurieren, die Elastic Beanstalk nicht unterstützt, können Sie dennoch eine Konfigurationsdatei verwenden und Ihre Einstellungen mithilfe des `resources`-Schlüssels angeben. Das folgende Beispiel legt Werte für die Amazon RDS-Eigenschaften `StorageType` und `Iops` fest.

Example.ebextensions/ db-instance-properties .config

```
Resources:
  AWSEBRDSDatabase:
    Type: AWS::RDS::DBInstance
    Properties:
      StorageType:io1
      Iops: 1000
```

Entkoppeln einer RDS-DB-Instance über die Konsole

Sie können Ihre Datenbank von einer Elastic-Beanstalk-Umgebung entkoppeln, ohne den Zustand der Umgebung zu beeinträchtigen. Berücksichtigen Sie die folgenden Anforderungen, bevor Sie die Datenbank entkoppeln:

- Was sollte mit der Datenbank passieren, nachdem sie entkoppelt wurde?

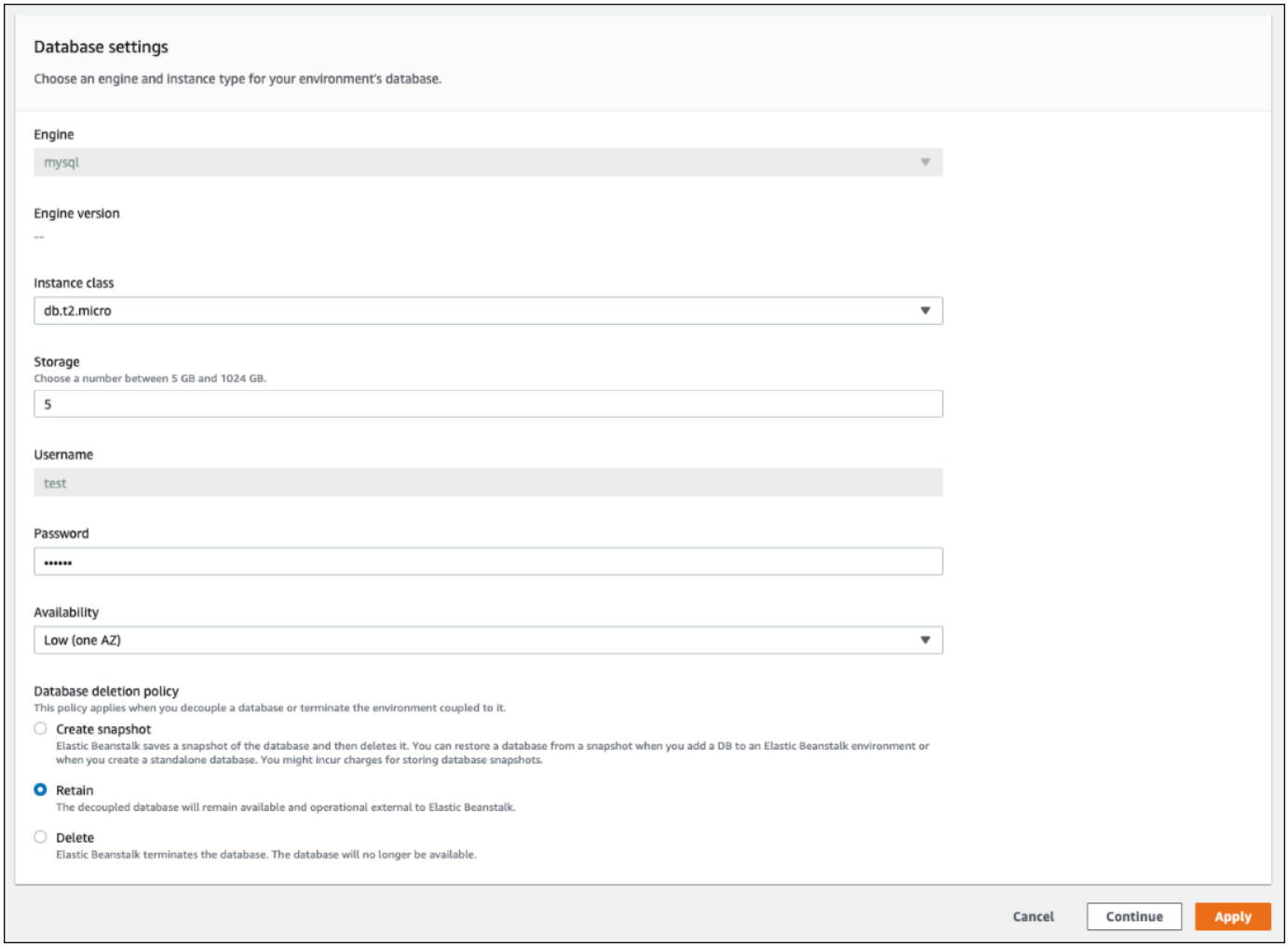
  Sie können einen Snapshot der Datenbank erstellen und dann beenden, die Datenbank als eigenständige Datenbank außerhalb von Elastic Beanstalk betriebsbereit halten oder die Datenbank dauerhaft löschen. Die Einstellung Richtlinie zum Löschen von Datenbanken bestimmt dieses Ergebnis. Eine detaillierte Beschreibung der Löschrichtlinien finden Sie unter Datenbank-Lebenszyklus in demselben Thema.

- Müssen Änderungen an den Datenbankkonfigurationseinstellungen vorgenommen werden, bevor sie entkoppelt wird?

  Wenn Sie Konfigurationsänderungen an der Datenbank vornehmen müssen, sollten Sie diese vor dem Entkoppeln der Datenbank anwenden. Dies schließt Änderungen an der Richtlinie zum Löschen von Datenbanken ein. Alle ausstehenden Änderungen, die gleichzeitig mit der Datenbankeinstellung Entkoppeln übermittelt werden, werden ignoriert, während nur die Entkopplungseinstellung angewendet wird.

So entkoppeln Sie eine DB-Instance aus einer Umgebung

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten) aus.
5. Überprüfen Sie alle Konfigurationswerte im Abschnitt Datenbankeinstellungen, insbesondere die Richtlinie zum Löschen von Datenbanken, die bestimmt, was mit der Datenbank nach der Entkopplung passiert.



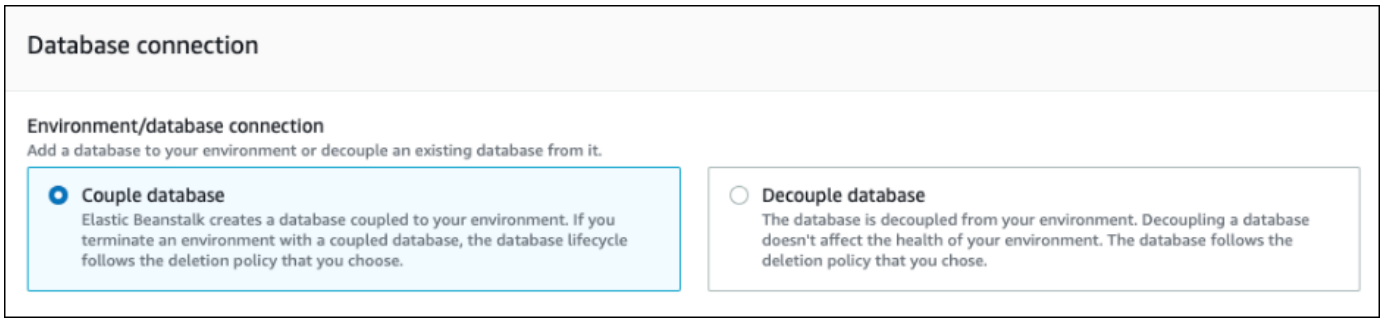
Wenn alle anderen Konfigurationseinstellungen korrekt sind, fahren Sie mit Schritt 6 fort, um die Datenbank zu entkoppeln.

> **Warning**
>
> Es ist wichtig, die Einstellung der Richtlinie zum Löschen von Datenbanken getrennt von Datenbank entkoppeln anzuwenden. Wenn Sie Anwenden auswählen, um sowohl Datenbank entkoppeln als auch eine neu ausgewählte Richtlinie zum Löschen von Datenbanken zu speichern, wird die von Ihnen ausgewählte neue Löschrichtlinie ignoriert. Elastic Beanstalk entkoppelt die Datenbank nach der vorgesetzten Löschrichtlinie. Wenn die zuvor festgelegte Löschrichtlinie `Delete` oder `Create Snapshot`, ist, riskieren Sie den Verlust der Datenbank, anstatt die beabsichtigte ausstehende Richtlinie zu befolgen.

Wenn eine der Konfigurationseinstellungen Aktualisierungen erfordert, gehen Sie wie folgt vor:

1. Nehmen Sie die erforderlichen Änderungen im Bereich Datenbankeinstellungen vor.
2. Wählen Sie Anwenden aus. Es dauert einige Minuten, um die Konfigurationsänderungen für Ihre Datenbank zu speichern.
3. Gehen Sie zurück zu Schritt 3 und wählen Sie Konfiguration aus dem Navigationsbereich.

6. Wechseln Sie zum Abschnitt Datenbankverbindung des Bereichs.



7. Wählen Sie Datenbank entkoppeln aus.
8. Wählen Sie Anwenden aus, um den Vorgang zum Entkoppeln der Datenbank zu starten.

Die Löschrichtlinieneinstellung bestimmt das Ergebnis der Datenbank und die Dauer, die zum Entkoppeln der Datenbank erforderlich ist.

- Wenn die Löschrichtlinie auf `Delete` gesetzt ist, wird die Datenbank gelöscht. Der Vorgang kann je nach Größe der Datenbank ungefähr 10-20 Minuten dauern.
- Wenn die Löschrichtlinie auf `Snapshot` gesetzt ist, wird ein Snapshot der Datenbank erstellt. Dann wird die Datenbank gelöscht. Die für diesen Prozess erforderliche Zeit variiert je nach Größe der Datenbank.
- Wenn die Löschrichtlinie auf `Retain` gesetzt ist, bleibt die Datenbank außerhalb der Elastic-Beanstalk-Umgebung betriebsbereit. Normalerweise dauert es weniger als fünf Minuten, um eine Datenbank zu entkoppeln.

Wenn Sie sich entschieden haben, die Datenbank außerhalb Ihrer Elastic-Beanstalk-Umgebung zu belassen, müssen Sie zusätzliche Schritte unternehmen, um sie zu konfigurieren. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon RDS. Wenn Sie

vorhaben, die Datenbank zu verwenden, die Sie für eine Produktionsumgebung entkoppeln, überprüfen Sie, ob der Speichertyp, den die Datenbank verwendet, für Ihren Workload geeignet ist. Weitere Informationen finden Sie unter DB-Instance-Speicher und Ändern einer DB-Instance im Amazon-RDS-Benutzerhandbuch.

Entkoppeln einer RDS-DB-Instance mithilfe von Konfigurationsdateien

Sie können Ihre DB-Instance von einer Elastic-Beanstalk-Umgebung entkoppeln, ohne den Zustand der Umgebung zu beeinträchtigen. Die Datenbank-Instance folgt der Richtlinie zum Löschen der Datenbank, die beim Entkoppeln der Datenbank angewendet wurde.

Beide Optionen, die zum Entkoppeln der Datenbank erforderlich sind, befinden sich im Namespace the section called “aws:rds:dbinstance”. Diese sind:

- Die Option `DBDeletionPolicy` legt die Löschrichtlinie fest. Werte können einer der folgenden sein: `Snapshot`, `Delete` oder `Retain`. Diese Werte sind unter Datenbank-Lebenszyklus in demselben Thema beschrieben.
- Die Option `HasCoupledDatabase` bestimmt, ob Ihre Umgebung über eine gekoppelte Datenbank verfügt.
  - Wenn auf `true` umgeschaltet, erstellt Elastic Beanstalk eine neue DB-Instance, die mit Ihrer Umgebung gekoppelt ist.
  - Wenn auf `false` umgeschaltet, beginnt Elastic Beanstalk mit der Entkopplung der DB-Instance von Ihrer Umgebung.

Wenn Sie Ihre Datenbankkonfiguration ändern möchten, bevor Sie sie entkoppeln, wenden Sie alle Konfigurationsänderungen zuerst in einem separaten Vorgang an. Dazu gehört auch das Ändern der `DBDeletionPolicy`-Konfiguration. Nachdem Ihre Änderungen übernommen wurden, führen Sie einen separaten Befehl aus, um die Entkopplungsoption festzulegen. Wenn Sie gleichzeitig andere Konfigurationseinstellungen und die Entkopplungseinstellung übermitteln, werden die anderen Konfigurationsoptionseinstellungen ignoriert, während die Entkopplungseinstellung angewendet wird.

> **Warning**
>
> Es ist wichtig, dass Sie die Befehle zum Anwenden der `DBDeletionPolicy`- und `HasCoupledDatabase`-Einstellungen als zwei separate Vorgänge ausführen. Wenn die aktive Löschrichtlinie bereits auf `Delete` oder `Snapshot` gesetzt ist, riskieren Sie den

Verlust der Datenbank. Die Datenbank folgt der derzeit aktiven Löschrichtlinie und nicht der von Ihnen beabsichtigten Richtlinie zum ausstehenden Löschen.

So entkoppeln Sie eine DB-Instance aus einer Umgebung

Folgen Sie diesen Schritten, um die Datenbank von Ihrer Elastic-Beanstalk-Umgebung zu entkoppeln. Sie können die EB-CLI oder die verwenden AWS CLI , um die Schritte abzuschließen. Weitere Informationen finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien.

1. Wenn Sie die Löschrichtlinie ändern möchten, richten Sie eine Konfigurationsdatei im folgenden Format ein. In diesem Beispiel wird die Löschrichtlinie beibehalten.

   Example

   ```
   option_settings:
     aws:rds:dbinstance:
       DBDeletionPolicy: Retain
   ```

2. Führen Sie den Befehl mit Ihrem bevorzugten Tool aus, um das Konfigurationsupdate abzuschließen.
3. Richten Sie eine Konfigurationsdatei ein, um `HasCoupledDatabase` auf `false` zu setzen.

   Example

   ```
   option_settings:
     aws:rds:dbinstance:
       HasCoupledDatabase: false
   ```

4. Führen Sie den Befehl mit Ihrem bevorzugten Tool aus, um das Konfigurationsupdate abzuschließen.

Die Löschrichtlinieneinstellung bestimmt das Ergebnis der Datenbank und die Dauer, die zum Entkoppeln der Datenbank erforderlich ist.

- Wenn die Löschrichtlinie auf `Delete` gesetzt ist, wird die Datenbank gelöscht. Der Vorgang kann je nach Größe der Datenbank ungefähr 10-20 Minuten dauern.

- Wenn die Löschrichtlinie auf `Snapshot` gesetzt ist, wird ein Snapshot der Datenbank erstellt. Dann wird die Datenbank gelöscht. Die für diesen Prozess erforderliche Zeit variiert je nach Größe der Datenbank.
- Wenn die Löschrichtlinie auf `Retain` gesetzt ist, bleibt die Datenbank außerhalb der Elastic-Beanstalk-Umgebung betriebsbereit. Normalerweise dauert es weniger als fünf Minuten, um eine Datenbank zu entkoppeln.

Wenn Sie sich entschieden haben, die Datenbank außerhalb Ihrer Elastic-Beanstalk-Umgebung zu belassen, müssen Sie zusätzliche Schritte unternehmen, um sie zu konfigurieren. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon RDS. Wenn Sie vorhaben, die Datenbank zu verwenden, die Sie für eine Produktionsumgebung entkoppeln, überprüfen Sie, ob der Speichertyp, den die Datenbank verwendet, für Ihren Workload geeignet ist. Weitere Informationen finden Sie unter DB-Instance-Speicher und Ändern einer DB-Instance im Amazon-RDS-Benutzerhandbuch.

Sicherheit Ihrer AWS Elastic Beanstalk Umgebung

Elastic Beanstalk bietet mehrere Optionen, mit denen Sie den Servicezugriff (Sicherheit) Ihrer Umgebung und der darin enthaltenen EC2 Amazon-Instances steuern können. In diesem Thema wird die Konfiguration dieser Optionen beschrieben

Sections

- Konfigurieren der Umgebungssicherheit
- Namespaces für die Konfiguration der Umgebungssicherheit

Konfigurieren der Umgebungssicherheit

Sie können die Sicherheitskonfiguration Ihrer Elastic Beanstalk-Umgebung in der Elastic Beanstalk-Konsole ändern.

So konfigurieren Sie den Service-Zugriff (Sicherheit) Ihrer Umgebung in der Elastic-Beanstalk-Konsole

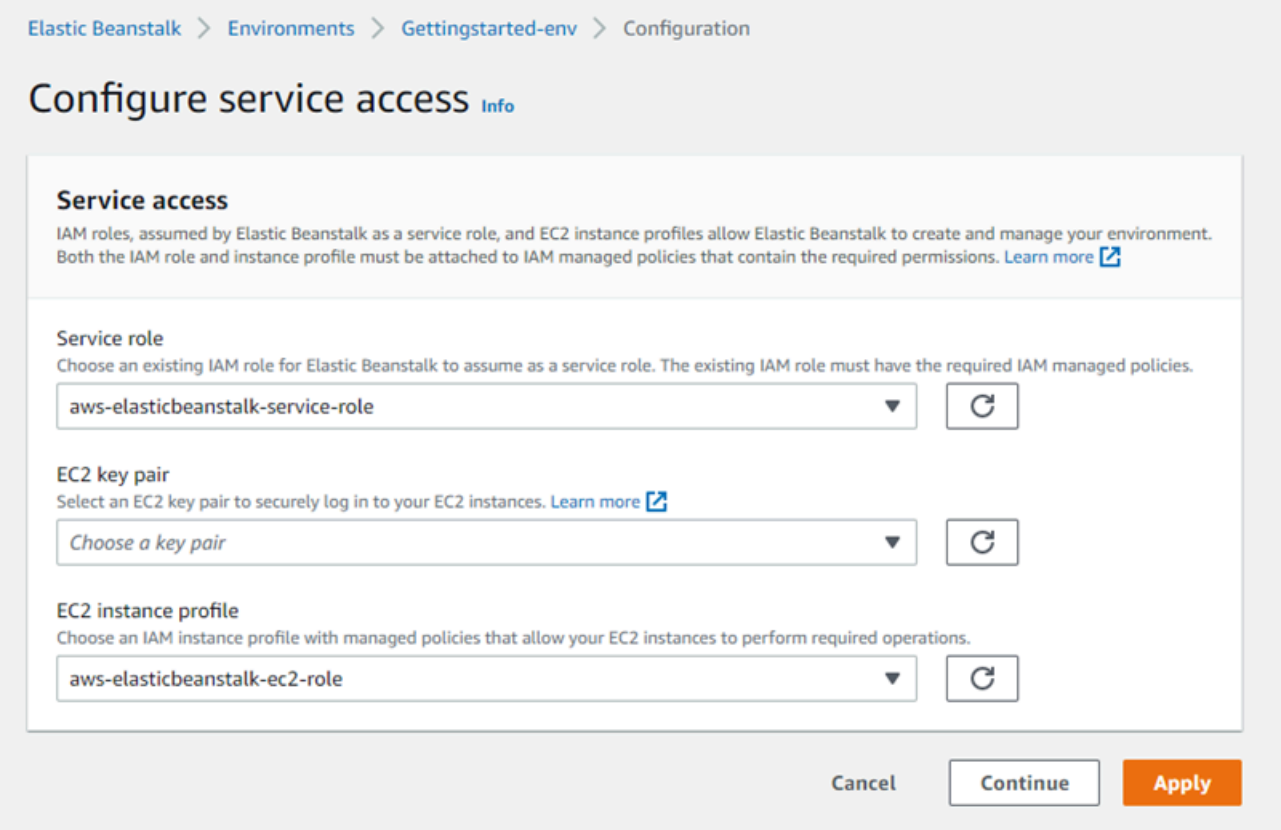
1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region

2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Service access (Service-Zugriff) die Option Edit (Bearbeiten) aus.

Die folgenden Einstellungen sind verfügbar.

Einstellungen

- Service – -Rolle
- EC2 key pair
- IAM-Instance-Profil



Service – -Rolle

Wählen Sie eine Service role (Servicerolle) aus, die Sie Ihrer Elastic Beanstalk-Umgebung zuordnen möchten. Elastic Beanstalk übernimmt die Service-Rolle, wenn es in Ihrem Namen auf andere AWS Services zugreift. Details hierzu finden Sie unter Rollen von Elastic Beanstalk Service verwalten.

EC2 key pair

Sie können sich mit einem Amazon-Schlüsselpaar sicher bei den Amazon Elastic Compute Cloud (Amazon EC2) -Instances anmelden, die für Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt wurden. EC2 Anweisungen zum Erstellen eines key pair finden Sie unter Erstellen eines Schlüsselpaars mit Amazon EC2 im EC2 Amazon-Benutzerhandbuch.

> **Note**
>
> Wenn Sie ein key pair erstellen, EC2 speichert Amazon eine Kopie Ihres öffentlichen Schlüssels. Wenn Sie es nicht mehr verwenden müssen, um eine Verbindung zu Umgebungsinstanzen herzustellen, können Sie es aus Amazon löschen EC2. Einzelheiten finden Sie unter Löschen Ihres Schlüsselpaars im EC2 Amazon-Benutzerhandbuch.

Wählen Sie ein EC2 key pair aus dem Drop-down-Menü aus, um es den Instanzen Ihrer Umgebung zuzuweisen. Beim Zuweisen eines Schlüsselpaars wird der öffentliche Schlüssel (zum Authentifizieren des privaten Schlüssels) auf der Instance gespeichert. Die Speicherung des lokalen Schlüssels hingegen erfolgt lokal. Der private Schlüssel wird niemals auf gespeichert AWS.

Weitere Informationen zum Herstellen einer Verbindung zu EC2 Amazon-Instances finden Sie unter Connect to Your Instance und Connecting to Linux/UNIX Instances from Windows using PuTTY im EC2 Amazon-Benutzerhandbuch.

IAM-Instance-Profil

Ein EC2 Instance-Profil ist eine IAM-Rolle, die auf Instances angewendet wird, die in Ihrer Elastic Beanstalk Beanstalk-Umgebung gestartet wurden. EC2 Amazon-Instances übernehmen die Rolle des Instance-Profils, um Anfragen zu signieren AWS und darauf zuzugreifen APIs, z. B. um Protokolle auf Amazon S3 hochzuladen.

Wenn Sie zum ersten Mal eine Umgebung mit der Elastic Beanstalk-Konsole erstellen, werden Sie von Elastic Beanstalk zum Erstellen eines Instance-Profils mit Standardberechtigungen aufgefordert.

Sie können diesem Profil Berechtigungen hinzufügen, um Ihren Instances Zugriff auf andere AWS Dienste zu gewähren. Details hierzu finden Sie unter Elastic Beanstalk Instance-Profile verwalten.

> **Note**
>
> Zuvor hat Elastic Beanstalk ein EC2 Standard-Instanzprofil erstellt, `aws-elasticbeanstalk-ec2-role` das beim ersten Erstellen einer Umgebung durch ein AWS Konto benannt wurde. Dieses Instance-Profil enthielt verwaltete Standardrichtlinien. Wenn Ihr Konto bereits über dieses Instance-Profil verfügt, können Sie es weiterhin Ihren Umgebungen zuweisen.
> Aktuelle AWS Sicherheitsrichtlinien erlauben es in diesem Fall jedoch nicht, dass ein AWS Service automatisch Rollen mit Vertrauensrichtlinien für andere AWS Dienste erstellt. EC2 Aufgrund dieser Sicherheitsrichtlinien erstellt Elastic Beanstalk kein `aws-elasticbeanstalk-ec2-role`-Standard-Instance-Profil mehr.

> **Note**
>
> Es gibt noch einen weiteren Aspekt der EC2 Instanzsicherheit, der Firewallregeln für EC2 Instanzen festlegt. Dies wird von EC2 Sicherheitsgruppen gesteuert. Weitere Informationen finden Sie unter Die EC2 Amazon-Instances für Ihre Elastic Beanstalk Beanstalk-Umgebung.

Namespaces für die Konfiguration der Umgebungssicherheit

Elastic Beanstalk bietet Konfigurationsoptionen in den folgenden Namespaces, damit Sie die Sicherheit Ihrer Umgebung anpassen können:

- `aws:elasticbeanstalk:environment` – Konfigurieren der Servicerolle der Umgebung mit der Option `ServiceRole`.
- `aws:autoscaling:launchconfiguration`— Konfigurieren Sie mithilfe der `SecurityGroups` Optionen`EC2KeyName`,, `IamInstanceProfile` und Berechtigungen für die EC2 Amazon-Instances der Umgebung. `DisableDefaultEC2SecurityGroup`

Die EB-CLI und Elastic-Beanstalk-Konsole wenden die für die vorhergehenden Optionen empfohlenen Werte an. Diese Einstellungen müssen entfernt werden, wenn Sie

Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter Empfohlene Werte.

Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren

Sie können Tags auf Ihre AWS Elastic Beanstalk Umgebungen anwenden. Tags sind Schlüssel-Wert-Paare, die Ressourcen zugeordnet AWS sind. Weitere Informationen zum Elastic Beanstalk-Ressourcen-Tagging, zu Anwendungsfälle, Einschränkungen für Tag-Schlüssel und -Werte sowie zu unterstützten Ressourcentypen finden Sie unter Markieren von Elastic Beanstalk-Anwendungsressourcen.

Elastic Beanstalk wendet Umgebungs-Tags auf die Umgebungsressource selbst sowie auf andere AWS Ressourcen an, die Elastic Beanstalk für die Umgebung erstellt. Sie können Tags zum Verwalten von Berechtigungen auf einer bestimmten Ressourcenebene in einer Umgebung verwenden. Weitere Informationen finden Sie unter Tagging Your Amazon EC2 Resources im EC2 Amazon-Benutzerhandbuch.

Standardmäßig wendet Elastic Beanstalk ein paar Tags auf Ihre Umgebung an:

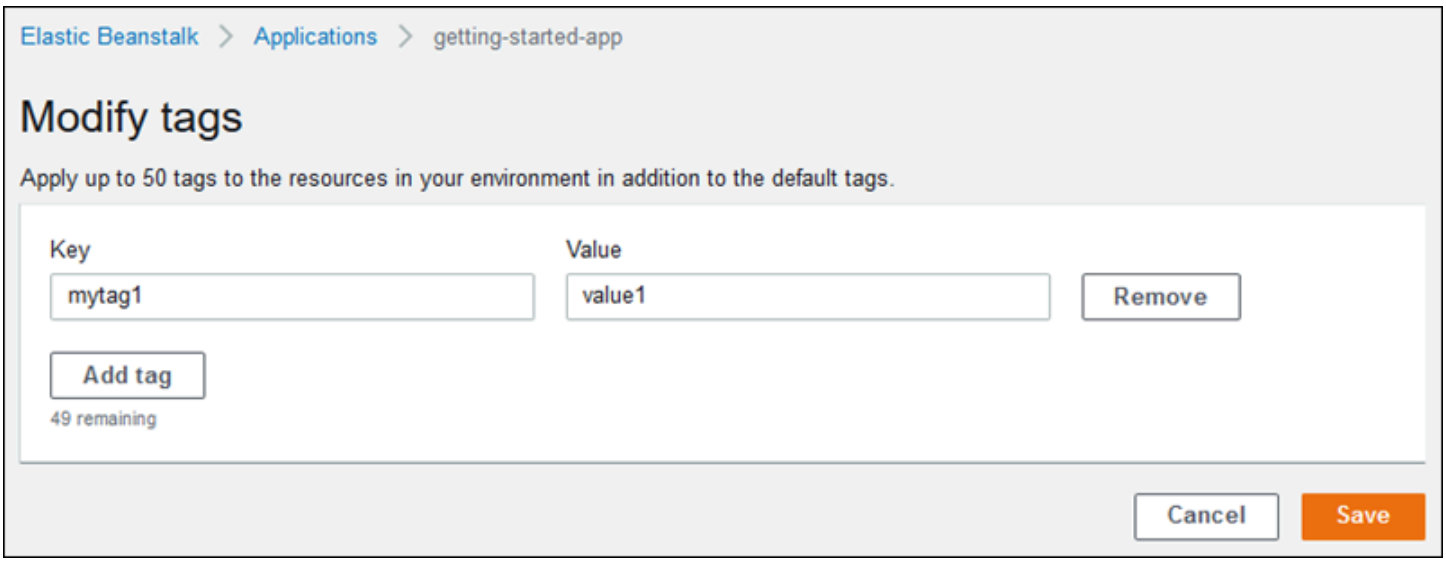
- `elasticbeanstalk:environment-name` – Der Name der Umgebung.
- `elasticbeanstalk:environment-id` – Die Umgebungs-ID.
- `Name`— Auch der Name der Umgebung. `Name`wird im EC2 Amazon-Dashboard verwendet, um Ressourcen zu identifizieren und zu sortieren.

Diese Standard-Tags können nicht bearbeitet werden.

Sie können Tags angeben, wenn Sie die Elastic Beanstalk-Umgebung erstellen. In einer vorhandenen Umgebung können Sie Tags hinzufügen oder entfernen sowie die Werte von vorhandenen Tags aktualisieren. Eine Umgebung kann einschließlich der Standard-Tags bis zu 50 Tags enthalten.

Hinzufügen von Tags beim Erstellen einer Umgebung

Wenn Sie eine Umgebung mit der Elastic Beanstalk-Konsole erstellen, können Sie auf der Konfigurationsseite Modify tags (Tags ändern) im Assistenten zum Erstellen neuer Umgebungen Tag-Schlüssel und -Werte angeben.



Wenn Sie mit der EB CLI Umgebungen erstellen, verwenden Sie die `--tags`-Option mit dem Befehl eb create, um Tags hinzuzufügen.

```
~/workspace/my-app$ eb create --tags mytag1=value1,mytag2=value2
```

Verwenden Sie bei den AWS CLI oder anderen API-basierten Clients den `--tags` Parameter für den Befehl. create-environment

```
$ aws elasticbeanstalk create-environment \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
      --application-name my-app --environment-name my-env --cname-prefix my-app --
version-label v1 --template-name my-saved-config
```

Gespeicherte Konfigurationen umfassen benutzerdefinierte Tags. Wenn Sie eine gespeicherte Konfiguration mit Tags während der Umgebungserstellung anwenden, werden diese Tags auf die neue Umgebung angewendet, solange Sie keine neuen Tags angeben. Wenn Sie Tags mit einer der vorstehenden Methoden zu einer Umgebung hinzufügen, werden sämtliche Tags, die in der gespeicherten Konfiguration definiert sind, verworfen.

Verwalten von Tags von bestehenden Umgebungen

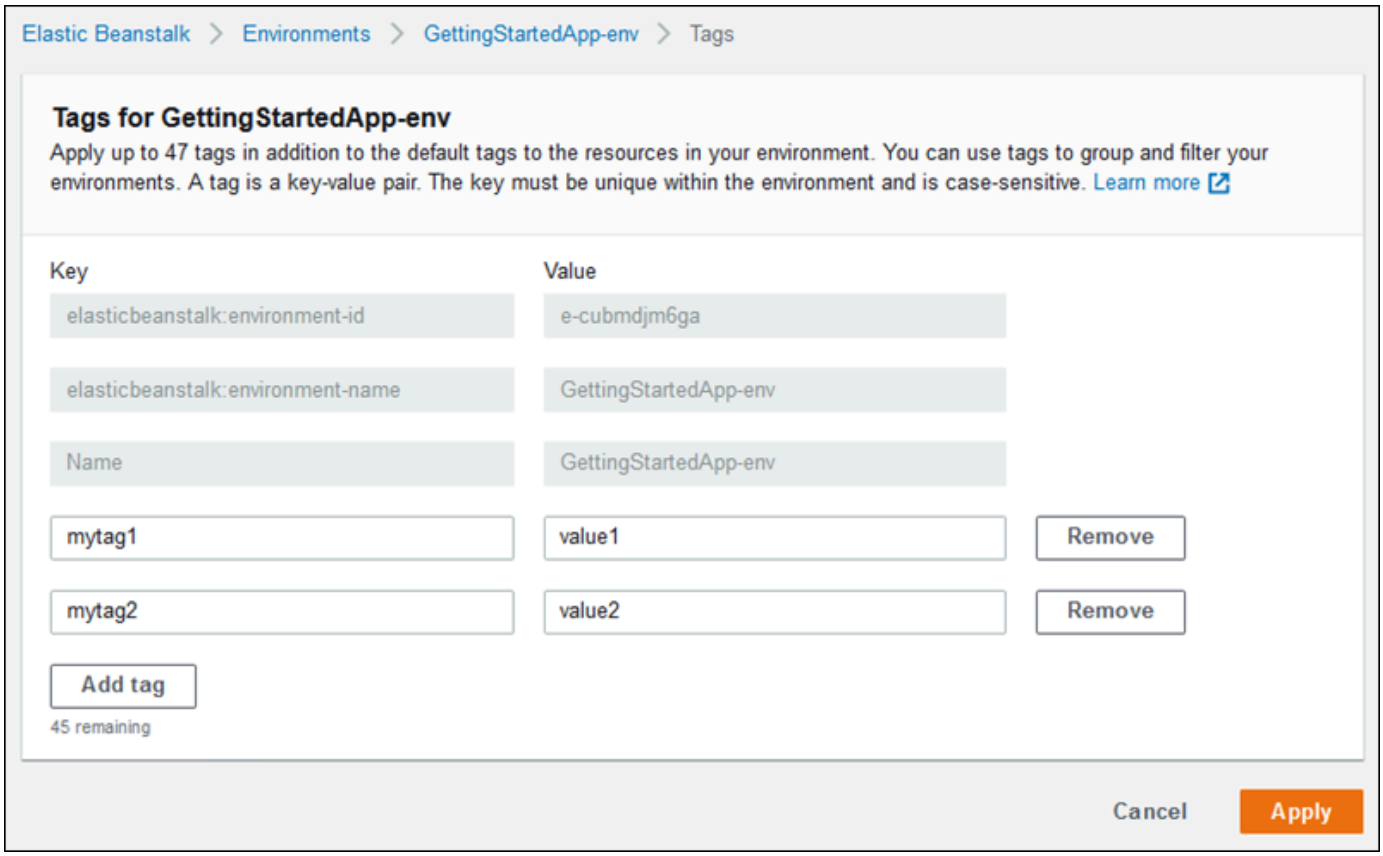
Sie können Tags in einer bestehenden Elastic Beanstalk-Umgebung hinzufügen, aktualisieren und löschen. Elastic Beanstalk wendet die Änderungen auf die Ressourcen Ihrer Umgebung an.

Es ist jedoch nicht möglich, die Standard-Tags zu bearbeiten, die Elastic Beanstalk für Ihre Umgebung nutzt.

So verwalten Sie die Tags einer Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Tags aus.

   Auf der Tag-Management-Seite wird die Liste der Tags angezeigt, die in der Umgebung vorhanden sind.



4. Sie können Tags hinzufügen, aktualisieren oder löschen:

   - Um ein Tag hinzuzufügen, geben Sie es in die leeren Felder am Ende der Liste ein. Um einen weiteren Tag hinzuzufügen, wählen Sie Add tag (Tag hinzufügen) aus. Elastic Beanstalk fügt ein weiteres Paar leerer Felder hinzu.
   - Um den Schlüssel oder Wert eines Tags zu aktualisieren, bearbeiten Sie die entsprechenden Felder in der Zeile des Tags.
   - Um ein Tag zu löschen, wählen Sie Remove (Entfernen) neben dem Wertfeld des Tags.

5. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Wenn Sie Umgebungen mit der EB CLI aktualisieren, verwenden Sie zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags den Befehl eb tags.

Mit dem folgenden Befehl werden beispielsweise die Tags in Ihrer Standardumgebung aufgelistet.

```
~/workspace/my-app$ eb tags --list
```

Mit dem folgenden Befehl wird das Tag `mytag1` aktualisiert und das Tag `mytag2` gelöscht.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter `eb tags`.

Verwenden Sie bei den AWS CLI oder anderen API-basierten Clients den list-tags-for-resource Befehl, um die Tags einer Umgebung aufzulisten.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-app/my-env"
```

Verwenden Sie den Befehl update-tags-for-resource zum Hinzufügen, Aktualisieren und Löschen von Tags in einer Umgebung.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-
app/my-env"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter `--tags-to-add` des Befehls update-tags-for-resource an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

> **Note**
>
> Um diese beiden AWS CLI Befehle in einer Elastic Beanstalk Beanstalk-Umgebung verwenden zu können, benötigen Sie den ARN der Umgebung. Verwenden Sie den folgenden Befehl, um den ARN abzurufen.

```
$ aws elasticbeanstalk describe-environments
```

Umgebungsvariablen und andere Softwareeinstellungen

Auf der Konfigurationsseite Updates, Überwachung und Protokollierung konfigurieren können Sie die Software auf den Amazon Elastic Compute Cloud (Amazon EC2) -Instances konfigurieren, auf denen Ihre Anwendung ausgeführt wird. Sie können Umgebungsvariablen, AWS X-Ray Debugging, Speichern und Streamen von Instance-Protokollen sowie plattformspezifische Einstellungen konfigurieren.

Themen

- Plattformspezifische Einstellungen konfigurieren
- Konfiguration von Umgebungseigenschaften (Umgebungsvariablen)
- Softwareeinstellung Namespaces
- Zugreifen auf Umgebungseigenschaften
- AWS X-Ray Debugging konfigurieren
- Anzeigen Ihrer Elastic Beanstalk-Umgebungsprotokolle

Plattformspezifische Einstellungen konfigurieren

Für die meisten Elastic Beanstalk-Plattformen können Sie neben den standardmäßigen Optionen für alle Umgebungen auch sprach- und Framework-spezifische Einstellungen festlegen. Diese werden im Abschnitt Plattformsoftware auf der Seite Aktualisierungen, Überwachung und Protokollierung konfigurieren angezeigt und können die folgenden Formen annehmen.

- Vordefinierte Umgebungseigenschaften – Die Ruby-Plattform verwendet Umgebungseigenschaften für Framework-Einstellungen wie `RACK_ENV` und `BUNDLE_WITHOUT`.
- Platzhalter-Umgebungseigenschaften – Die Tomcat-Plattform definiert eine Umgebungseigenschaft namens `JDBC_CONNECTION_STRING`, die keinen bestimmten Wert erhält. Solche Einstellungen waren auf älteren Plattformversionen gängiger.
- Konfigurationsoptionen – Die meisten Plattformen legen Konfigurationsoptionen in plattformspezifischen oder freigegebenen Namespaces wie `aws:elasticbeanstalk:xray` oder `aws:elasticbeanstalk:container:python` fest.

So konfigurieren Sie plattformspezifische Einstellungen in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Nehmen Sie unter Plattformsoftware die erforderlichen Änderungen an der Optionseinstellung vor.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Informationen zu den plattformspezifischen Optionen und zum Abrufen von Umgebungseigenschaftswerten in Ihrem Code finden Sie im Thema zu Plattformen für Ihre Sprache bzw. Ihr Framework:

- Docker – the section called “Konfiguration der Umgebung”
- Go – Verwenden der Go-Plattform von Elastic Beanstalk
- Java SE – Elastic Beanstalk Java SE-Plattform verwenden
- Tomcat – Elastic Beanstalk Tomcat-Plattform verwenden
- .NET Core unter Linux – Verwenden des Elastic Beanstalk Beanstalk-.NET-Kerns auf einer Linux-Plattform
- .NET – Verwendung der Elastic Beanstalk .NET Windows-Plattform
- Node.js – Verwenden der Elastic Beanstalk Node.js-Plattform
- PHP – Verwenden der Elastic Beanstalk PHP-Plattform
- Python – Verwenden der Elastic Beanstalk Python-Plattform
- Ruby – Verwenden der Elastic Beanstalk-Ruby-Plattform

Konfiguration von Umgebungseigenschaften (Umgebungsvariablen)

Sie können Umgebungseigenschaften (auch als Umgebungsvariablen bezeichnet) verwenden, um Endpunkte, Debug-Einstellungen und andere Informationen an Ihre Anwendung zu übergeben.

Umgebungsvariablen helfen Ihnen dabei, Ihre Anwendung in mehreren Umgebungen für unterschiedliche Zwecke wie Entwicklung, Testen, Staging und Produktion auszuführen.

Wenn Sie Ihrer Umgebung eine Datenbank hinzufügen, legt Elastic Beanstalk außerdem Umgebungsvariablen fest, die Sie beispielsweise in Ihrem Anwendungscode lesen können`RDS_HOSTNAME`, um ein Verbindungsobjekt oder eine Verbindungszeichenfolge zu erstellen.

So konfigurieren Sie Umgebungsvariablen in der Elastic Beanstalk Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Scrollen Sie nach unten zu Runtime-Umgebungsvariablen.
6. Wählen Sie Umgebungsvariable hinzufügen aus.
7. Wählen Sie als Quelle die Option Klartext aus.

   > **Note**
   >
   > Die Werte Secrets Manager und SSM Parameter Store in der Drop-down-Liste dienen zur Konfiguration von Umgebungsvariablen als Secrets zum Speichern sensibler Daten wie Anmeldeinformationen und API-Schlüssel. Weitere Informationen finden Sie unter Elastic Beanstalk mit AWS Secrets Manager und AWS Systems Manager Parameter Store verwenden.

8. Geben Sie den Namen der Umgebungsvariablen und die Wertepaare der Umgebungsvariablen ein.
9. Wenn Sie weitere Variablen hinzufügen müssen, wiederholen Sie Schritt 6 bis Schritt 8.
10. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Beschränkungen von Umgebungseigenschaften

- Schlüssel dürfen beliebige alphanumerische Zeichen sowie die folgenden Symbole enthalten: `_ . : / + \ - @`

  Die aufgeführten Symbole gelten für Umgebungseigenschaftsschlüssel, möglicherweise aber nicht für Umgebungsvariablennamen auf der Plattform Ihrer Umgebung. Wenn Sie sicherstellen möchten, dass Ihre Umgebungsvariablen mit allen Plattformen kompatibel sind, beschränken Sie sich auf folgendes Muster: `[A-Z_][A-Z0-9_]*`
- Werte dürfen beliebige alphanumerische Zeichen, Leerzeichen sowie die folgenden Symbole enthalten: `_ . : / = + \ - @ ' "`

  > **Note**
  >
  > Einige Zeichen in Umgebungseigenschaftswerten müssen durch das Escape-Zeichen geschützt werden. Verwenden Sie den umgekehrten Schrägstrich (\), um einige Sonderzeichen und Steuerzeichen darzustellen. Die folgende Liste enthält Beispiele für die Darstellung einiger Zeichen, die durch das Escape-Zeichen geschützt werden müssen:
  >
  > - umgekehrter Schrägstrich (\) – zur Darstellung `\\` verwenden
  > - einzelnes Anführungszeichen (') – zur Darstellung `\'` verwenden
  > - doppeltes Anführungszeichen (") – zur Darstellung `\"` verwenden
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.
- Die Gesamtgröße aller Umgebungseigenschaften darf 4.096 Byte nicht überschreiten, wenn sie als Zeichenfolgen im Format *`key`* = gespeichert werden. *`value`*

Softwareeinstellung Namespaces

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben durchführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Sie können mit den Konfigurationsdateien von Elastic Beanstalk Umgebungseigenschaften und Konfigurationsoptionen in Ihrem Quellcode festlegen. Verwenden Sie den `aws:elasticbeanstalk:application:environmentNamespace`, um Umgebungseigenschaften zu definieren.

Example.ebextensions/options.config

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    API_ENDPOINT: www.example.com/api
```

Wenn Sie Konfigurationsdateien oder CloudFormation Vorlagen verwenden, um benutzerdefinierte Ressourcen zu erstellen, können Sie eine CloudFormation Funktion verwenden, um Informationen über die Ressource abzurufen und sie während der Bereitstellung dynamisch einer Umgebungseigenschaft zuzuweisen. Das folgende Beispiel aus dem elastic-beanstalk-samples GitHub Repository verwendet die Ref-Funktion, um den ARN eines Amazon SNS SNS-Themas abzurufen, das es erstellt, und weist es einer Umgebungseigenschaft mit dem Namen zu. `NOTIFICATION_TOPIC`

> **Hinweise**
>
> - Wenn Sie eine CloudFormation Funktion verwenden, um eine Umgebungseigenschaft zu definieren, zeigt die Elastic Beanstalk Beanstalk-Konsole den Wert der Eigenschaft an, bevor die Funktion ausgewertet wird. Sie können das `get-config`-Plattformskript verwenden, um die Werte der Umgebungseigenschaften zu bestätigen, die für Ihre Anwendung verfügbar sind.
> - Die Multicontainer Docker-Plattform wird nicht zur Erstellung von Container-Ressourcen verwendet CloudFormation . Daher unterstützt diese Plattform die Definition von Umgebungseigenschaften mithilfe CloudFormation von Funktionen nicht.

Example.Ebextensions/sns-topic.config

```
Resources:
  NotificationTopic:
    Type: AWS::SNS::Topic

option_settings:
  aws:elasticbeanstalk:application:environment:
    NOTIFICATION_TOPIC: '`{"Ref" : "NotificationTopic"}`'
```

Sie können mithilfe dieses Features auch Informationen aus CloudFormation -Pseudoparametern bereitstellen. In diesem Beispiel wird die derzeitige Region abgerufen und einer Eigenschaft namens `AWS_REGION` zugewiesen.

Example.Ebextensions/env-regionname.config

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    AWS_REGION: ``{"Ref" : "AWS::Region"}``
```

Die meisten Elastic Beanstalk-Plattformen definieren zusätzliche Namespaces mit Optionen zur Konfiguration von Software, die auf der Instance ausgeführt wird, wie den Reverse-Proxy, der Anforderungen an Ihre Anwendung weiterleitet. Weitere Informationen zu den für Ihre Plattform verfügbaren Namespaces finden Sie unter:

- Go – Namespace der Go-Konfiguration
- Java SE – Java SE-Konfigurations-Namespace
- Tomcat – Tomcat-Konfigurations-Namespaces
- .NET Core unter Linux – .NET Core unter Linux-Konfigurations-Namespace
- .NET – Der Namespace aws:elasticbeanstalk:container:dotnet:apppool
- Node.js – Node.js-Konfigurations-Namespace
- PHP – Namespaces für die Konfiguration
- Python – Namespaces der Python-Konfiguration
- Ruby – Namespaces der Ruby-Konfigurationen

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Zugreifen auf Umgebungseigenschaften

In den meisten Fällen können Sie in Ihrem Anwendungscode auf Umgebungseigenschaften wie Umgebungsvariablen zugreifen. In der Regel werden Umgebungseigenschaften jedoch nur an die Anwendung übergeben und können nicht angezeigt werden, indem Sie sich mit einer Instance in Ihrer Umgebung verbinden und `env` ausführen.

- Go – `os.Getenv`

```
endpoint := os.Getenv("API_ENDPOINT")
```

- Java SE – `System.getenv`

```
String endpoint = System.getenv("API_ENDPOINT");
```

- Tomcat — `System.getProperty` und `System.getenv`

  Tomcat-Plattformversionen, die am oder nach dem 26. März 2025 veröffentlicht wurden, können auch für den `System.getenv` Zugriff auf Klartext-Umgebungsvariablen verwendet werden. Sie können es weiterhin für den Zugriff `System.getProperty` auf Klartext-Umgebungsvariablen verwenden. Umgebungsvariablen, die als Geheimnisse gespeichert sind, sind jedoch nur über `System.getenv` verfügbar.

```
String endpoint = System.getProperty("API_ENDPOINT");
```

```
String endpoint = System.getenv("API_ENDPOINT");
```

> **Important**
>
> Der zusätzliche `System.getenv` Zugriff auf Umgebungsvariablen in Versionen der Tomcat-Plattform, die am oder nach dem 26. März 2025 veröffentlicht wurden, kann zu unerwartetem Verhalten in Anwendungen führen, die Umgebungsvariablen Vorrang vor Java-Systemeigenschaften geben, oder wenn explizit von `System.getProperty` zu gewechselt wird. `System.getenv` Weitere Informationen und Handlungsempfehlungen finden Sie unter Verwenden der Elastic Beanstalk Tomcat-Plattform.

- .NET Core unter Linux – `Environment.GetEnvironmentVariable`

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

- .NET – `appConfig`

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;
string endpoint = appConfig["API_ENDPOINT"];
```

- Node.js – `process.env`

```
var endpoint = process.env.API_ENDPOINT
```

- PHP – `$_SERVER`

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

- Python – `os.environ`

```
import os
endpoint = os.environ['API_ENDPOINT']
```

- Ruby – `ENV`

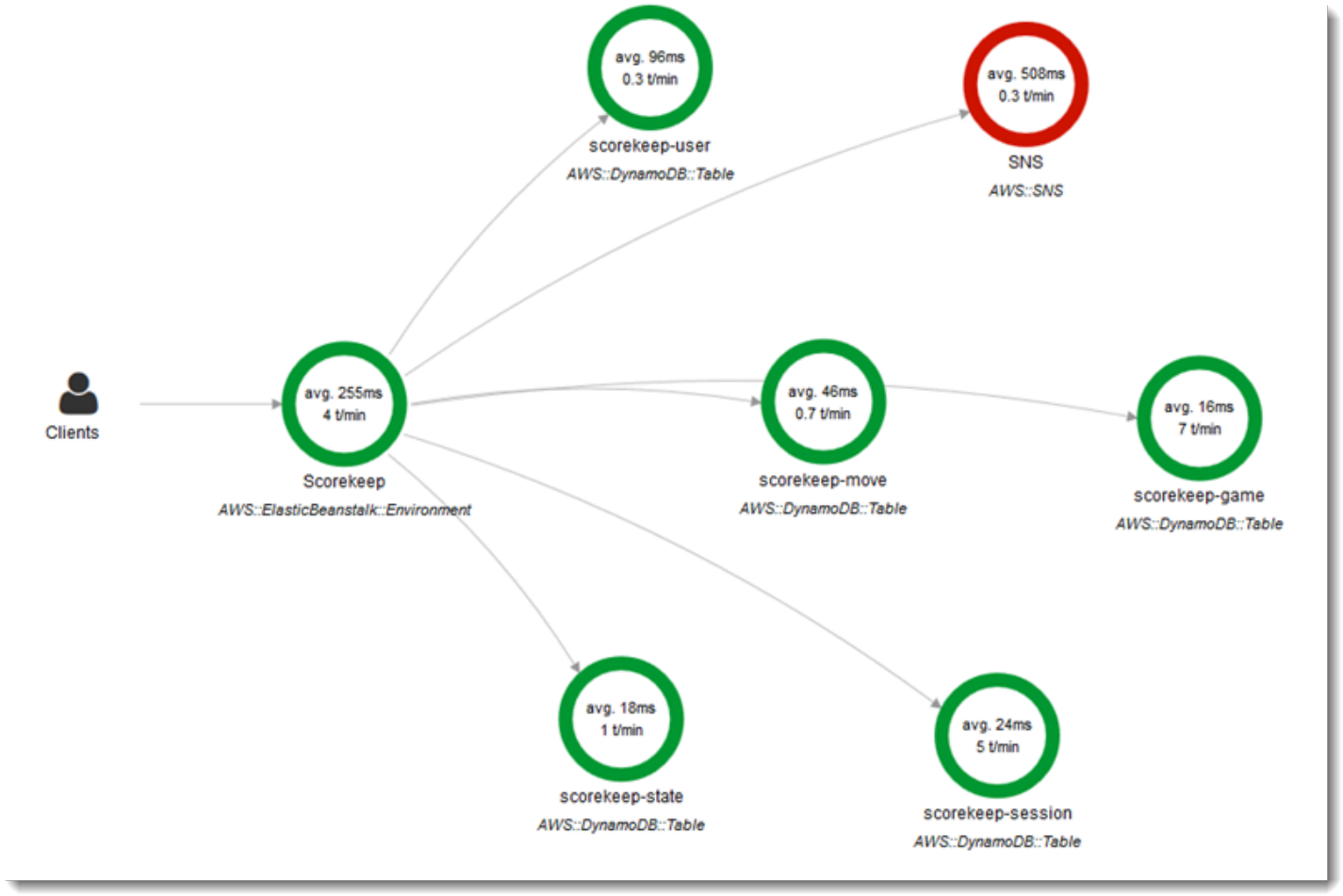
```
endpoint = ENV['API_ENDPOINT']
```

Außerhalb des Anwendungscodes, beispielsweise in Skripten, die während der Bereitstellung ausgeführt werden, können Sie auf Umgebungseigenschaften mit dem `get-config`-Plattformskript zugreifen. Im elastic-beanstalk-samples GitHub Repository finden Sie Beispielkonfigurationen, die `get-config` verwenden.

AWS X-Ray Debugging konfigurieren

Sie können die AWS Elastic Beanstalk Konsole oder eine Konfigurationsdatei verwenden, um den AWS X-Ray Daemon auf den Instanzen in Ihrer Umgebung auszuführen. X-Ray ist ein AWS Dienst, der Daten über die Anfragen sammelt, die Ihre Anwendung bedient, und anhand dieser Daten eine Service-Map erstellt, anhand derer Sie Probleme mit Ihrer Anwendung und Optimierungsmöglichkeiten identifizieren können.

> **Note**
>
> X-Ray wird nicht in allen Regionen angeboten. Wenn Sie eine Umgebung in einer dieser Regionen erstellen, können Sie den X-Ray-Daemon auf den Instances in Ihrer Umgebung nicht ausführen.
> Informationen zu den in den einzelnen Regionen angebotenen AWS Diensten finden Sie in der Regionentabelle.



X-Ray bietet ein SDK, das Sie verwenden können, um Ihren Anwendungscode zu instrumentalisieren, sowie eine Daemon-Anwendung, die Debugging-Informationen vom SDK an die X-Ray-API weiterleitet.

Unterstützte Plattformen

Sie können das X-Ray SDK mit den folgenden Elastic Beanstalk-Plattformen verwenden:

- Go – Version 2.9.1 und höher
- Java 8 – Version 2.3.0 und höher
- Java 8 mit Tomcat 8 – Version 2.4.0 und höher
- Node.js – Version 3.2.0 und höher
- Windows Server – alle Plattformversionen, die am oder nach dem 18. Dezember 2016 veröffentlicht wurden
- Python - Version 2.5.0 und höher

Auf unterstützten Plattformen können Sie eine Konfigurationsoption für die Ausführung des X-Ray-Daemon auf den Instances in Ihrer Umgebung verwenden. Sie können den Daemon in der Elastic Beanstalk-Konsole oder über eine Konfigurationsdatei aktivieren.

Um Daten auf X-Ray hochzuladen, benötigt der X-Ray-Daemon IAM-Berechtigungen in der AWSXrayWriteOnlyAccessverwalteten Richtlinie. Diese Berechtigungen sind im Elastic Beanstalk-Instance-Profil enthalten. Wenn Sie nicht das Standard-Instance-Profil verwenden, finden Sie Informationen unter Erteilen der Berechtigung zum Senden von Daten an X-Ray für den Daemon im AWS X-Ray -Entwicklerhandbuch.

Debugging mit X-Ray erfordert die Verwendung des X-Ray-SDK. Anweisungen und Beispielanwendungen finden Sie unter Erste Schritte mit AWS X-Ray im AWS X-Ray -Benutzerhandbuch.

Wenn Sie eine Plattformversion verwenden, die nicht über den Daemon verfügt, können Sie sie trotzdem mit einem Skript in einer Konfigurationsdatei ausführen. Weitere Informationen finden Sie unter Manuelles Herunterladen und Ausführen des X-Ray-Daemons (Erweitert) im AWS X-Ray -Entwicklerhandbuch.

Sections

- Konfigurieren des -Debuggings
- Der Namespace aws:elasticbeanstalk:xray

Konfigurieren des -Debuggings

Sie können den X-Ray-Daemon auf einer laufenden Umgebung in der Elastic Beanstalk-Konsole aktivieren.

So aktivieren Sie das Debugging in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Wählen Sie im Abschnitt Amazon X-Ray die Option Aktiviert aus.

6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Sie können diese Option auch während der Umgebungserstellung aktivieren. Weitere Informationen finden Sie unter Der Assistent zum Erstellen einer neuen Umgebung.

Der Namespace aws:elasticbeanstalk:xray

Sie können mit der `XRayEnabled`-Option im `aws:elasticbeanstalk:xray`-Namespace Debugging aktivieren.

Um das Debugging bei der Bereitstellung Ihrer Anwendung automatisch zu aktivieren, legen Sie die Option in einer Konfigurationsdatei in Ihrem Quellcode wie folgt fest.

Example.ebextensions/debugging.config

```
option_settings:
  aws:elasticbeanstalk:xray:
    XRayEnabled: true
```

Anzeigen Ihrer Elastic Beanstalk-Umgebungsprotokolle

AWS Elastic Beanstalk bietet zwei Möglichkeiten, regelmäßig Protokolle der EC2 Amazon-Instances einzusehen, auf denen Ihre Anwendung ausgeführt wird:

- Konfigurieren Sie Ihre Elastic Beanstalk-Umgebung, um gedrehte Instance-Protokolle in den Amazon S3-Bucket der Umgebung hochzuladen.
- Konfigurieren Sie die Umgebung so, dass Instanzprotokolle an Amazon CloudWatch Logs gestreamt werden.

Wenn Sie das Streaming von Instance-Logs in CloudWatch Logs konfigurieren, erstellt Elastic Beanstalk CloudWatch Log-Gruppen für Proxy- und Deployment-Logs auf den EC2 Amazon-Instances und überträgt diese Protokolldateien in Echtzeit an CloudWatch Logs. Weitere Informationen zu Instance-Protokollen finden Sie unter Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen.

Wenn Sie Enhanced Health für Ihre Umgebung aktivieren, können Sie zusätzlich zu den Instance-Logs die Umgebung so konfigurieren, dass Integritätsinformationen in Logs gestreamt

werden. CloudWatch Wenn sich der Zustand der Umgebung ändert, fügt Elastic Beanstalk einen Datensatz mit dem neuen Status und einer Beschreibung der Ursache der Änderung zu einer Zustandsprotokollgruppe hinzu. Weitere Informationen zum Streaming des Umgebungszustands finden Sie unter Zustandsinformationen der Elastic Beanstalk Beanstalk-Umgebung an Amazon Logs streamen CloudWatch.

Konfigurieren der Anzeige von Instance-Protokollen

Um Instance-Protokolle anzuzeigen, können Sie in der Elastic Beanstalk-Konsole die Rotation von Instance-Protokollen und das Protokoll-Streaming aktivieren.

So konfigurieren Sie die Rotation von Instance-Protokollen und das Protokoll-Streaming in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Wählen Sie im Abschnitt S3-Protokollspeicher die Option Aktiviert unter Protokolle rotieren aus, um das Hochladen rotierter Protokolle in Amazon S3 zu aktivieren.
6. konfigurieren Sie im Abschnitt CloudWatch Instanz-Log-Streaming in Logs die folgenden Einstellungen:
   - Protokollstreaming – Wählen Sie die Option Aktiviert aus, um das Protokollstreaming zu aktivieren.
   - Aufbewahrung — Geben Sie in Logs die Anzahl der Tage an, für die Logs aufbewahrt CloudWatch werden sollen.
   - Lebenszyklus — Wählen Sie die Option Protokolle bei Beendigung löschen, um CloudWatch Protokolle sofort aus den Protokollen zu löschen, wenn die Umgebung beendet wird, anstatt darauf zu warten, dass sie ablaufen.
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Nachdem Sie das Protokoll-Streaming aktiviert haben, können Sie zur Konfigurationskategorie oder -seite Software zurückkehren und dort den Link Log Groups (Protokollgruppen) suchen. Klicken Sie auf diesen Link, um Ihre Instance-Logs in der CloudWatch Konsole zu sehen.

Konfigurieren der Anzeige von Umgebungszustandsprotokollen

Um Umgebungszustandsprotokolle anzuzeigen, können Sie in der Elastic Beanstalk-Konsole das Streamen der Umgebungszustandsprotokolle aktivieren.

So konfigurieren Sie das Streamen von Umgebungszustandsprotokollen mit der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Gehen Sie zum Abschnitt Überwachung.
6. Konfigurieren Sie unter Health Event Streaming to CloudWatch Logs die folgenden Einstellungen:
   - Protokollstreaming – Wählen Sie Aktiviert aus, um das Protokollstreaming zu aktivieren.
   - Aufbewahrung — Geben Sie in Logs die Anzahl der Tage an, für die CloudWatch Protokolle aufbewahrt werden sollen.
   - Lebenszyklus — Wählen Sie die Option Protokolle bei Beendigung löschen, um CloudWatch Protokolle sofort aus den Protokollen zu löschen, wenn die Umgebung beendet wird, anstatt darauf zu warten, dass sie ablaufen.
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Namespace für das Anzeigen von Protokolldateien

Die folgenden Namespaces enthalten Einstellungen für die Protokollanzeige:

- `aws:elasticbeanstalk:hostmanager` – Konfiguration für das Hochladen von rotierten Protokollen in Amazon S3.
- `aws:elasticbeanstalk:cloudwatch:logs`— Konfigurieren Sie das Streaming von Instanzprotokollen auf CloudWatch.
- `aws:elasticbeanstalk:cloudwatch:logs:health`— Konfigurieren Sie das Streaming zur Umgebungsintegrität auf CloudWatch.

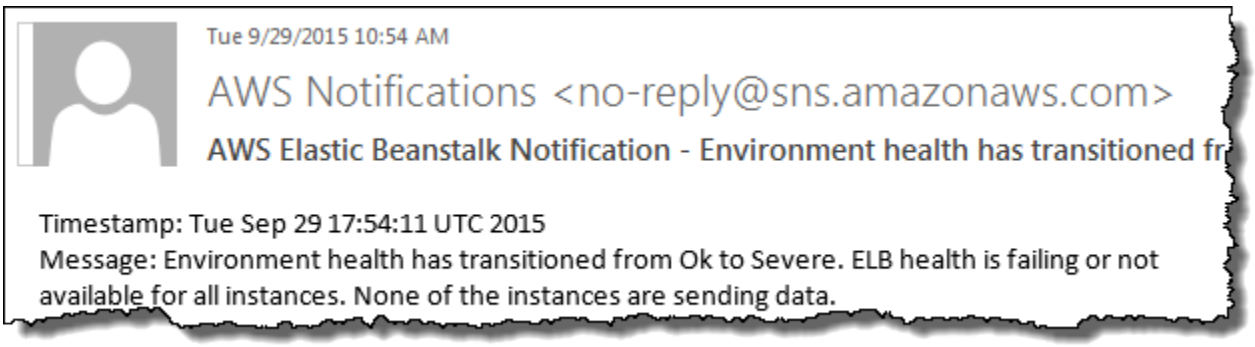
Elastic Beanstalk-Umgebungsbenachrichtigungen mit Amazon SNS

Sie können Ihre AWS Elastic Beanstalk Umgebung so konfigurieren, dass Amazon Simple Notification Service (Amazon SNS) verwendet wird, um Sie über wichtige Ereignisse zu informieren, die sich auf Ihre Anwendung auswirken. Um E-Mails zu erhalten, AWS wenn ein Fehler auftritt oder sich der Zustand Ihrer Umgebung ändert, geben Sie bei der Erstellung einer Umgebung oder zu einem späteren Zeitpunkt eine E-Mail-Adresse an.

> **Note**
>
> Elastic Beanstalk verwendet Amazon SNS für Benachrichtigungen. Informationen zu den Preisen von Amazon SNS finden Sie unter https://aws.amazon.com/sns/pricing/.

Wenn Sie Benachrichtigungen für Ihre Umgebung konfigurieren, erstellt Elastic Beanstalk in Ihrem Namen ein Amazon SNS-Thema für Ihre Umgebung. Um Nachrichten an ein Amazon SNS-Thema zu senden, muss Elastic Beanstalk über die erforderliche Berechtigung verfügen. Weitere Informationen finden Sie unter Konfigurieren von Berechtigungen zum Senden von Benachrichtigungen.

Wenn ein relevantes Ereignis eintritt, sendet Elastic Beanstalk eine Nachricht an das Thema. Anschließend leitet Amazon SNS die empfangenen Nachrichten an die Abonnenten des Themas weiter. Zu den besonderen Ereignissen zählen Fehler bei der Umgebungserstellung und alle Änderungen am Umgebungs- und Instance-Zustand. Ereignisse für Amazon EC2 Auto Scaling-Vorgänge (wie das Hinzufügen oder Entfernen von Instances zu bzw. aus der Umgebung) und andere Informationsereignisse lösen keine Benachrichtigungen aus.



Sie können beim Erstellen einer Umgebung oder zu einem späteren Zeitpunkt eine E-Mail-Adresse in die Elastic Beanstalk-Konsole eingeben. Dadurch wird ein Amazon SNS-Thema erstellt und abonniert. Elastic Beanstalk verwaltet den Lebenszyklus des Themas, löscht dieses nach Abschluss aus der Umgebung oder wenn Sie Ihre E-Mail-Adresse aus der Environment Management Console entfernen.

Der Namespace `aws:elasticbeanstalk:sns:topics` bietet Optionen zum Konfigurieren eines Amazon SNS-Themas mit Konfigurationsdateien, CLI oder einem SDK. Mit einer dieser Methoden können Sie den Subscriber-Typ und den Endpunkt konfigurieren. Als Subscriber-Typ können Sie eine Amazon SQS-Warteschlange oder HTTP-URL auswählen.

Sie können Amazon SNS-Benachrichtigungen lediglich aktivieren oder deaktivieren. Abhängig von der Größe und Zusammensetzung Ihrer Umgebung können sehr häufig Benachrichtigungen an das Thema gesendet werden. Für die Konfiguration von Benachrichtigungen, die unter bestimmten Umständen gesendet werden sollen, stehen Ihnen andere Optionen zur Verfügung Sie können bei Amazon ereignisgesteuerte Regeln einrichten, die Sie benachrichtigen EventBridge , wenn Elastic Beanstalk Ereignisse ausgibt, die bestimmte Kriterien erfüllen. Alternativ können Sie Ihre Umgebung so konfigurieren, dass benutzerdefinierte Messwerte veröffentlicht werden, und CloudWatchAmazon-Alarme einrichten, sodass Sie benachrichtigt werden, wenn diese Messwerte einen fehlerhaften Schwellenwert erreichen.

Benachrichtigungen mit der Elastic Beanstalk-Konsole konfigurieren

In der Elastic Beanstalk-Konsole können Sie eine E-Mail-Adresse eingeben, um ein Amazon SNS-Thema für Ihre Umgebung zu erstellen.

So konfigurieren Sie Benachrichtigungen in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region

2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Scrollen Sie nach unten zum Bereich E-Mail-Benachrichtigungen.
6. Geben Sie eine E-Mail-Adresse ein.
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Wenn Sie eine E-Mail-Adresse für Benachrichtigungen eingeben, erstellt Elastic Beanstalk ein Amazon SNS-Thema für Ihre Umgebung und fügt ein Abonnement hinzu. Amazon SNS sendet eine E-Mail an die registrierte Adresse, um das Abonnement zu bestätigen. Zum Aktivieren des Abonnements und zum Empfangen von Benachrichtigungen müssen Sie auf den Link in der Bestätigungs-E-Mail klicken.

Konfigurieren von Benachrichtigungen mithilfe von Konfigurationsoptionen

Verwenden Sie die Optionen im `aws:elasticbeanstalk:sns:topics`-Namespace, um Amazon SNS-Benachrichtigungen für Ihre Umgebung zu konfigurieren. Sie können diese Optionen über Konfigurationsdateien, eine CLI oder ein SDK festlegen.

- Benachrichtigungsendpunkt – Die E-Mail-Adresse, Amazon SQS-Warteschlange oder URL, an die Benachrichtigungen gesendet werden sollen. Wenn Sie diesen Option einstellen, werden für den angegebenen Endpunkt eine SQS-Warteschlange und ein Abonnement erstellt. Wenn der Endpunkt keine E-Mail-Adresse ist, müssen Sie zudem die `Notification Protocol`-Option festlegen. SNS validiert den Wert von `Notification Endpoint` basierend auf dem Wert von `Notification Protocol`. Wenn Sie diese Option mehrfach festlegen, werden zusätzliche Abonnements für das Thema erstellt. Wenn Sie diese Option entfernen, wird das Thema gelöscht.
- Benachrichtigungsprotokoll – Das Protokoll, das zum Senden von Benachrichtigungen an den `Notification Endpoint` verwendet wird. Diese Option lautet standardmäßig `email`. Legen Sie diese Option auf `email-json` fest, um JSON-formatierte E-Mails zu senden, auf `http` oder `https`, um JSON-formatierte Benachrichtigungen an einen HTTP-Endpunkt zu posten, oder auf `sqs`, um Benachrichtigungen an eine SQS-Warteschlange zu übermitteln.

> **Note**
>
> AWS Lambda Benachrichtigungen werden nicht unterstützt.

- ARN des Benachrichtigungsthemas – Nach dem Festlegen eines Benachrichtungsendpunkts für Ihre Umgebung lesen Sie diese Einstellung, um den ARN des SNS-Themas abzurufen. Sie können diese Option auch festlegen, um ein bestehendes SNS-Thema für Benachrichtigungen zu verwenden. Ein Thema, das Sie mit dieser Option an Ihre Umgebung anfügen, wird nicht gelöscht, wenn Sie diese Option ändern oder die Umgebung beenden.

  Um Amazon SNS-Benachrichtigungen zu konfigurieren, benötigen Sie die erforderlichen Berechtigungen. Wenn Ihr IAM-Benutzer die Richtlinie Elastic Beanstalk AdministratorAccess — AWSElastic Beanstalk für verwaltete Benutzer verwendet, sollten Sie bereits über die erforderlichen Berechtigungen verfügen, um das Amazon SNS SNS-Standardthema zu konfigurieren, das Elastic Beanstalk für Ihre Umgebung erstellt. Wenn Sie jedoch ein Amazon SNS-Thema konfigurieren, das nicht von Elastic Beanstalk verwaltet wird, müssen Sie Ihrer Benutzerrolle die folgende Richtlinie hinzufügen.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:SetTopicAttributes",
        "sns:GetTopicAttributes",
        "sns:Subscribe",
        "sns:Unsubscribe",
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:sns_topic_name"
      ]
    }
  ]
}
```

- Name des Benachrichtigungsthemas – Legen Sie diese Option fest, um den Namen des Amazon-SNS-Themas anzupassen, der für Umgebungsbenachrichtigungen verwendet wird. Wenn ein Thema mit demselben Namen bereits vorhanden ist, fügt Elastic Beanstalk dieses Thema an die Umgebung an.

  > **Warning**
  >
  > Wenn Sie ein vorhandenes SNS-Thema mit `Notification Topic Name` an eine Umgebung anfügen, löscht Elastic Beanstalk das Thema, falls Sie in Zukunft die Umgebung beenden oder diese Einstellung ändern.

  Wenn Sie diese Option ändern, wird auch der `Notification Topic ARN` geändert. Wenn ein Thema bereits an die Umgebung angefügt ist, löscht Elastic Beanstalk das alte Thema und erstellt dann ein neues Thema und ein neues Abonnement.

  Wenn Sie einen benutzerdefinierten Themennamen verwenden, müssen Sie auch einen ARN eines extern erstellten benutzerdefinierten Themas angeben. Die Richtlinie für verwaltete Benutzer erkennt ein Thema mit einem benutzerdefinierten Namen nicht automatisch, so dass Sie Ihren IAM-Benutzern benutzerdefinierte Amazon SNS-Berechtigungen erteilen müssen. Verwenden Sie eine ähnliche Richtlinie wie diejenige, die für ein benutzerdefiniertes Thema ARN verwendet wird, jedoch mit den folgenden Ergänzungen:

  - Nehmen Sie zwei weitere Aktionen in die `Actions`-Liste auf, insbesondere: `sns:CreateTopic`, `sns:DeleteTopic`
  - Wenn Sie den `Notification Topic Name` Namen eines benutzerdefinierten Themas in einen anderen ändern, müssen Sie auch den Namen beider Themen in die Liste aufnehmen. ARNs `Resource` Alternativ können Sie einen regulären Ausdruck einschließen, der beide Themen abdeckt. Auf diese Weise hat Elastic Beanstalk die Berechtigung, das alte Thema zu löschen und das neue zu erstellen.

Die EB-CLI und Elastic-Beanstalk-Konsole wenden die für die vorhergehenden Optionen empfohlenen Werte an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter Empfohlene Werte.

Konfigurieren von Berechtigungen zum Senden von Benachrichtigungen

In diesem Abschnitt werden Sicherheitsüberlegungen erläutert, die sich auf Benachrichtigungen beziehen, die Amazon SNS verwenden. Es gibt zwei verschiedene Fälle:

- Verwenden des standardmäßige Amazon SNS-Thema, das Elastic Beanstalk für Ihre Umgebung erstellt.
- Bereitstellen eines externen Amazon SNS-Themas über Konfigurationsoptionen.

Die Standardzugriffsrichtlinie für ein Amazon SNS-Thema erlaubt nur dem Besitzer des Themas, es zu veröffentlichen oder zu abonnieren. Durch die richtige Richtlinienkonfiguration kann Elastic Beanstalk jedoch die Erlaubnis erteilt werden, in einem der beiden in diesem Abschnitt beschriebenen Fälle ein Amazon SNS-Thema zu veröffentlichen. Weitere Informationen finden Sie in den folgenden Abschnitten.

Berechtigungen für ein Standard-Thema

Wenn Sie Benachrichtigungen für Ihre Umgebung konfigurieren, erstellt Elastic Beanstalk ein Amazon SNS-Thema für Ihre Umgebung. Um Nachrichten an ein Amazon SNS-Thema zu senden, muss Elastic Beanstalk über die erforderliche Berechtigung verfügen. Wenn die Umgebung die durch die Elastic Beanstalk-Konsole oder die EB-CLI für die Umgebung generierte Service-Rolle oder die Service-verknüpfte Überwachungsrolle des Kontos verwendet, sind keine weiteren Schritte erforderlich. Diese verwalteten Rollen enthalten die erforderliche Berechtigung, die es Elastic Beanstalk ermöglicht, Nachrichten an das Amazon SNS-Thema zu senden.

Wenn Sie jedoch beim Erstellen Ihrer Umgebung eine benutzerdefinierte Service-Rolle bereitgestellt haben, müssen Sie sicherstellen, dass diese benutzerdefinierte Service-Rolle die folgende Richtlinie enthält.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
```

```
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:ElasticBeanstalkNotifications*"
      ]
    }
  ]
}
```

Berechtigungen für ein externes Thema

Konfigurieren von Benachrichtigungen mithilfe von Konfigurationsoptionen erklärt, wie Sie das Amazon SNS-Thema von Elastic Beanstalk durch ein anderes Amazon SNS-Thema ersetzen können. Wenn Sie das Thema ersetzt haben, muss Elastic Beanstalk überprüfen, ob Sie die Berechtigung haben, zu diesem SNS-Thema zu veröffentlichen, damit Sie das SNS-Thema mit der Umgebung verknüpfen können. Sie sollten `sns:Publish` haben. Die Servicerolle verwendet dieselbe Berechtigung. Um dies zu überprüfen, sendet Elastic Beanstalk als Teil Ihrer Aktion zum Erstellen oder Aktualisieren der Umgebung eine Test-Benachrichtigung an SNS. Wenn dieser Test fehlschlägt, scheitert auch Ihr Versuch, die Umgebung zu erstellen oder zu aktualisieren. Elastic Beanstalk zeigt eine Meldung an, die den Grund für diesen Fehler erläutert.

Wenn Sie eine benutzerdefinierte Servicerolle für Ihre Umgebung bereitstellen, müssen Sie sicherstellen, dass Ihre benutzerdefinierte Servicerolle die folgende Richtlinie enthält, damit Elastic Beanstalk Nachrichten an das Amazon SNS-Thema senden kann. Ersetzen Sie im folgenden Code *sns_topic_name* durch den Namen des Amazon SNS-Themas, das Sie in den Konfigurationsoptionen angegeben haben.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:sns_topic_name"
      ]
    }
```

```
  ]
}
```

Weitere Informationen zur Amazon SNS-Zugriffssteuerung finden Sie unter Beispielfälle für Amazon SNS Zugriffskontrolle im Entwicklerhandbuch für Amazon Simple Notification .

Amazon Virtual Private Cloud (Amazon VPC) mit Elastic Beanstalk konfigurieren

Amazon Virtual Private Cloud (Amazon VPC) ist der Netzwerkservice, der den Datenverkehr sicher an die EC2 Instances weiterleitet, auf denen Ihre Anwendung in Elastic Beanstalk ausgeführt wird. Wenn Sie beim Start Ihrer Umgebung keine VPC konfigurieren, verwendet Elastic Beanstalk die standardmäßige VPC.

Sie können Ihre Umgebung in einer benutzerdefinierten VPC starten, um die Netzwerk- und Sicherheitseinstellungen anzupassen. Mit Elastic Beanstalk können Sie wählen, welche Subnetze Sie für Ihre Ressourcen verwenden möchten und wie IP-Adressen für die Instances und Load Balancer in Ihrer Umgebung konfiguriert werden sollen. Eine Umgebung ist für eine VPC gesperrt, wenn Sie sie erstellen. Sie können aber Subnetz- und IP-Adresseinstellungen in einer ausgeführten Umgebung ändern.

VPC-Einstellungen in der Elastic Beanstalk-Konsole konfigurieren

Wenn Sie beim Erstellen der Umgebung eine benutzerdefinierte VPC ausgewählt haben, können Sie die VPC-Einstellungen in der Elastic Beanstalk-Konsole ändern.

So konfigurieren Sie die VPC-Einstellungen Ihrer Umgebung

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Network (Netzwerk) die Option Edit (Bearbeiten).

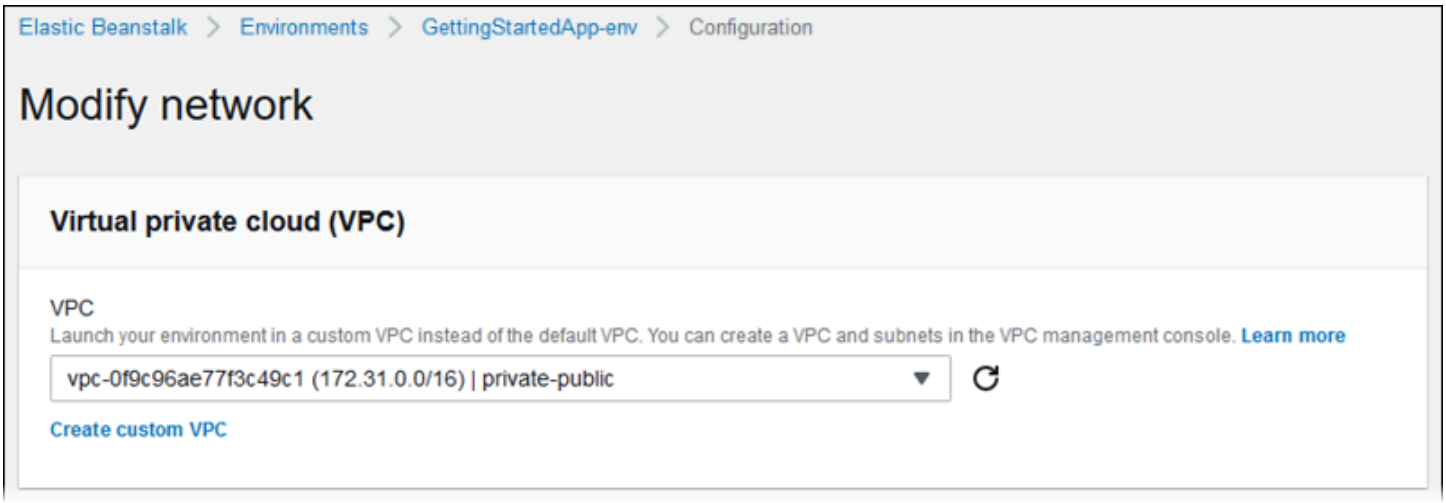
Die folgenden Einstellungen sind verfügbar.

Optionen

- VPC
- Load Balancer-Sichtbarkeit
- Load Balancer-Subnetze
- Öffentliche IP-Adresse der Instance
- Instance-Subnetze
- Datenbanksubnetze

VPC

Wählen Sie eine VPC für die Umgebung. Sie können diese Einstellung nur während des Erstellens der Umgebung ändern.



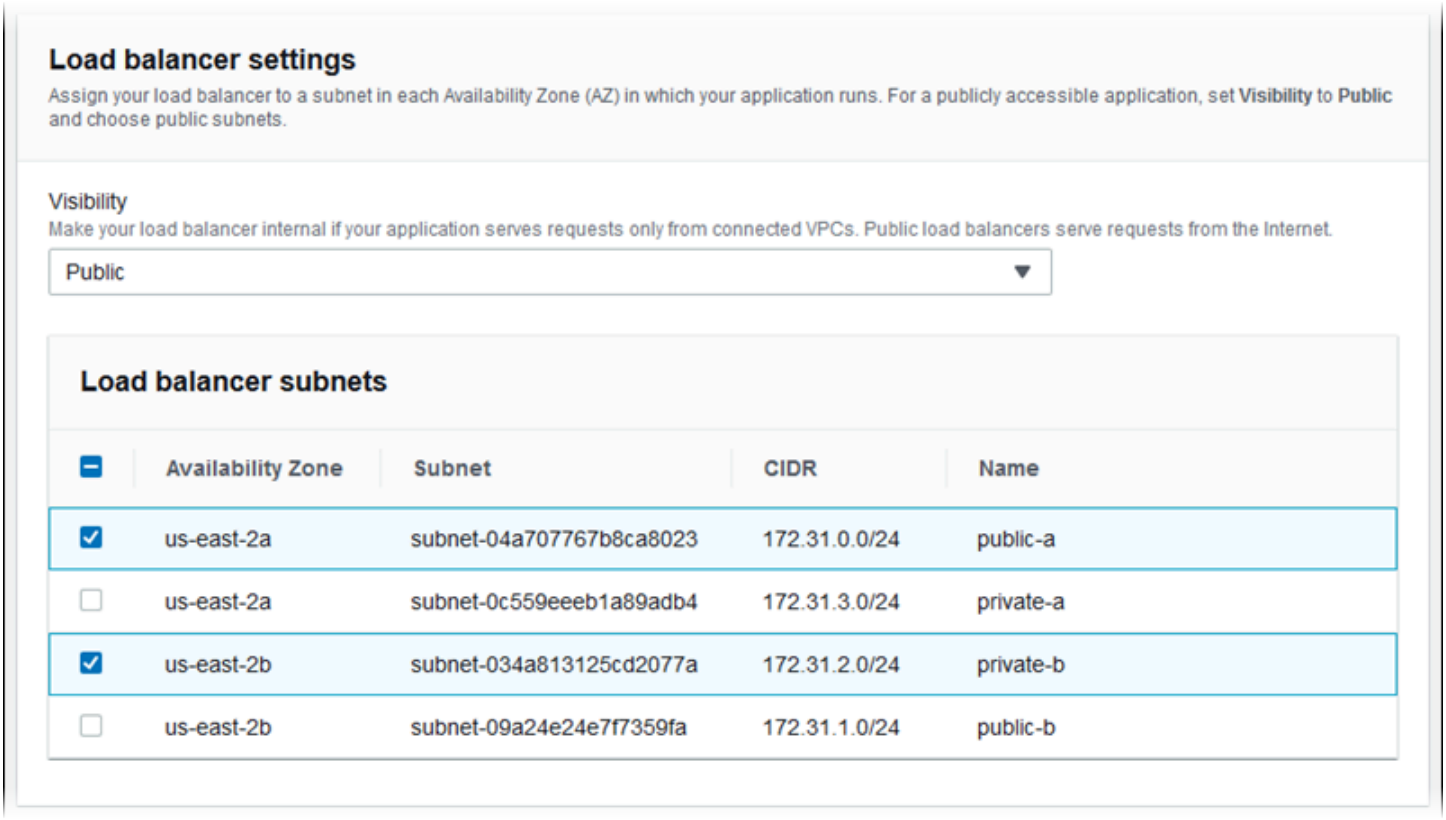
Load Balancer-Sichtbarkeit

Wählen Sie für eine Load Balancer-Umgebung das Load Balancer-Schema. Standardmäßig ist der Load Balancer öffentlich mit einer öffentlichen IP-Adresse und einem öffentlichen Domänennamen. Wenn Ihre Anwendung nur Datenverkehr innerhalb Ihrer VPC oder über eine verbundene VPN abwickelt, deaktivieren Sie diese Option und wählen Sie private Subnetze für Ihren Load Balancer aus, um den Load Balancer als intern zu kennzeichnen und den Zugriff über das Internet zu deaktivieren.

Load Balancer-Subnetze

Wählen Sie für eine Load Balancer-Umgebung die Subnetze, die der Load Balancer verwendet, um den Datenverkehr abzuwickeln. Wählen Sie für eine öffentliche Anwendung öffentliche Subnetze.

Verwenden Sie Subnetze in mehreren Availability Zones, um eine hohe Verfügbarkeit zu erzielen. Wählen Sie für eine interne Anwendung private Subnetze und deaktivieren Sie die Load Balancer-Sichtbarkeit.

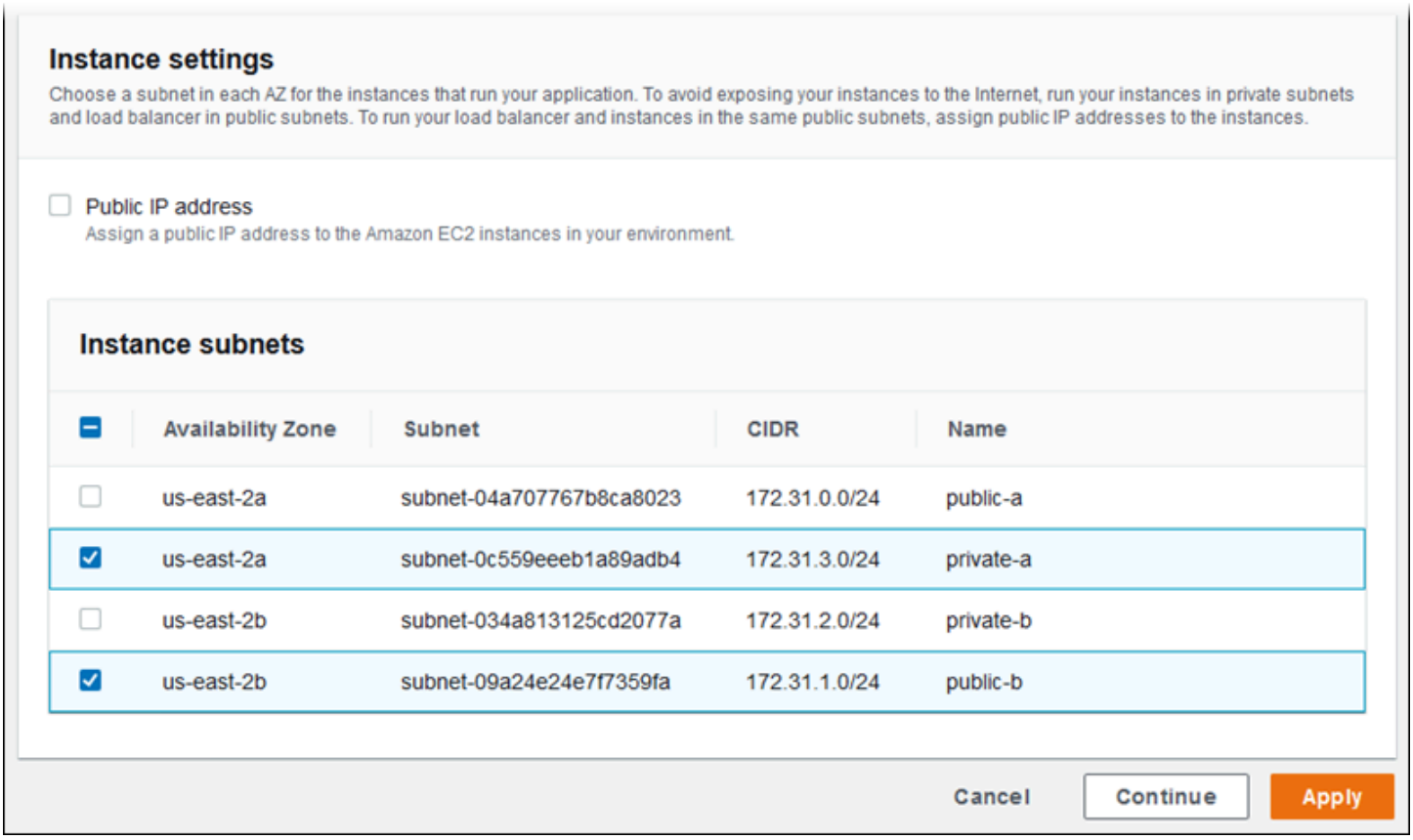


Öffentliche IP-Adresse der Instance

Wenn Sie öffentliche Subnetze für Ihre Anwendungs-Instances wählen, aktivieren Sie öffentliche IP-Adressen, damit diese über das Internet weitergeleitet werden können.

Instance-Subnetze

Wählen Sie Subnetze für Ihre Anwendungs-Instances. Wählen Sie mindestens ein Subnetz für jede Availability Zone, die Ihr Load Balancer verwendet. Wenn Sie private Subnetze für Ihre Instances verwenden, muss Ihre VPC über einen NAT-Gateway in einem öffentlichen Subnetz verfügen, den die Instances für den Internetzugriff verwenden können.



Datenbanksubnetze

Wenn Sie eine Amazon RDS-Datenbank ausführen, die Ihrer Elastic Beanstalk-Umgebung angehängt ist, wählen Sie Subnetze für Ihre Datenbank-Instances aus. Hohe Verfügbarkeit erzielen Sie, indem Sie die Datenbank zu einer Multi-AZ-Datenbank machen und ein Subnetz für jede Availability Zone wählen. Um sicherzustellen, dass Ihre Anwendung eine Verbindung zu Ihrer Datenbank herstellen kann, führen Sie beide in denselben Subnetzen aus.

Der aws:ec2:vpc-Namespace

Sie können die Konfigurationsoptionen im `aws:ec2:vpc`-Namespace für die Konfiguration der Netzwerkeinstellungen Ihrer Umgebung verwenden.

Die folgende Konfigurationsdatei verwendet Optionen in diesem Namespace, um die VPC und Subnetze der Umgebung für eine öffentlich-private Konfiguration festzulegen. Zum Festlegen der VPC-ID in einer Konfigurationsdatei muss die Datei im Anwendungsquellen-Bundle während der Erstellung der Umgebung enthalten sein. Unter Festlegen von Konfigurationsoptionen während der Umgebungserstellung finden Sie weitere Methoden zum Konfigurieren dieser Einstellungen während des Erstellens der Umgebung.

Example.ebextensions/vpc.config – Public-private

```
option_settings:
   aws:ec2:vpc:
      VPCId: vpc-087a68c03b9c50c84
      AssociatePublicIpAddress: 'false'
      ELBScheme: public
      ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
      Subnets: subnet-026c6117b178a9c45,subnet-0839e902f656e8bd1
```

Dieses Beispiel zeigt eine öffentlich-öffentliche Konfiguration, bei der der Load Balancer und die EC2 Instances in denselben öffentlichen Subnetzen ausgeführt werden.

Example.ebextensions/vpc.config – Public-public

```
option_settings:
   aws:ec2:vpc:
      VPCId: vpc-087a68c03b9c50c84
      AssociatePublicIpAddress: 'true'
      ELBScheme: public
      ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
      Subnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
```

Migration von Elastic Beanstalk Beanstalk-Umgebungen von EC2 -Classic zu einer VPC

In diesem Thema werden verschiedene Optionen für die Migration Ihrer Elastic Beanstalk Beanstalk-Umgebungen von einer EC2 -Classic-Netzwerkplattform zu einem Amazon Virtual Private Cloud (Amazon VPC) -Netzwerk beschrieben.

Wenn Sie Ihr AWS Konto vor dem 4. Dezember 2013 erstellt haben, haben Sie möglicherweise Umgebungen, die in einigen Umgebungen die EC2 -Classic-Netzwerkkonfiguration verwenden. AWS-Regionen Alle AWS Konten, die am oder nach dem 4. Dezember 2013 erstellt wurden, sind in jeder AWS Region bereits ausschließlich VPC-fähig. Die einzigen Ausnahmen sind, wenn Amazon EC2 -Classic aufgrund einer Support-Anfrage aktiviert wurde.

> **Note**
>
> Sie können die Konfigurationseinstellungen für das Netzwerk Ihrer Umgebung in der Kategorie Network configuration (Netzwerkkonfiguration) auf der Seite Configuration overview (Konfigurationsübersicht) in der Elastic-Beanstalk-Konsole anzeigen.

Warum Sie migrieren sollten

Amazon EC2 -Classic wird am 15. August 2022 das Ende des Standardsupports erreichen. Um Unterbrechungen Ihrer Workloads zu vermeiden, empfehlen wir Ihnen, vor dem 15. August 2022 von Amazon EC2 -Classic zu einer VPC zu migrieren. Wir bitten Sie außerdem, in future keine AWS Ressourcen auf Amazon EC2 -Classic zu starten und stattdessen Amazon VPC zu verwenden.

Wenn Sie Ihre Elastic Beanstalk Beanstalk-Umgebungen von Amazon EC2 -Classic zu Amazon VPC migrieren, müssen Sie ein neues Konto erstellen. AWS Sie müssen auch Ihre AWS EC2 -Classic-Umgebungen in Ihrem neuen Konto neu erstellen. AWS Für die Verwendung der Standard-VPC sind keine zusätzlichen Konfigurationsarbeiten für Ihre Umgebungen erforderlich. Wenn die Standard-VPC Ihren Anforderungen nicht entspricht, können Sie manuell eine benutzerdefinierte VPC erstellen und sie Ihren Umgebungen zuordnen.

Wenn Ihr vorhandenes AWS Konto Ressourcen enthält, die Sie nicht auf ein neues AWS Konto migrieren können, können Sie Ihrem aktuellen Konto alternativ eine VPC hinzufügen. Konfigurieren Sie dann Ihre Umgebungen für die Verwendung der VPC.

Weitere Informationen finden Sie im Blogbeitrag „EC2Classic Networking is Retiring — So bereiten Sie sich vor“.

Migrieren Sie eine Umgebung von EC2 -Classic in ein neues AWS Konto (empfohlen)

Wenn Sie noch kein AWS Konto haben, das am oder nach dem 4. Dezember 2013 erstellt wurde, erstellen Sie ein neues Konto. Sie werden Ihre Umgebungen zu diesem neuen Konto migrieren.

1. Ihr neues AWS Konto stellt eine Standard-VPC für seine Umgebungen bereit. Wenn Sie keine benutzerdefinierte VPC erstellen müssen, fahren Sie mit Schritt 2 fort.

   Sie können eine benutzerdefinierte VPC auf eine der folgenden Arten erstellen:

- Erstellen Sie schnell eine VPC mit dem Amazon-VPC-Konsolenassistenten mit einer der verfügbaren Konfigurationsoptionen. Weitere Informationen finden Sie unter Konfigurationen über den Amazon VPC-Konsolenassistent.
- Erstellen Sie eine benutzerdefinierte VPC auf der Amazon-VPC-Konsole, wenn Sie spezifischere Anforderungen an Ihre VPC haben. Wir empfehlen Ihnen, dies beispielsweise zu tun, wenn Ihr Anwendungsfall eine bestimmte Anzahl von Subnetzen erfordert. Weitere Informationen finden Sie unter VPCs Subnetze.
- Erstellen Sie mithilfe des elastic-beanstalk-samplesRepositorys auf der GitHub Website eine VPC, wenn Sie lieber AWS CloudFormation Vorlagen mit Ihren Elastic Beanstalk Beanstalk-Umgebungen verwenden möchten. Dieses Repository enthält Vorlagen. CloudFormation Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon VPC.

> **Note**
>
> Sie können auch eine benutzerdefinierte VPC erstellen, während Sie die Umgebung in Ihrem neuen AWS Konto mithilfe des Assistenten zum Erstellen einer neuen Umgebung neu erstellen. Wenn Sie den Assistenten verwenden und eine benutzerdefinierte VPC erstellen möchten, leitet Sie der Assistent zur Amazon-VPC-Konsole weiter.

2. Erstellen Sie in Ihrem neuen AWS Konto eine neue Umgebung. Wir empfehlen, dass die Umgebung dieselbe Konfiguration wie Ihre bestehende Umgebung in dem AWS Konto enthält, von dem Sie migrieren. Sie können dies auf eine der folgenden Arten ausführen.

> **Note**
>
> Wenn Ihre neue Umgebung nach der Migration denselben CNAME verwenden muss, beenden Sie die ursprüngliche Umgebung auf der EC2 -Classic-Plattform. Dies gibt den CNAME zur Verwendung frei. Dies kann jedoch zu Ausfallzeiten für diese Umgebung führen und es besteht auch das Risiko, dass ein anderer Kunde Ihren CNAME zwischen dem Beenden Ihrer EC2 -Classic-Umgebung und dem Erstellen der neuen verwendet. Weitere Informationen finden Sie unter Elastic Beanstalk-Umgebung terminieren.
> Für Umgebungen mit einem eigenen proprietären Domänennamen gibt es dieses Problem für den CNAME nicht. Sie können einfach Ihr Domain Name System (DNS) aktualisieren, sodass Anfragen an den neuen CNAME weitergeleitet werden.

- Verwenden Sie den Assistenten für die Erstellung einer neuen Umgebung in der Elastic Beanstalk-Konsole. Der Assistent enthält eine Option zum Erstellen einer benutzerdefinierten VPC. Wenn Sie keine benutzerdefinierte VPC erstellen möchten, wird eine Standard-VPC zugewiesen.
- Verwenden Sie die Elastic Beanstalk Command Line Interface (EB CLI), um Ihre Umgebung in Ihrem neuen Konto neu zu erstellen. AWS Eines der Beispiele in der Befehlsbeschreibung eb create zeigt die Erstellung einer Umgebung in einer benutzerdefinierten VPC. Wenn Sie keine VPC-ID angeben, verwendet die Umgebung die Standard-VPC.

  Mit diesem Ansatz können Sie eine gespeicherte Konfigurationsdatei für beide Konten verwenden. AWS Daher müssen Sie nicht alle Konfigurationsinformationen manuell eingeben. Sie müssen jedoch die Konfigurationseinstellungen für die EC2 -Classic-Umgebung, die Sie migrieren, mit dem Befehl eb config save speichern. Kopieren Sie die gespeicherte Konfigurationsdatei in ein neues Verzeichnis für die Umgebung im neuen Konto.

  > **Note**
  >
  > Sie müssen einige Daten in der gespeicherten Konfigurationsdatei bearbeiten, bevor Sie diese im neuen Konto verwenden können. Sie müssen auch Informationen zu Ihrem vorherigen Konto mit den richtigen Daten für Ihr neues Konto aktualisieren. Beispielsweise müssen Sie den Amazon-Ressourcennamen (ARN) der AWS Identity and Access Management (IAM)-Rolle durch den ARN der IAM-Rolle für das neue Konto ersetzen.

  Wenn Sie den Befehl eb create mit `cfg` verwenden, wird die neue Umgebung mit der angegebenen gespeicherten Konfigurationsdatei erstellt. Weitere Informationen finden Sie unter Verwenden gespeicherter Elastic Beanstalk-Konfigurationen.

Migrieren Sie eine Umgebung von EC2 -Classic innerhalb desselben Kontos AWS

Ihr vorhandenes AWS Konto enthält möglicherweise Ressourcen, die Sie nicht auf ein neues AWS Konto migrieren können. In diesem Fall müssen Sie Ihre Umgebungen neu erstellen und eine VPC für jede von Ihnen erstellte Umgebung manuell konfigurieren.

Migration Ihrer Umgebungen zu einer benutzerdefinierten VPC

Voraussetzungen

Bevor Sie beginnen, müssen Sie über eine VPC verfügen. Sie können eine nicht standardmäßige (benutzerdefinierte) VPC auf eine der folgenden Arten erstellen:

- Erstellen Sie schnell eine VPC mit dem Amazon-VPC-Konsolenassistenten mit einer der verfügbaren Konfigurationsoptionen. Weitere Informationen finden Sie unter Konfigurationen über den Amazon VPC-Konsolenassistent.
- Erstellen Sie eine benutzerdefinierte VPC auf der Amazon-VPC-Konsole, wenn Sie spezifischere Anforderungen an Ihre VPC haben. Wir empfehlen Ihnen, dies beispielsweise zu tun, wenn Ihr Anwendungsfall eine bestimmte Anzahl von Subnetzen erfordert. Weitere Informationen finden Sie unter VPCs und Subnetze.
- Erstellen Sie mithilfe des elastic-beanstalk-samplesRepositorys auf der GitHub Website eine VPC, wenn Sie lieber AWS CloudFormation Vorlagen mit Ihren Elastic Beanstalk Beanstalk-Umgebungen verwenden möchten. Dieses Repository enthält Vorlagen. CloudFormation Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon VPC.

In den folgenden Schritten verwenden Sie die generierte VPC-ID und das generierte Subnetz, IDs wenn Sie die VPC in der neuen Umgebung konfigurieren.

1. Erstellen Sie eine neue Umgebung mit der gleichen Konfiguration wie Ihre vorhandene Umgebung. Sie können dies auf eine der folgenden Arten ausführen.

   > **Note**
   >
   > Mit der Funktion „Saved Configurations (Gespeicherte Konfigurationen)“ können Sie Ihre Umgebungen im neuen Konto neu erstellen. Diese Funktion kann die Umgebungskonfiguration speichern, sodass Sie sie anwenden können, wenn Sie andere Umgebungen erstellen oder aktualisieren. Weitere Informationen finden Sie unter Verwenden gespeicherter Elastic Beanstalk-Konfigurationen.

   - Wenden Sie mithilfe der Elastic Beanstalk Beanstalk-Konsole eine gespeicherte Konfiguration aus Ihrer EC2 -Classic-Umgebung an, wenn Sie die neue Umgebung konfigurieren. Diese Konfiguration verwendet die VPC. Weitere Informationen finden Sie unter Verwenden gespeicherter Elastic Beanstalk-Konfigurationen.

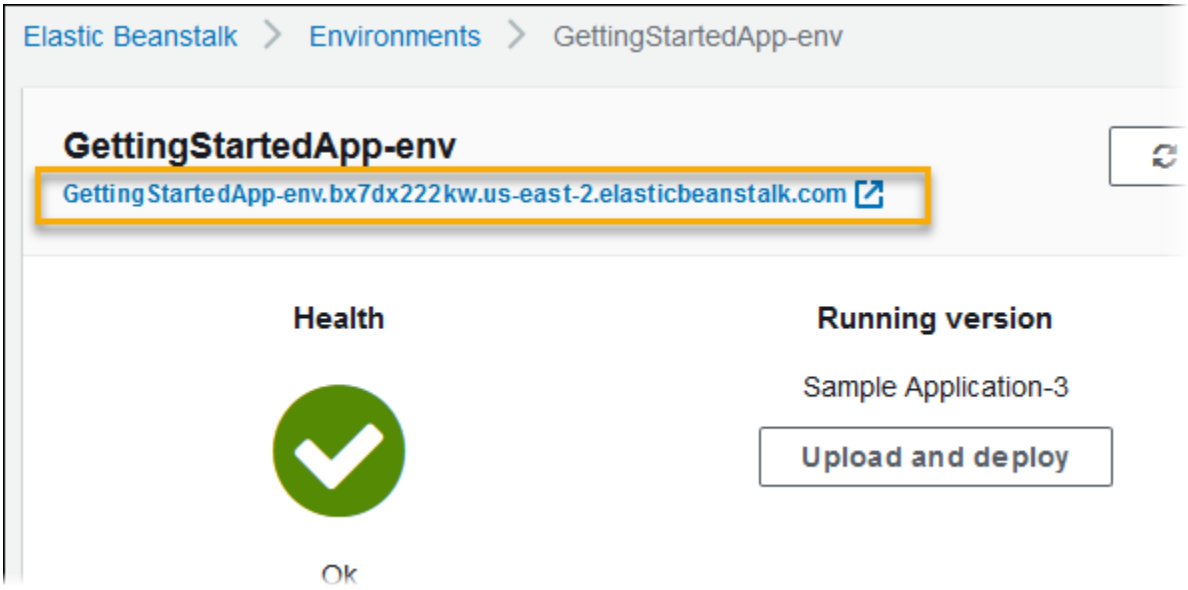
- Führen Sie über die Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) den Befehl eb create aus, um Ihre Umgebung neu zu erstellen. Geben Sie die Parameter aus der ursprünglichen Umgebung und den VPC-Bezeichner an. Eines der Beispiele in der eb create-Befehlsbeschreibung zeigt, wie Sie eine Umgebung in einer benutzerdefinierten VPC erstellen.
- Verwenden Sie AWS Command Line Interface (AWS CLI) und erstellen Sie Ihre Umgebung mit dem Befehl neu. elasticbeanstalk create-environment Geben Sie die Parameter der ursprünglichen Umgebung mit dem VPC-Bezeichner an. Detaillierte Anweisungen finden Sie unter Elastic Beanstalk Beanstalk-Umgebungen mit der CLI erstellen AWS.

2. Tauschen Sie die CNAMEs der vorhandenen Umgebung durch die neue Umgebung aus. Auf diese Weise kann die neue Umgebung, die Sie erstellt haben, mit der vertrauten Adresse referenziert werden. Sie können die EB CLI oder die AWS CLI verwenden.
   - Tauschen Sie mit der EB-CLI die Umgebung aus, CNAMEs indem eb swap Sie den Befehl ausführen. Weitere Informationen finden Sie unter Einrichtung der EB-Befehlszeilenschnittstelle (EB CLI) zur Verwaltung von Elastic Beanstalk.
   - Tauschen Sie CNAMEs mit dem Befehl elasticbeanstalk swap-environment-cnames die Umgebung aus. AWS CLI Weitere Informationen finden Sie in der AWS CLI - Befehlsreferenz.

Der Domainname Ihrer Elastic Beanstalk-Umgebung

Standardmäßig steht Ihre Umgebung Benutzern in einer Subdomain von `elasticbeanstalk.com` zur Verfügung. Wenn Sie eine Umgebung erstellen, können Sie einen Hostnamen für Ihre Anwendung auswählen. Die Subdomain und die Domain werden automatisch in *`region`*`.elasticbeanstalk.com` eingetragen.

Um Benutzer zu Ihrer Umgebung weiterzuleiten, registriert Elastic Beanstalk einen CNAME-Eintrag, der auf den Load Balancer Ihrer Umgebung verweist. Sie können die URL der Anwendung Ihrer Umgebung mit dem aktuellen Wert des CNAME auf der Umgebungsübersichtsseite der Elastic Beanstalk-Konsole anzeigen.



Wählen Sie die URL auf der Übersichtsseite oder wählen Sie im Navigationsbereich die Option Go to environment (Zur Umgebung wechseln), um zur Webseite Ihrer Anwendung zu navigieren.

Sie können den CNAME-Wert der Umgebung ändern, indem Sie diesen mit dem CNAME einer anderen Umgebung austauschen. Detaillierte Anweisungen finden Sie unter Blau/Grün-Bereitstellungen mit Elastic Beanstalk.

Wenn Sie einen Domainnamen besitzen, können Sie Amazon Route 53 verwenden, um diesen in Ihre Umgebung aufzulösen. Sie können einen Domainnamen mit Amazon Route 53 erwerben oder den Domainnamen verwenden, den Sie von einem anderen Anbieter gekauft haben.

Informationen zum Erwerb eines Domainnamens mit Route 53 finden Sie unter Registrieren einer neuen Domain im Entwicklerhandbuch von Amazon Route 53.

Weitere Informationen zur Verwendung einer benutzerdefinierten Domain finden Sie unter Weiterleiten des Datenverkehrs an eine AWS Elastic Beanstalk -Umgebung im Entwicklerhandbuch von Amazon Route 53.

> **Important**
>
> Wenn Sie eine Umgebung beenden, müssen Sie auch alle CNAME-Zuordnungen löschen, die Sie erstellt haben, damit andere Kunden einen verfügbaren Hostnamen wiederverwenden können. Achten Sie darauf, DNS-Einträge zu löschen, die auf Ihre beendete Umgebung verweisen, um zu verhindern, dass ein DNS-Eintrag hängenbleibt. Ein hängender DNS-Eintrag kann den Internetdatenverkehr, der für Ihre Domain bestimmt ist, Sicherheitslücken aussetzen. Er kann auch andere Risiken bergen.

Weitere Informationen finden Sie unter Schutz vor hängenden Delegierungsdatensätzen in Route 53 im Entwicklerhandbuch von Amazon Route 53. Weitere Informationen zu fehlernden DNS-Einträgen finden Sie auch im AWS Sicherheits-Blog unter Enhanced Domain Protections for Amazon CloudFront Requests.

Konfigurieren von Elastic Beanstalk-Umgebungen (Erweitert)

Wenn Sie eine AWS Elastic Beanstalk Umgebung erstellen, stellt Elastic Beanstalk alle AWS Ressourcen bereit und konfiguriert sie, die für die Ausführung und Unterstützung Ihrer Anwendung erforderlich sind. Neben den Metadaten und dem Aktualisierungsverhalten der Umgebung können Sie auch diese Ressourcen anpassen, indem Sie Werte für Konfigurationsoptionen angeben. Beispielsweise möchten Sie vielleicht eine Amazon SQS SQS-Warteschlange und einen Alarm zur Warteschlangentiefe hinzufügen, oder Sie möchten vielleicht einen ElastiCache Amazon-Cluster hinzufügen.

Die meisten Konfigurationsoptionen haben Standardwerte, die automatisch von Elastic Beanstalk übernommen werden. Sie können diese Standardwerte mit Konfigurationsdateien, gespeicherten Konfigurationen, Befehlszeilenoptionen oder mit dem direkten Aufruf der Elastic Beanstalk-API überschreiben. Die EB CLI und die Elastic Beanstalk-Konsole übernehmen ebenfalls empfohlene Werte für einige Optionen.

Indem Sie Ihrem Quell-Bundle eine Konfigurationsdatei beifügen, können Sie die Umgebung einfach gleichzeitig mit der Bereitstellung der Anwendungsversion anpassen. Beim Anpassen der Software auf Ihrer Instance ist es vorteilhafter, eine Konfigurationsdatei zu verwenden, als ein benutzerdefiniertes AMI zu erstellen, da Sie keinen Satz von AMIs verwalten müssen.

Möglicherweise muss die Software, von der Ihre Anwendung abhängig ist, im Rahmen der Anwendungsbereitstellung angepasst und konfiguriert werden. Bei diesen Dateien kann es sich entweder um von der Anwendung benötigte Abhängigkeiten (z. B. zusätzliche Pakete aus dem "yum"-Repository) oder um Konfigurationsdateien handeln (wie z. B. ein Ersatz der Datei httpd.conf zum Überschreiben bestimmter Standardeinstellungen von AWS Elastic Beanstalk).

Themen

- Konfigurationsoptionen
- Erweiterte Umgebungsanpassung mit Konfigurationsdateien (.ebextensions)
- Verwenden gespeicherter Elastic Beanstalk-Konfigurationen
- Umgebungsmanifest (env.yaml)
- Verwenden eines benutzerdefinierten Amazon Machine Images (AMI) in Ihrer Elastic Beanstalk Beanstalk-Umgebung
- Bereitstellen von statischen Dateien
- Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung

Konfigurationsoptionen

Elastic Beanstalk definiert eine große Anzahl von Konfigurationsoptionen, mithilfe derer Sie das Verhalten Ihrer Umgebung und die darin enthaltenen Ressourcen konfigurieren können. Die Konfigurationsoptionen sind in Namespaces wie `aws:autoscaling:asg` unterteilt, die Optionen für die Auto-Scaling-Gruppe einer Umgebung definieren.

Die Elastic-Beanstalk-Konsole und die EB-CLI legen beim Erstellen einer Umgebung Konfigurationsoptionen fest, einschließlich von Ihnen explizit festgelegter Optionen und empfohlener Werte, die vom Client definiert werden. Sie können Konfigurationsoptionen auch in gespeicherten Konfigurationen und Konfigurationsdateien festlegen. Wenn dieselbe Option an mehreren Orten festgelegt wird, wird der zu verwendende Wert durch die Rangfolge bestimmt.

Konfigurationsoptionseinstellungen können im Textformat definiert und vor der Umgebungserstellung gespeichert, während der Umgebungserstellung unter Verwendung eines beliebigen unterstützten Clients übernommen und nach der erfolgten Umgebungserstellung hinzugefügt, geändert oder entfernt werden. Eine detaillierte Aufstellung aller verfügbaren Methoden für die Verwendung von Konfigurationsoptionen in jeder der drei Phasen finden Sie in den folgenden Themen:

- Festlegen von Konfigurationsoptionen vor der Umgebungserstellung
- Festlegen von Konfigurationsoptionen während der Umgebungserstellung
- Festlegen von Konfigurationsoptionen nach der Umgebungserstellung

Eine vollständige Liste der Namespaces und Optionen einschließlich der Standard- und unterstützten Werte finden Sie unter Allgemeine Optionen für alle Umgebungen und Plattformspezifische Optionen.

Precedence

Während der Umgebungserstellung werden die Konfigurationsoptionen mit folgender Rangfolge (von der höchsten bis zur niedrigsten) aus mehreren Quellen übernommen:

- Direkt auf die Umgebung angewendete Einstellungen — Einstellungen, die während eines Vorgangs zum Erstellen einer Umgebung oder zum Aktualisieren der Umgebung auf der Elastic Beanstalk Beanstalk-API von einem beliebigen Client angegeben wurden, einschließlich der Elastic Beanstalk Beanstalk-Konsole, EB CLI und. AWS CLI SDKs Elastic-Beanstalk-Konsole und EB-CLI weisen außerdem einigen Optionen für dieses Level empfohlene Werte zu, sofern diese nicht überschrieben werden.

- Gespeicherte Konfigurationen – Optionseinstellungen, die nicht direkt in die Umgebung übernommen, sondern aus einer gespeicherten Konfiguration geladen werden (sofern vorhanden).
- Konfigurationsdateien (.ebextensions) – Optionseinstellungen, die weder direkt in die Umgebung übernommen noch in einer gespeicherten Konfiguration angegeben, sondern aus Konfigurationsdateien des Ordners `.ebextensions` im Stamm des Quell-Bundles der Anwendung geladen werden.

  Konfigurationsdateien werden in alphabetischer Reihenfolge ausgeführt. So wird beispielsweise `.ebextensions/01run.config` vor `.ebextensions/02do.config` ausgeführt.
- Standardwerte – Wenn für eine Konfigurationsoption ein Standardwert vorhanden ist, wird dieser nur übernommen, sofern diese Option auf keiner der oben genannten Ebenen festgelegt wurde.

Falls dieselbe Konfigurationsoption an mehreren Orten definiert ist, gilt die Einstellung mit der höchsten Priorität. Eine Einstellung wird als Teil der Umgebungskonfiguration gespeichert, sofern sie aus einer gespeicherten Konfiguration geladen oder direkt in der Umgebung bereitgestellt wird. Diese Einstellungen können mit der AWS CLI oder mit der EB-CLI entfernt werden.

Einstellungen in Konfigurationsdateien werden nicht direkt in die Umgebung übernommen und können nicht ohne Änderung der Konfigurationsdateien und Bereitstellung einer neuen Anwendungsversion entfernt werden. Wenn Sie eine Einstellung, die mit einer der anderen Methoden übernommen wurde, entfernen, wird dieselbe Einstellung aus den Konfigurationsdateien im Quell-Bundle geladen.

Beispiel: Sie legen im Rahmen der Umgebungserstellung mit der Elastic-Beanstalk-Konsole, einer Befehlszeilenoption oder einer gespeicherten Konfiguration die Mindestanzahl der Instances in der Umgebung auf 5 fest. Im Quell-Bundle der Anwendung ist ebenfalls eine Konfigurationsdatei enthalten, in der die Mindestanzahl der Instances mit 2 angegeben wird.

Wenn Sie die Umgebung erstellen, setzt Elastic Beanstalk die Option `MinSize` im Namespace `aws:autoscaling:asg` auf 5. Wenn Sie anschließend die Option aus der Umgebungskonfiguration entfernen, wird der Wert aus der Konfigurationsdatei geladen, sodass die Mindestanzahl der Instances auf 2 festgelegt wird. Falls Sie dann die Konfigurationsdatei aus dem Quell-Bundle entfernen und eine neue Bereitstellung ausführen, verwendet Elastic Beanstalk den Standardwert 1.

Empfohlene Werte

Die Elastic-Beanstalk-Befehlszeilenschnittstelle (EB-CLI) und die Elastic-Beanstalk-Konsole stellen für einige Konfigurationsoptionen empfohlene Werte bereit. Diese Werte können

von den Standardwerten abweichen und werden im Rahmen der Umgebungserstellung auf API-Ebene festgelegt. Mithilfe der empfohlenen Werte kann Elastic Beanstalk die Standardumgebungskonfiguration optimieren, ohne abwärts inkompatible Änderungen an der API vorzunehmen.

Beispielsweise legen sowohl die EB-CLI als auch die Elastic Beanstalk Beanstalk-Konsole die Konfigurationsoption für den EC2 Instance-Typ (`InstanceType`im `aws:autoscaling:launchconfiguration` Namespace) fest. Jeder Client bietet eine andere Möglichkeit zur Überschreibung der Standardeinstellung. In der Konsole können Sie über ein Dropdown-Menü auf der Seite Configuration Details (Konfigurationsdetails) im Assistenten Create New Environment (Neue Umgebung erstellen) einen anderen Instance-Typ auswählen. In der EB-CLI verwenden Sie den Parameter `--instance_type` für eb create.

Da die empfohlenen Werte auf API-Ebene festgelegt werden, überschreiben diese die Werte, die Sie für dieselben Optionen in Konfigurationsdateien oder gespeicherten Konfigurationen definiert haben. Folgende Optionen werden festgelegt:

Elastic-Beanstalk-Konsole

- Namespace: `aws:autoscaling:launchconfiguration`

  Optionsnamen: `IamInstanceProfile`, `EC2KeyName`, `InstanceType`
- Namespace: `aws:autoscaling:updatepolicy:rollingupdate`

  Optionsnamen: `RollingUpdateType` und `RollingUpdateEnabled`
- Namespace: `aws:elasticbeanstalk:application`

  Optionsname: `Application Healthcheck URL`
- Namespace: `aws:elasticbeanstalk:command`

  Optionsnamen: `DeploymentPolicy`, `BatchSize` und `BatchSizeType`
- Namespace: `aws:elasticbeanstalk:environment`

  Optionsname: `ServiceRole`
- Namespace: `aws:elasticbeanstalk:healthreporting:system`

  Optionsnamen: `SystemType` und `HealthCheckSuccessThreshold`
- Namespace: `aws:elasticbeanstalk:sns:topics`

Optionsname: `Notification Endpoint`

- Namespace: `aws:elasticbeanstalk:sqsd`

  Optionsname: `HttpConnections`

- Namespace: `aws:elb:loadbalancer`

  Optionsname: `CrossZone`

- Namespace: `aws:elb:policies`

  Optionsnamen: `ConnectionDrainingTimeout` und `ConnectionDrainingEnabled`

EB-CLI

- Namespace: `aws:autoscaling:launchconfiguration`

  Optionsnamen: `IamInstanceProfile`, `InstanceType`

- Namespace: `aws:autoscaling:updatepolicy:rollingupdate`

  Optionsnamen: `RollingUpdateType` und `RollingUpdateEnabled`

- Namespace: `aws:elasticbeanstalk:command`

  Optionsnamen: `BatchSize` und `BatchSizeType`

- Namespace: `aws:elasticbeanstalk:environment`

  Optionsname: `ServiceRole`

- Namespace: `aws:elasticbeanstalk:healthreporting:system`

  Optionsname: `SystemType`

- Namespace: `aws:elb:loadbalancer`

  Optionsname: `CrossZone`

- Namespace: `aws:elb:policies`

  Optionsnamen: `ConnectionDrainingEnabled`

Festlegen von Konfigurationsoptionen vor der Umgebungserstellung

AWS Elastic Beanstalk unterstützt eine Vielzahl von Konfigurationsoptionen, mit denen Sie die Einstellungen ändern können, die auf Ressourcen in Ihrer Umgebung angewendet werden. Viele dieser Optionen haben Standardwerte, die Sie überschreiben können, um Ihre Umgebung anzupassen. Andere Optionen lassen sich für die Aktivierung zusätzlicher Funktionen konfigurieren.

Elastic Beanstalk unterstützt zwei Methoden, um die Einstellungen der Konfigurationsoptionen zu speichern. Konfigurationsdateien im YAML- oder JSON-Format können im Quellcode der Anwendung in ein Verzeichnis mit dem Namen `.ebextensions` eingebunden und als Teil des Quell-Bundles der Anwendung bereitgestellt werden. Die Konfigurationsdateien lassen sich lokal erstellen und verwalten.

Gespeicherte Konfigurationen sind Vorlagen, die Sie aus einer ausgeführten Umgebung oder einer JSON-Optionsdatei erstellen und in Elastic Beanstalk speichern. Vorhandene gespeicherte Konfigurationen können auch erweitert werden, um eine neue Konfiguration zu erstellen.

> **Note**
>
> Die in Konfigurationsdateien und gespeicherten Konfigurationen definierten Einstellungen haben eine geringere Priorität als Einstellungen, die während oder nach der Umgebungserstellung konfiguriert wurden, einschließlich der empfohlenen Werte, die von der Elastic-Beanstalk-Konsole und der EB-CLI zugewiesen werden. Details dazu finden Sie unter Precedence.

Zudem können Optionen auch in Form eines JSON-Dokuments angegeben und im Rahmen einer Umgebungserstellung oder -aktualisierung mit der EB-CLI oder der direkt in Elastic Beanstalk bereitgestellt werde AWS CLI. Werden Optionen auf diese Weise direkt in Elastic Beanstalk bereitgestellt, setzt dies alle anderen Methoden außer Kraft.

Eine vollständige Liste der verfügbaren Optionen finden Sie unter Konfigurationsoptionen.

Methoden

- Konfigurationsdateien (.ebextensions)
- Gespeicherte Konfigurationen
- JSON-Dokument

- EB-CLI-Konfiguration

Konfigurationsdateien (`.ebextensions`)

Konfigurieren Sie mit `.ebextensions` die Optionen, die für eine ordnungsgemäße Funktionsweise der Anwendung erforderlich sind, und stellen Sie Standardwerte für andere Optionen bereit, die mit einer höheren Priorität überschrieben werden können. Die in `.ebextensions` spezifizierten Optionen weisen die niedrigste Priorität auf und werden von Einstellungen mit jeder anderen Priorität außer Kraft gesetzt.

Zur Verwendung von Konfigurationsdateien erstellen Sie einen Ordner mit dem Namen `.ebextensions` auf der obersten Ebene des Projektquellcodes. Fügen Sie eine Datei mit der Erweiterung `.config` hinzu und geben Sie die Optionen wie folgt an:

```
option_settings:
  - namespace:  namespace
    option_name:  option name
    value:  option value
  - namespace:  namespace
    option_name:  option name
    value:  option value
```

Beispielsweise wird mit der folgenden Konfigurationsdatei die Zustandsprüfungs-URL für die Anwendung auf festgeleg `/health`:

`healthcheckurl.config`

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application
    option_name:  Application Healthcheck URL
    value:  /health
```

Im JSON-Format:

```
{
 "option_settings" :
    [
      {
        "namespace" : "aws:elasticbeanstalk:application",
```

```
        "option_name" : "Application Healthcheck URL",
        "value" : "/health"
      }
    ]
}
```

Damit wird der Elastic-Load-Balancing-Load-Balancer in der Elastic-Beanstalk-Umgebung so konfiguriert, dass eine HTTP-Anforderung über den Pfad `/health` an die einzelnen EC2-Instances gesendet wird, um deren Zustand zu bestimmen.

> **Note**
>
> Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Binden Sie das Verzeichnis `.ebextensions` in das Quell-Bundle der Anwendung ein und stellen Sie es in einer neuen oder vorhandenen Elastic-Beanstalk-Umgebung bereit.

Konfigurationsdateien unterstützen neben `option_settings` noch weitere Abschnitte für die Anpassung der Software und Dateien, die auf den Servern in der Umgebung ausgeführt werden. Weitere Informationen finden Sie unter .Ebextensions.

Gespeicherte Konfigurationen

Erstellen Sie eine gespeicherte Konfiguration, um Einstellungen zu speichern, die Sie während oder nach einer Umgebungserstellung mit der Elastic-Beanstalk-Konsole, der EB-CLI oder der einer vorhandenen Umgebung zugewiesenen habe AWS CLI. Gespeicherte Konfigurationen gehören zu einer Anwendung und können auf neue oder bestehende Umgebungen für die Anwendung angewendet werden.

Clients

- Elastic-Beanstalk-Konsole
- EB-CLI
- AWS CLI

Elastic-Beanstalk-Konsole

So erstellen Sie eine gespeicherte Konfiguration (Elastic-Beanstalk-Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie Actions (Aktionen) und anschließend Save Configuration (Konfiguration speichern) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Aktion abzuschließen.

Gespeicherte Konfigurationen werden im S3-Bucket von Elastic Beanstalk in einem Ordner gespeichert, der nach der Anwendung benannt ist. Beispielsweise befinden sich Konfigurationen für die Anwendung `my-app` in der Region us-west-2 für die Kontonummer 123456789012 unter `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app`.

EB-CLI

Die EB-CLI bietet unter eb config auch Unterbefehle für die Interaktion mit gespeicherten Konfigurationen:

So erstellen Sie eine gespeicherte Konfiguration (EB-CLI)

1. Speichern Sie die zugeordnete aktuelle Konfiguration der Umgebung:

   ```
   ~/project$ eb config save --cfg my-app-v1
   ```

   In der EB-CLI wird die Konfiguration unter gespeichert `~/project/.elasticbeanstalk/saved_configs/my-app-v1.cfg.yml`
2. Nehmen Sie bei Bedarf lokale Änderungen an der gespeicherten Konfiguration vor.
3. Laden Sie die gespeicherte Konfiguration in S3 hoch:

   ```
   ~/project$ eb config put my-app-v1
   ```

AWS CLI

Erstellen Sie mit eine gespeicherte Konfiguration aus einer ausgeführten Umgebung `aws elasticbeanstalk create-configuration-template`

So erstellen Sie eine gespeicherte Konfiguration (AWS CLI)

1. Ermitteln Sie die Umgebungs-ID der Elastic-Beanstalk-Umgebung mit `describe-environments`:

```
$ aws elasticbeanstalk describe-environments --environment-name my-env
{
    "Environments": [
        {
            "ApplicationName": "my-env",
            "EnvironmentName": "my-env",
            "VersionLabel": "89df",
            "Status": "Ready",
            "Description": "Environment created from the EB CLI using \"eb create
\"",
            "EnvironmentId": "e-vcghmm2zwk",
            "EndpointURL": "awseb-e-v-AWSEBLoa-1JUM8159RA11M-43V6ZI1194.us-
west-2.elb.amazonaws.com",
            "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.2 running Multi-
container Docker 1.7.1 (Generic)",
            "CNAME": "my-env-nfptuqaper.elasticbeanstalk.com",
            "Health": "Green",
            "AbortableOperationInProgress": false,
            "Tier": {
                "Version": " ",
                "Type": "Standard",
                "Name": "WebServer"
            },
            "HealthStatus": "Ok",
            "DateUpdated": "2015-10-01T00:24:04.045Z",
            "DateCreated": "2015-09-30T23:27:55.768Z"
        }
    ]
}
```

2. Speichern Sie die aktuelle Konfiguration der Umgebung mit `create-configuration-template`:

```
$ aws elasticbeanstalk create-configuration-template --environment-id e-vcghmm2zwk
 --application-name my-app --template-name v1
```

Elastic Beanstalk speichert die Konfiguration in Ihrem Elastic-Beanstalk-Bucket in Amazon-S3.

JSON-Dokument

Wenn Sie die AWS CLI zum Erstellen und Aktualisieren von Umgebungen verwenden, können Sie auch Konfigurationsoptionen im JSON-Format bereitstellen. Eine Bibliothek mit Konfigurationsdateien in JSON ist nützlich, wenn Sie die AWS CLI zum Erstellen und Verwalten von Umgebungen verwenden.

Beispielsweise wird mit folgendem JSON-Dokument die Zustandsprüfungs-URL für die Anwendung auf festgeleg `/health`:

~/ebconfigs/healthcheckurl.json

```
[
  {
    "Namespace": "aws:elasticbeanstalk:application",
    "OptionName": "Application Healthcheck URL",
    "Value": "/health"
  }
]
```

EB-CLI-Konfiguration

Die EB-CLI unterstützt nicht nur gespeicherte Konfigurationen sowie die direkte Umgebungskonfiguration mit eb config-Befehlen, sondern verfügt auch über eine Konfigurationsdatei. Mithilfe der darin enthaltenen Option `default_ec2_keyname` können Sie ein Amazon-EC2-Schlüsselpaar für den SSH-Zugriff auf Instances in der Umgebung angeben. In der EB-CLI wird diese Option verwendet, um die Konfigurationsoption `EC2KeyName` im Namespace `aws:autoscaling:launchconfiguration` festzulegen.

~/ .yml workspace/my-app/.elasticbeanstalk/config

```
branch-defaults:
  master:
    environment: my-env
```

```
  develop:
    environment: my-env-dev
deploy:
  artifact: ROOT.war
global:
  application_name: my-app
  default_ec2_keyname: my-keypair
  default_platform: Tomcat 8 Java 8
  default_region: us-west-2
  profile: null
  sc: git
```

Festlegen von Konfigurationsoptionen während der Umgebungserstellung

Wenn Sie eine AWS Elastic Beanstalk Umgebung mithilfe der Elastic Beanstalk Beanstalk-Konsole, der EB CLI AWS CLI, eines SDK oder der Elastic Beanstalk Beanstalk-API erstellen, können Sie Werte für Konfigurationsoptionen angeben, um Ihre Umgebung und die darin gestarteten AWS Ressourcen anzupassen.

Konfigurationsänderungen, die nicht einmalig sind, können Sie in Konfigurationsdateien speichern, und zwar lokal, im Quell-Bundle oder in Amazon-S3.

In diesem Thema werden die Verfahren aller Methoden zum Festlegen von Konfigurationsoptionen während der Umgebungserstellung dargelegt.

Clients

- In der Elastic-Beanstalk-Konsole
- Verwenden der EB-CLI
- Verwenden Sie den AWS CLI

In der Elastic-Beanstalk-Konsole

Wenn Sie eine Elastic-Beanstalk-Umgebung in der Elastic-Beanstalk-Konsole erstellen, können Sie Konfigurationsoptionen mit Konfigurationsdateien, gespeicherten Konfigurationen und Formularen im Assistenten Create New Environment (Neue Umgebung erstellen) bereitstellen.

Methoden

- Verwenden von Konfigurationsdateien (.ebextensions)
- Verwenden einer gespeicherten Konfiguration

- Verwenden des Assistenten für die neue Umgebung

Verwenden von Konfigurationsdateien (**`.ebextensions`**)

Binden Sie `.config`-Dateien in den Ordner `.ebextensions` im Quell-Bundle der Anwendung ein.

Weitere Informationen zu Konfigurationsdateien finden Sie unter .Ebextensions.

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Laden Sie das Quell-Bundle wie üblich während der Umgebungserstellung in Elastic Beanstalk hoch.

Die Elastic-Beanstalk-Konsole weist einigen Konfigurationsoptionen empfohlene Werte zu und stellt für andere Optionen Formularfelder bereit. Über die Elastic-Beanstalk-Konsole konfigurierte Optionen werden direkt in der Umgebung bereitgestellt und überschreiben die Einstellungen in den Konfigurationsdateien.

Verwenden einer gespeicherten Konfiguration

Beim Erstellen einer neuen Umgebung mit der Elastic-Beanstalk-Konsole besteht einer der ersten Schritte darin, eine Konfiguration auszuwählen. Bei der Konfiguration kann es sich um eine predefined configuration (vordefinierte Konfiguration) (in der Regel die neueste Version einer Plattform wie PHP oder Tomcat) oder um eine saved configuration (gespeicherte Konfiguration) handeln.

So wenden Sie eine gespeicherte Konfiguration während der Umgebungserstellung an (Elastic-Beanstalk-Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Saved configurations (Gespeicherte Konfigurationen) aus.

4. Wählen Sie die gespeicherte Konfiguration aus, die Sie anwenden möchten, und wählen Sie dann Launch environment (Umgebung starten).
5. Führen Sie die Schritte im Assistenten aus, um die Umgebung zu erstellen.

Gespeicherte Konfigurationen sind anwendungsspezifisch. Weitere Informationen zum Erstellen von gespeicherten Konfigurationen finden Sie unter Gespeicherte Konfigurationen.

Verwenden des Assistenten für die neue Umgebung

Die meisten standardmäßigen Konfigurationsoptionen finden Sie auf der Seite Configure more options (Weitere Optionen konfigurieren) des Create New Environment-Assistenten. Wenn Sie eine Amazon-RDS-Datenbank erstellen oder eine VPC für die Umgebung konfigurieren, stehen für diese Ressourcen zusätzliche Konfigurationsoptionen zur Verfügung.

So legen Sie Konfigurationsoptionen während der Umgebungserstellung fest (Elastic-Beanstalk-Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Applications (Anwendungen).
3. Wählen Sie eine Anwendung aus oder erstellen Sie eine.
4. Wählen Sie Actions (Aktionen) und dann Create Environment (Umgebung erstellen).
5. Fahren Sie mithilfe des Assistenten fort und wählen Sie Configure more options (Weitere Optionen konfigurieren) aus.
6. Wählen Sie in einer oder mehreren Konfigurationskategorien die configuration presets (Konfigurationsvoreinstellungen) und dann Edit (Bearbeiten), um eine Gruppe verwandter Konfigurationsoptionen zu ändern.
7. Wählen Sie nach der Optionsauswahl Create environment (Umgebung erstellen) aus.

Alle Optionen, die Sie im Assistenten für die neue Umgebung festgelegt haben, werden direkt in die Umgebung übernommen und überschreiben alle Optionseinstellungen aus den von Ihnen angewendeten gespeicherten Konfigurationen oder Konfigurationsdateien (`.ebextensions`). Sie können Einstellungen entfernen, nachdem die Umgebung mithilfe der EB-CLI erstellt wurde, oder AWS CLI um die Einstellungen in gespeicherten Konfigurationen oder Konfigurationsdateien anzuzeigen.

Weitere Informationen über den neuen Umgebungsassistenten finden Sie unter Der Assistent zum Erstellen einer neuen Umgebung.

Verwenden der EB-CLI

Methoden

- Verwenden von Konfigurationsdateien (.ebextensions)
- Verwenden von gespeicherten Konfigurationen
- Verwenden von Befehlszeilenoptionen

Verwenden von Konfigurationsdateien (`.ebextensions`)

Binden Sie `.config`-Dateien unter `.ebextensions` in den Projektordner ein, um sie mit dem Anwendungscode bereitzustellen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter .Ebextensions.

```
~/workspace/my-app/
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- .elasticbeanstalk
|   `-- config.yml
|-- index.php
`-- styles.css
```

Erstellen Sie die Umgebung und stellen Sie den entsprechenden Quellcode mit berei eb create.

```
~/workspace/my-app$ eb create my-env
```

Verwenden von gespeicherten Konfigurationen

Um eine gespeicherte Konfiguration bei der Umgebungserstellung mit eb create anzuwenden, verwenden Sie die Option `--cfg`.

```
~/workspace/my-app$ eb create --cfg savedconfig
```

Sie können die gespeicherte Konfiguration in Ihrem Projektordner oder in Ihrem Elastic-Beanstalk-Speicherort auf Amazon-S3 speichern. In vorherigem Beispiel sucht die EB-CLI zunächst nach

einer gespeicherten Konfigurationsdatei mit dem Namen `savedconfig.cfg.yml` im Ordner `.elasticbeanstalk/saved_configs/`. Geben Sie keine Dateinamenerweiterungen (`.cfg.yml`) an, wenn Sie eine gespeicherte Konfiguration mit `--cfg` übernehmen.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- healthcheckurl.config
|-- .elasticbeanstalk
|   |-- saved_configs
|   |   `-- savedconfig.cfg.yml
|   `-- config.yml
|-- index.php
`-- styles.css
```

Wenn die EB-CLI die Konfiguration nicht lokal findet, sucht sie im Elastic Beanstalk-Speicherort in Amazon-S3. Weitere Informationen zum Erstellen, Bearbeiten und Hochladen gespeicherter Konfigurationen finden Sie unter Gespeicherte Konfigurationen.

Verwenden von Befehlszeilenoptionen

Der eb create-Befehl der EB-CLI bietet mehrere Optionen, mit denen sich während der Umgebungserstellung Konfigurationsoptionen festlegen lassen. Mithilfe dieser Optionen können Sie eine RDS-Datenbank zur Umgebung hinzufügen, eine VPC konfigurieren oder empfohlene Werte überschreiben.

Beispielsweise verwendet die EB-CLI standardmäßig den Instance-Typ `t2.micro`. Mit der Option `--instance_type` können Sie einen anderen Instance-Typ auswählen.

```
$ eb create my-env --instance_type t2.medium
```

Verwenden Sie die `--database`-Optionen, um eine Amazon-RDS-DB-Instance zu erstellen und mit der Umgebung zu verknüpfen.

```
$ eb create --database.engine postgres --database.username dbuser
```

Falls Sie den Umgebungsnamen, das Datenbankpasswort oder andere Parameter nicht angeben, die zur Umgebungserstellung benötigt werden, werden Sie von der EB-CLI zur Eingabe aufgefordert.

Eine vollständige Liste der verfügbaren Optionen sowie Anwendungsbeispiele finden Sie unter eb create.

Verwenden Sie den AWS CLI

Wenn Sie den `create-environment` Befehl verwenden, um eine Elastic Beanstalk Beanstalk-Umgebung mit dem zu erstellen AWS CLI, wendet der AWS CLI keine empfohlenen Werte an. Alle Konfigurationsoptionen werden in den Konfigurationsdateien im von Ihnen vorgegebenen Quell-Bundle definiert.

Methoden

- Verwenden von Konfigurationsdateien (.ebextensions)
- Verwenden einer gespeicherten Konfiguration
- Verwenden von Befehlszeilenoptionen

Verwenden von Konfigurationsdateien (**`.ebextensions`**)

Um Konfigurationsdateien auf eine Umgebung anzuwenden, die Sie mit dem erstellen AWS CLI, nehmen Sie sie in das Anwendungsquellpaket auf, das Sie auf Amazon S3 hochladen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter .Ebextensions.

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Um ein Anwendungsquellpaket hochzuladen und eine Umgebung mit dem zu erstellen AWS CLI

1. Wenn Sie noch keinen Elastic-Beanstalk-Bucket in Amazon S3 haben, erstellen Sie einen mit `create-storage-location`.

   ```
   $ aws elasticbeanstalk create-storage-location
   {
       "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
   }
   ```

2. Laden Sie das Quell-Bundle der Anwendung in Amazon S3 hoch.

```
$ aws s3 cp sourcebundle.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/
sourcebundle.zip
```

3. Erstellen Sie die Anwendungsversion.

```
$ aws elasticbeanstalk create-application-version --application-name my-app --
version-label v1 --description MyAppv1 --source-bundle S3Bucket="elasticbeanstalk-
us-west-2-123456789012",S3Key="my-app/sourcebundle.zip" --auto-create-application
```

4. Erstellen Sie die Umgebung.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-
name my-env --version-label v1 --solution-stack-name "64bit Amazon Linux 2015.03
 v2.0.0 running Tomcat 8 Java 8"
```

Verwenden einer gespeicherten Konfiguration

Um eine gespeicherte Konfiguration während der Erstellung für eine Umgebung zu verwenden, nutzen Sie den Parameter `--template-name`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name
 my-env --template-name savedconfig --version-label v1
```

Wenn Sie eine gespeicherte Konfiguration angeben, müssen Sie keinen Lösungs-Stack-Namen spezifizieren. In den gespeicherten Konfigurationen ist bereits ein Lösungs-Stack definiert, sodass Elastic Beanstalk eine Fehlermeldung ausgibt, wenn Sie beide Optionen angeben.

Verwenden von Befehlszeilenoptionen

Mit dem Parameter `--option-settings` können Sie Konfigurationsoptionen im JSON-Format angeben.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name
 my-env --version-label v1 --template-name savedconfig --option-settings '[
  {
    "Namespace": "aws:elasticbeanstalk:application",
    "OptionName": "Application Healthcheck URL",
    "Value": "/health"
  }
```

```
]
```

Laden Sie JSON mit dem Präfix `file://` aus einer Datei.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-
name my-env --version-label v1 --template-name savedconfig --option-settings file://
healthcheckurl.json
```

Elastic Beanstalk übernimmt Optionseinstellungen, die Sie mit der Option `--option-settings` spezifiziert haben, direkt in die Umgebung. Falls dieselben Optionen in einer gespeicherten Konfiguration oder Konfigurationsdatei angegeben sind, werden die Werte von `--option-settings` überschrieben.

Festlegen von Konfigurationsoptionen nach der Umgebungserstellung

Sie können die Optionseinstellungen in einer laufenden Umgebung ändern, indem Sie gespeicherte Konfigurationen anwenden, ein neues Quell-Bundle mit Konfigurationsdateien (`.ebextensions`) hochladen oder ein JSON-Dokument nutzen. Die EB-CLI und die Elastic-Beanstalk-Konsole enthalten auch Client-spezifische Funktionen für das Einrichten und Aktualisieren von Konfigurationsoptionen.

Wenn Sie eine Konfigurationsoption festlegen oder ändern, können Sie ein vollständiges Umgebungs-Update auslösen, je nach Schweregrad der Änderung. Beispiel: Änderungen an Optionen in `aws:autoscaling:launchconfiguration`, z. B. `InstanceType`, verlangen, dass die Amazon-EC2-Instances in Ihrer Umgebung neu bereitgestellt werden müssen. Dadurch wird ein fortlaufendes Update ausgelöst. Andere Konfigurationsänderungen können ohne Unterbrechung oder erneute Bereitstellung angewendet werden.

Sie können Optionseinstellungen mit der EB-CLI oder AWS CLI Befehlen aus einer Umgebung entfernen. Wenn Sie eine Option entfernen, die direkt auf einer Umgebung auf einer API-Ebene festgelegt wurde, können Einstellungen in Konfigurationsdateien, die anderenfalls durch Einstellungen verborgen sind, die direkt auf eine Umgebung angewendet werden, eingeblendet und wirksam werden.

Einstellungen in gespeicherten Konfigurationen und Konfigurationsdateien können überschrieben werden, indem Sie diese Option mit einer der anderen Konfigurationsmethoden direkt in der Umgebung festlegen. Diese können jedoch nur vollständig entfernt werden, indem Sie eine aktualisierte gespeicherte Konfiguration oder eine Konfigurationsdatei anwenden. Wenn eine Option nicht in einer gespeicherten Konfiguration, Konfigurationsdatei oder direkt auf eine Umgebung

festgelegt ist, wird der Standardwert angewendet, sofern vorhanden. Details dazu finden Sie unter Precedence.

Clients

- Die Elastic-Beanstalk-Konsole.
- Die EB-CLI
- Das AWS CLI

Die Elastic-Beanstalk-Konsole.

Sie können die Einstellungen der Konfigurationsoptionen in der Elastic-Beanstalk-Konsole durch die Bereitstellung eines Anwendungsquell-Bundles mit Konfigurationsdateien aktivieren, eine gespeicherte Konfiguration zuweisen oder die Umgebung direkt mit der Seite Configuration (Konfiguration) in der Environment Management Console ändern.

Methoden

- Verwenden von Konfigurationsdateien (.ebextensions)
- Verwenden einer gespeicherten Konfiguration
- Elastic-Beanstalk-Konsole verwenden

Verwenden von Konfigurationsdateien (`.ebextensions`)

Aktualisieren Sie Konfigurationsdateien in Ihrem Quellverzeichnis, erstellen Sie ein neues Quell-Bundle und stellen Sie die neue Version in Ihrer Elastic-Beanstalk-Umgebung bereit, um die Änderungen zu übernehmen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter .Ebextensions.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.

5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Änderungen an Konfigurationsdateien überschreiben Optionseinstellungen in gespeicherten Konfigurationen oder Einstellungen, die direkt auf die Umgebung auf API-Ebene anwendet werden, nicht. Details dazu finden Sie unter Precedence.

Verwenden einer gespeicherten Konfiguration

Wenden Sie eine gespeicherte Konfiguration auf eine laufende Umgebung an, um Optionseinstellungen anzuwenden, die durch sie definiert werden.

So wenden Sie eine gespeicherte Konfiguration auf eine laufende Umgebung an (Elastic-Beanstalk-Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Saved configurations (Gespeicherte Konfigurationen) aus.
4. Wählen Sie die gespeicherte Konfiguration aus, die Sie anwenden möchten, und wählen Sie dann Load (Laden).
5. Wählen Sie eine Umgebung und anschließend Load (Laden) aus.

Einstellungen in einer gespeicherten Konfiguration überschreiben Einstellungen in Konfigurationsdateien und werden von Einstellungen überschrieben, die mit der Environment Management Console konfiguriert werden.

Weitere Informationen zum Erstellen von gespeicherten Konfigurationen finden Sie unter Gespeicherte Konfigurationen.

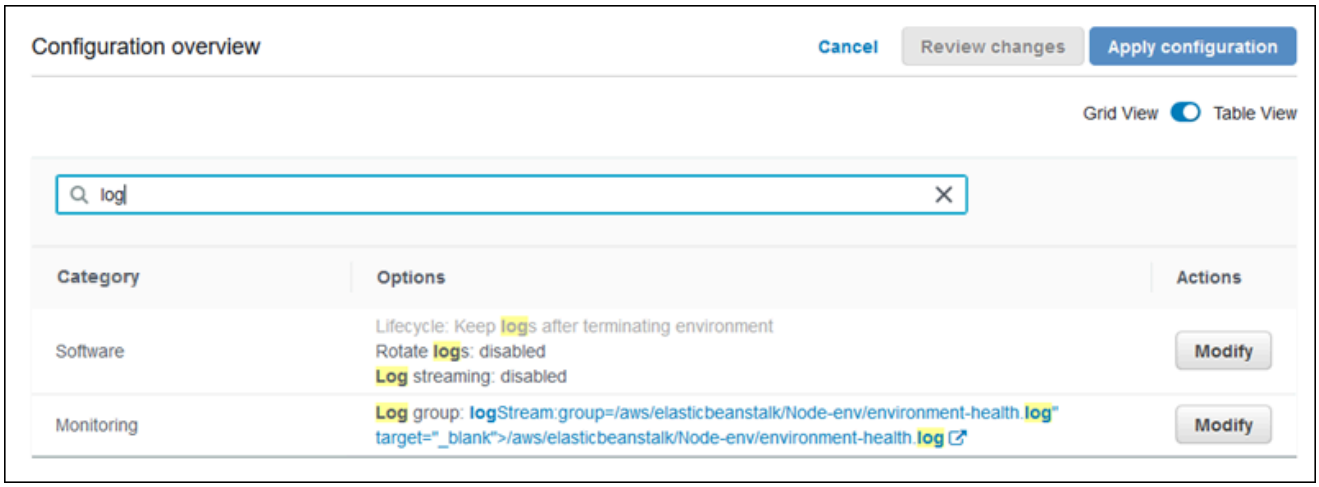
Elastic-Beanstalk-Konsole verwenden

Die Elastic-Beanstalk-Konsole stellt viele Konfigurationsoptionen für jede Umgebung auf der Seite Configuration (Konfiguration) bereit.

So ändern Sie die Konfigurationsoptionen in einer laufenden Umgebung (Elastic-Beanstalk-Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Suchen Sie die Konfigurationsseite, die Sie bearbeiten möchten:
   - Wenn Sie die Option sehen, an der Sie interessiert sind, oder Sie wissen, in welcher Konfigurationskategorie sie sich befindet, wählen Sie in der Konfigurationskategorie für sie die Option Edit (Bearbeiten).
   - Um eine Option zu suchen, aktivieren Sie Table View (Tabellenansicht) und geben Sie dann Suchbegriffe in das Suchfeld ein. Während der Eingabe wird die Liste auf die Optionen verkürzt, die mit Ihren Suchbegriffen übereinstimmen.

     Wenn Sie die Option sehen, nach der Sie suchen, wählen Sie in der Konfigurationskategorie, in der sie enthalten ist, die Option Edit (Bearbeiten).



5. Ändern Sie die Einstellungen und wählen Sie anschließend Save (Speichern) aus.
6. Wiederholen Sie bei Bedarf die vorherigen beiden Schritte in weiteren Konfigurationskategorien.
7. Wählen Sie Apply (Anwenden) aus.

Änderungen an den Konfigurationsoptionen in der Environment Management Console werden direkt auf die Umgebung angewendet. Diese Änderungen überschreiben die Einstellungen für dieselben Optionen in Konfigurationsdateien oder gespeicherten Konfigurationen. Details dazu finden Sie unter Precedence.

Weitere Informationen zum Ändern von Konfigurationsoptionen für eine ausgeführte Umgebung mit der Elastic-Beanstalk-Konsole finden Sie in den entsprechenden Themen unter Konfigurieren von Elastic-Beanstalk-Umgebungen.

Die EB-CLI

Sie können die Einstellungen der Konfigurationsoptionen mit der EB-CLI durch die Bereitstellung eines Quellcodes mit Konfigurationsdateien aktivieren, Einstellungen aus einer gespeicherten Konfiguration anwenden oder die Umgebungskonfiguration direkt mit dem eb config-Befehl ändern.

Methoden

- Verwenden von Konfigurationsdateien (.ebextensions)
- Verwenden einer gespeicherten Konfiguration
- benutze eb config
- benutze eb setenv

Verwenden von Konfigurationsdateien (**`.ebextensions`**)

Binden Sie `.config`-Dateien unter `.ebextensions` in den Projektordner ein, um sie mit dem Anwendungscode bereitzustellen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter .Ebextensions.

```
~/workspace/my-app/
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- .elasticbeanstalk
|   `-- config.yml
|-- index.php
`-- styles.css
```

Stellen Sie Ihren Quellcode mit bereit eb deploy.

```
~/workspace/my-app$ eb deploy
```

Verwenden einer gespeicherten Konfiguration

Um eine gespeicherte Konfiguration auf eine laufende Umgebung anzuwenden, nutzen Sie den Befehl eb config. Verwenden Sie die `--cfg`-Option mit dem Namen der gespeicherten Konfiguration, um die Einstellungen auf Ihre Umgebung anzuwenden.

```
$ eb config --cfg v1
```

In diesem Beispiel ist `v1` der Name einer zuvor erstellten und gespeicherten Konfigurationsdatei.

Einstellungen, die mit diesem Befehl auf eine Umgebung angewendet wurden, überschreiben Einstellungen, die während der Umgebungserstellung angewendet wurden, sowie Einstellungen, die in den Konfigurationsdateien in Ihrem Quell-Bundle der Anwendung definiert sind.

benutze eb config

Mit dem EB-CLI-eb config-Befehl können Sie Optionseinstellungen direkt auf einer Umgebung festlegen und entfernen, indem Sie einen Text-Editor verwenden.

Wenn Sie eb config ausführen, zeigt die EB-CLI Einstellungen in Ihrer Umgebung aus allen Quellen, einschließlich Konfigurationsdateien, gespeicherte Konfigurationen, empfohlene Werte, Optionen, die direkt auf der Umgebung festgelegt sind, und API-Standardeinstellungen.

> **Note**
>
> eb config zeigt keine Umgebungseigenschaften. Zum Festlegen von Umgebungseigenschaften, die Sie in Ihrer Anwendung lesen können, verwenden Sie eb setenv.

Das folgende Beispiel zeigt Einstellungen, die im `aws:autoscaling:launchconfiguration`-Namespace angewendet wurden. Diese Einstellungen umfassen Folgendes:

- Zwei empfohlene Werte für `IamInstanceProfile` und `InstanceType`, die von der EB-CLI während der Umgebungserstellung angewendet wurden.
- Die Option `EC2KeyName`, die während der Erstellung basierend auf der Repository-Konfiguration direkt auf der Umgebung festgelegt wurde.
- API-Standardwerte für andere Optionen.

```
ApplicationName: tomcat
DateUpdated: 2015-09-30 22:51:07+00:00
EnvironmentName: tomcat
SolutionStackName: 64bit Amazon Linux 2015.03 v2.0.1 running Tomcat 8 Java 8
settings:
...
aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: null
    EC2KeyName: my-key
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    ImageId: ami-1f316660
    InstanceType: t2.micro
...
```

Sie legen Sie Konfigurationsoptionen mit fest oder ändern Sie eb config

1. Führen Sie eb config aus, um die Umgebungskonfiguration anzuzeigen.

   ```
   ~/workspace/my-app/$ eb config
   ```

2. Ändern der Einstellungswerte mit dem Standard-Text-Editor.

   ```
   aws:autoscaling:launchconfiguration:
       BlockDeviceMappings: null
       EC2KeyName: my-key
       IamInstanceProfile: aws-elasticbeanstalk-ec2-role
       ImageId: ami-1f316660
       InstanceType: t2.medium
   ```

3. Speichern Sie die vorübergehende Konfigurationsdatei und beenden Sie sie.
4. Die EB-CLI aktualisiert Ihre Umgebungskonfiguration.

Das Einrichten der Konfigurationsoptionen mit eb config überschreibt Einstellungen aus allen anderen Quellen.

Sie können auch Optionen aus Ihrer Umgebung mit entfernen eb config.

So entfernen Sie Konfigurationsoptionen (EB-CLI)

1. Führen Sie eb config aus, um die Umgebungskonfiguration anzuzeigen.

```
~/workspace/my-app/$ eb config
```

2. Ersetzen Sie gezeigte Werte mit der Zeichenfolge `null`. Sie können auch die gesamte Zeile mit der Option, die Sie entfernen möchten, löschen.

```
aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: null
    EC2KeyName: my-key
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    ImageId: ami-1f316660
    InstanceType: null
```

3. Speichern Sie die vorübergehende Konfigurationsdatei und beenden Sie sie.
4. Die EB-CLI aktualisiert Ihre Umgebungskonfiguration.

Durch das Entfernen von Optionen aus Ihrer Umgebung mit eb config sind Einstellungen für dieselben Optionen möglich, die aus den Konfigurationsdateien im Quell-Bundle Ihrer Anwendung ermittelt werden. Details dazu finden Sie unter Precedence.

benutze eb setenv

Zum Festlegen von Umgebungseigenschaften mit der EB-CLI verwenden Sie eb setenv.

```
~/workspace/my-app/$ eb setenv ENVVAR=TEST
INFO: Environment update is starting.
INFO: Updating environment my-env's configuration settings.
INFO: Environment health has transitioned from Ok to Info. Command is executing on all
 instances.
INFO: Successfully deployed new configuration to environment.
```

Mit diesem Befehl werden Umgebungseigenschaften im `aws:elasticbeanstalk:application:environment`-Namespace festgelegt. Nach einer kurzen Aktualisierung stehen die mit eb setenv definierten Umgebungseigenschaften für die Anwendung zur Verfügung.

Zeigen Sie die Umgebungseigenschaften für Ihre Umgebung mit a eb printenv.

```
~/workspace/my-app/$ eb printenv
 Environment Variables:
```

```
    ENVVAR = TEST
```

Das AWS CLI

Sie können die Einstellungen der Konfigurationsoptionen mit dem aktualisieren, AWS CLI indem Sie ein Quellpaket bereitstellen, das Konfigurationsdateien enthält, eine remote gespeicherte Konfiguration anwenden oder die Umgebung direkt mit dem `aws elasticbeanstalk update-environment` Befehl ändern.

Methoden

- Verwenden von Konfigurationsdateien (.ebextensions)
- Verwenden einer gespeicherten Konfiguration
- Verwenden von Befehlszeilenoptionen

Verwenden von Konfigurationsdateien (**`.ebextensions`**)

Um Konfigurationsdateien auf eine laufende Umgebung mit dem anzuwenden AWS CLI, nehmen Sie sie in das Anwendungsquellpaket auf, das Sie auf Amazon S3 hochladen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter .Ebextensions.

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

So laden Sie das Quell-Bundle der Anwendung hoch und wenden es auf eine ausgeführte Umgebung an (AWS CLI)

1. Wenn Sie noch keinen Elastic-Beanstalk-Bucket in Amazon S3 haben, erstellen Sie einen mit `create-storage-location`:

   ```
   $ aws elasticbeanstalk create-storage-location
   {
       "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
   }
   ```

2. Laden Sie das Quell-Bundle der Anwendung in Amazon S3 hoch.

```
$ aws s3 cp sourcebundlev2.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/
sourcebundlev2.zip
```

3. Erstellen Sie die Anwendungsversion.

```
$ aws elasticbeanstalk create-application-version --application-
name my-app --version-label v2 --description MyAppv2 --source-bundle
 S3Bucket="elasticbeanstalk-us-west-2-123456789012",S3Key="my-app/
sourcebundlev2.zip"
```

4. Aktualisieren Sie die Umgebung.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --version-label
 v2
```

Verwenden einer gespeicherten Konfiguration

Sie können eine gespeicherte Konfiguration auf eine laufende Umgebung mit der `--template-name`-Option auf dem `aws elasticbeanstalk update-environment`-Befehl anwenden.

Die gespeicherte Konfiguration muss sich in Ihrem Elastic-Beanstalk-Bucket in einem Pfad mit dem Namen Ihrer Anwendung unter befinde `resources/templates`. Beispiel: Die Vorlage `v1` für die Anwendung `my-app` in der Region USA West (Oregon) (us-west-2) für Konto 123456789012 befindet sich unter `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/v1`.

So wenden Sie eine gespeicherte Konfiguration auf eine laufende Umgebung an (AWS CLI)

- Geben Sie die gespeicherte Konfiguration in einem `update-environment`-Aufruf mit der Option `--template-name` an.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --template-
name v1
```

Elastic Beanstalk platziert Konfigurationen in diesem Speicherort, wenn sie mit erstellt wurde `aws elasticbeanstalk create-configuration-template`. Sie können auch gespeicherte Konfigurationen lokal ändern und sie selbst an diesem Speicherort ablegen.

Verwenden von Befehlszeilenoptionen

So ändern Sie Konfigurationsoptionen mit einem JSON-Dokument (AWS CLI)

1. Definieren Sie die Optionseinstellungen im JSON-Format in einer lokalen Datei.
2. Führen Sie `update-environment` mit der `--option-settings`-Option aus.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --option-
settings file://~/ebconfigs/as-zero.json
```

In diesem Beispiel definiert `as-zero.json` Optionen, die die Umgebung mit einem Minimum und Maximum an Null-Instances konfigurieren. Dies stoppt die Instances in der Umgebung, ohne die Umgebung zu beenden.

`~/ebconfigs/as-zero.json`

```
[
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MinSize",
        "Value": "0"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MaxSize",
        "Value": "0"
    },
    {
        "Namespace": "aws:autoscaling:updatepolicy:rollingupdate",
        "OptionName": "RollingUpdateEnabled",
        "Value": "false"
    }
]
```

> **Note**
>
> Das Einrichten der Konfigurationsoptionen mit `update-environment` überschreibt Einstellungen aus allen anderen Quellen.

Sie können auch Optionen aus Ihrer Umgebung mit entfernen `update-environment`.

So entfernen Sie Konfigurationsoptionen (AWS CLI)

- Führen Sie den Befehl `update-environment` mit der Option `--options-to-remove` aus.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --options-to-
remove Namespace=aws:autoscaling:launchconfiguration,OptionName=InstanceType
```

Durch das Entfernen von Optionen aus Ihrer Umgebung mit `update-environment` sind Einstellungen für dieselben Optionen möglich, die aus den Konfigurationsdateien im Quell-Bundle Ihrer Anwendung ermittelt werden. Wenn eine Option nicht mit einer der folgenden Methoden konfiguriert ist, wird der API-Standardwert angewendet, falls vorhanden. Details dazu finden Sie unter Precedence.

Allgemeine Optionen für alle Umgebungen

Namespaces

- aws:autoscaling:asg
- aws:autoscaling:launchconfiguration
- aws:autoscaling:scheduledaction
- aws:autoscaling:trigger
- aws:autoscaling:updatepolicy:rollingupdate
- aws:ec2:instances
- aws:ec2:vpc
- aws:elasticbeanstalk:application
- aws:elasticbeanstalk:application:environment
- aws:elasticbeanstalk:application:environmentsecrets
- aws:elasticbeanstalk:cloudwatch:logs
- aws:elasticbeanstalk:cloudwatch:logs:health
- aws:elasticbeanstalk:command
- aws:elasticbeanstalk:environment
- aws:elasticbeanstalk:environment:process:default
- aws:elasticbeanstalk:environment:process:process_name

- aws:elasticbeanstalk:environment:proxy:staticfiles
- aws:elasticbeanstalk:healthreporting:system
- aws:elasticbeanstalk:hostmanager
- aws:elasticbeanstalk:managedactions
- aws:elasticbeanstalk:managedactions:platformupdate
- aws:elasticbeanstalk:monitoring
- aws:elasticbeanstalk:sns:topics
- aws:elasticbeanstalk:sqsd
- aws:elasticbeanstalk:trafficsplitting
- aws:elasticbeanstalk:xray
- aws:elb:healthcheck
- aws:elb:loadbalancer
- aws:elb:listener
- aws:elb:listener:listener_port
- aws:elb:policies
- aws:elb:policies:policy_name
- aws:elbv2:listener:default
- aws:elbv2:listener:listener_port
- aws:elbv2:listenerrule:rule_name
- aws:elbv2:loadbalancer
- aws:rds:dbinstance

aws:autoscaling:asg

Konfigurieren Sie die Auto-Scaling-Gruppe der Umgebung. Weitere Informationen finden Sie unter the section called “Auto Scaling-Gruppe”.

Namespace: **`aws:autoscaling:asg`**

Name	Beschreibung	Standard	Zulässige Werte
Availability Zones	Availability Zones (AZs) sind unterschiedliche Standorte innerhalb einer	`Any`	`Any` `Any 1`

Name	Beschreibung	Standard	Zulässige Werte
	AWS Region, die so konzipiert sind, dass sie von Ausfällen in anderen Regionen isoliert sind AZs. Sie bieten kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen AZs in derselben Region. Wählen Sie die Anzahl der AZs für Ihre Instances.		`Any 2` `Any 3`
Cooldown	Ruhephasen verhindern, dass Amazon-EC2-Auto-Scaling zusätzlic he Skalierungen initiiert, bevor die Auswirkungen von vorherigen Skalierungen erkennbar werden. Eine Ruhephase ist die Zeit in Sekunden, die nach einer Skalierung verstreichen muss, bevor eine weitere Skalierung eingeleitet werden kann.	`360`	`0` auf `10000`

Custom Availability Zones	Definieren Sie die AZs für Ihre Instanzen.	Keine	`us-east-1a`
			`us-east-1b`
			`us-east-1c`
			`us-east-1d`
			`us-east-1e`
			`eu-centra l-1`

Name	Beschreibung	Standard	Zulässige Werte

EnableCapacityRebalancing	Gibt an, ob die Funktion Kapazitätsrebalancierung für Spot-Instances in der Auto-Scaling-Gruppe aktiviert werden soll. Weitere Informationen finden Sie unter Kapazitätsrebalancierung im Benutzerhandbuch zum Amazon-EC2-Auto-Scaling.
	
	Diese Option ist nur relevant, wenn `EnableSpot` auf `true` im `aws:ec2:instances` -Namespace gesetzt wird und es mindestens eine Spot-Instance in Ihrer Auto-Scaling-Gruppe gibt.
		`false`	`true`
			
			`false`

MinSize	Die minimale Anzahl von Instances, die Sie in Ihrer Auto-Scaling-Gruppe haben möchten.	1	1 auf `10000`
MaxSize	Die maximale Anzahl von Instances, die Sie in Ihrer Auto-Scaling-Gruppe haben möchten.	4	1 auf `10000`

aws:autoscaling:launchconfiguration

Konfigurieren Sie die Amazon Elastic Compute Cloud (Amazon-EC2)-Instances Ihrer Umgebung.

Die Instances, die in Ihrer Umgebung verwendet werden, werden entweder mit einer Amazon-EC2-Startvorlage oder einer Startkonfigurationsressource für Auto-Scaling-Gruppen erstellt. Diese folgenden Optionen funktionieren mit beiden Ressourcentypen.

Weitere Informationen finden Sie unter the section called “ EC2 Amazon-Instanzen”. Weitere Informationen zu Amazon Elastic Block Store (EBS) finden Sie auch im Amazon EBS-Kapitel im Amazon EC2 EC2-Benutzerhandbuch.

Namespace: **`aws:autoscaling:launchconfiguration`**

Name	Beschreibung	Standard	Zulässige Werte

DisableDefaultEC2SecurityGroup	Wenn Elastic Beanstalk auf den Standardwert von gesetzt ist`false`, erstellt es eine Standardsicherheitsgruppe, die Traffic aus dem Internet oder dem Load Balancer auf den Standardports für HTTP (80) zulässt. Es fügt diese Sicherheitsgruppe den EC2-Instances der Umgebung bei, wenn es die Umgebung erstellt.
	
	Wenn `true` Elastic Beanstalk auf gesetzt ist, weist Elastic Beanstalk den EC2-Instances für eine neue Umgebung nicht die Standardsicherheitsgruppe zu. Für eine bestehende Umgebung hebt Elastic Beanstalk die Zuweisung der standardmäßigen EC2-Sicherheitsgruppe zu den EC2-Instances Ihrer Umgebung auf. Daher müssen Sie auch die folgenden Konfigurationen festlegen:
	
	• Die `SecurityGroups` Option dieses Namespaces erfordert mindestens einen Wert, um Ihre benutzerdefinierte (n) Sicherheitsgruppe (n) zu definieren.
	• In Umgebungen mit einem Load Balancer müssen Sie die `SecurityGroups` Optionen auch in einem anderen Namespace festlegen, um benutzerdefinierte Sicherheitsgruppen für den Load Balancer zu konfigurieren. Für
		`false`	`true`
			
			`false`

Name	Beschreibung	Standard	Zulässige Werte
	Load Balancer für Anwendungen legen Sie die Option im Namespace fest. aws:elbv2:loadbalancer Für klassische Load Balancer legen Sie die Option im Namespace fest. aws:elb:loadbalancer • Weitere Informationen finden Sie unter EC2-Sicherheitsgruppen verwalten. Wenn ein Wert für EC2KeyName in einer Umgebung angegeben wird, für die `true` eine Standardsicherheitsgruppe `DisableDefaultEC2SecurityGroup` festgelegt ist, wird dieser Wert nicht mit den EC2-Instances verknüpft.		

Name	Beschreibung	Standard	Zulässige Werte

DisableIMDSv1	Auf einstellen, `true` um den Instanz-Metadatendienst Version 1 (IMDSv1) zu deaktivieren und durchzusetzen IMDSv2.
	
	Wird auf gesetzt`false`, um sowohl als auch IMDSv1 zu aktivieren IMDSv2.
	
	Die Instances für Ihre Umgebung lauten standardmäßig wie folgt, basierend auf dem Plattformbetriebssystem:
	
	• Windows Server AL2 und früher — aktiviert beide IMDSv1 und IMDSv2 (DisableIMDSv1Standardeinstellung) `false`
	• AL2023— nur aktivieren IMDSv2 (DisableIMDSv1standardmäßig) `true`
	
	Weitere Informationen finden Sie unter Konfiguration des Instance-Metadatenservice.
	
	⚠ Important
	Diese Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorl
		`false` – Plattformen, die auf Windows Server, Amazon Linux 2 und früheren Versionen basieren
		
		`true` – auf Amazon Linux 2023 basierende Plattformen
			`true`
			
			`false`

Name	Beschreibung	Standard	Zulässige Werte
	agen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerd efinierte Richtlinien anstelle unserer verwalteten Richtlini en verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungs konfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk -Umgebung auf Launch-Te mplates.		

Name	Beschreibung	Standard	Zulässige Werte
EC2KeyName	Sie können ein Schlüsselpaar verwenden, um sich sicher an der EC2-Instance anzumelden. Wenn ein Wert für `EC2KeyName` in einer Umgebung angegeben wird, für die `true` eine Standardsicherheitsgruppe `DisableDefaultEC2SecurityGroup` festgelegt ist, wird dieser Wert nicht mit den EC2-Instances verknüpft. **Note** Wenn Sie die Elastic-Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert.	Keine	

Name	Beschreibung	Standard	Zulässige Werte
IamInstanceProfile	Ein Instance-Profil ermöglicht AWS Identity and Access Management (IAM) -Benutzern und AWS -Diensten den Zugriff auf temporäre Sicherheitsanmeldedaten, um AWS API-Aufrufe zu tätigen. Geben Sie den Namen des Instance-Profils oder dessen ARN an. Beispiele: • `aws-elasticbeanstalk-ec2-role` • `arn:aws:iam::123456789012:instance-profile/aws-elasticbeanstalk-ec2-role` **Note** Wenn Sie die Elastic-Beanstalk-Konsole oder EB-CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdati festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.	Keine	Instance-Profilname oder ARN.

Name	Beschreibung	Standard	Zulässige Werte
ImageId	Sie können das Standard-AMI (Amazon Machine Image) mit Ihrer eigenen benutzerdefinierten AMI-ID überschreiben. Beispiel: `ami-1f316660`	Keine	

Name	Beschreibung	Standard	Zulässige Werte
InstanceType	Der Instance-Typ, mit dem Ihre Anwendung in einer Elastic-Beanstalk-Umgebung ausgeführt wird. **Important** Die Option `InstanceType` ist veraltet. Sie wird durch die neuere und leistungsfähigere Option `InstanceTypes` im `aws:ec2:instances` -Namespace ersetzt. Sie können diese neue Option verwenden, um eine Liste mit einem oder mehreren Instance-Typen für Ihre Umgebung anzugeben. Der erste Wert auf dieser List entspricht dem Wert der Option `InstanceType` , die in dem hier beschriebenen `aws:autoscaling:launchconfiguration` -Namespace enthalten ist. Wir empfehlen, dass Sie Instance-Typen mithilfe der neuen Option angeben. Wenn die neue Option angegeben wird, hat sie vor der vorherigen Vorrang. Weitere Informationen finden Sie unter the section called “Der Namespace aws:ec2:instances”.	Variiert je nach Konto und Region.	Ein EC2-Instance-Typ. Variiert je nach Konto, Region und Availability Zone. Sie können eine Liste von Amazon-EC2-Instance-Typen erhalten, die nach diesen Werten gefiltert werden. Weitere Informationen finden Sie unter Verfügbare Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch.

Name	Beschreibung	Standard	Zulässige Werte
	Die verfügbaren Instance-Typen hängen von den verwendeten Availability Zones und Region ab. Wenn Sie ein Subnetz wählen, bestimmt die Availability Zone, die dieses Subnetz enthält, die verfügbaren Instance-Typen. • Elastic Beanstalk unterstützt keine Amazon EC2 Mac-Instance-Typen. • Weitere Informationen zu Amazon EC2 EC2-Instance-Familien und -Typen finden Sie unter Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch. • Weitere Informationen zu den verfügbaren Instance-Typen in allen Regionen finden Sie unter Verfügbare Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch. **Note** Wenn Sie die Elastic-Beanstalk-Konsole oder EB-CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.		

Name	Beschreibung	Standard	Zulässige Werte

LaunchTemplateTagPropagationEnabled	Stellen Sie den Wert auf `true` ein, um die Propagierung von Umgebungs-Tags an die Startvorlagen für bestimmte Ressourcen zu ermöglichen, die der Umgebung bereitgestellt werden.
	
	Elastic Beanstalk kann Tags nur für die folgenden Ressourcen an Startvorlagen propagieren:
	
	• EBS-Datenträger
	• EC2-Instances
	• EC2-Netzwerkschnittstellen
	• CloudFormation Vorlagen starten, die eine Ressource definieren
	
	Diese Einschränkung besteht darin, dass Tags bei der Vorlagenerstellung CloudFormation nur für bestimmte Ressourcen zulässig sind. Weitere Informationen finden Sie TagSpecification im AWS CloudFormation Benutzerhandbuch.
	
	⚠ Important
	
	• Wenn Sie diesen Optionswert für eine bestehende Umgebung von `false` in `true` ändern, kann dies eine gravierende Änderung für zuvor vorhandene Tags herbeiführen.
		`false`	`true`
			
			`false`

Name	Beschreibung	Standard	Zulässige Werte
	• Wenn diese Funktion aktiviert ist, erfordert die Propagierung von Tags den Austausch von EC2, was zu Ausfallzeiten führen kann. Sie können fortlaufe nde Updates aktivieren, um Konfigurationsänderungen stapelweise anzuwenden und Ausfallzeiten während der Aktualisierung zu vermeiden. Weitere Informati onen finden Sie unter Konfigurationsänderungen. Weitere Informationen über Startvorl agen finden Sie in den folgenden Abschnitten: • Startvorlagen im Benutzerhandbuch für Amazon EC2 Auto Scaling. • Arbeiten mit Vorlagen im Benutzerh andbuch von AWS CloudFormation • Elastic-Beanstalk-Vorlagenausschnitt e im Benutzerhandbuch von AWS CloudFormation Weitere Informationen zu dieser Option finden Sie unter Tag-Propagierung zu Startvorlagen.		

Name	Beschreibung	Standard	Zulässige Werte
MonitoringInterval	Das Intervall (in Minuten), in dem CloudWatch Amazon-Metriken zurückgegeben werden sollen.	`5 minute`	`1 minute` `5 minute`

Name	Beschreibung	Standard	Zulässige Werte
SecurityGroups	Führt die Amazon EC2-Sicherheitsgruppen-IDs auf, die den EC2-Instances in der Auto Scaling Scaling-Gruppe zugewiesen werden sollen, um Firewallregeln für die Instances zu definieren. Verwenden Sie diese Option zusammen mit`DisableDefaultEC2SecurityGroup`, um Ihre eigenen benutzerdefinierten Sicherheitsgruppen anzuhängen, die Firewallregeln für die EC2-Instances definieren. Weitere Informationen finden Sie unter Umgebungen mit Lastenausgleich (mit mehreren Instanzen). **Important** Möglicherweise müssen Sie einige zusätzliche Konfigurationen vornehmen, um zu verhindern, dass eingehender Datenverkehr zu Ihren EC2-Instances blockiert wird. Dies gilt nur für Umgebungen mit mehreren Instanzen und benutzerdefinierten EC2-Sicherheitsgruppen. Die EC2-Sicherheitsgruppen müssen eine Regel für eingehenden Datenverkehr enthalten, die Zugriff auf den vom Load Balancer weitergeleiteten	`elasticbeanstalk-default`	

Name	Beschreibung	Standard	Zulässige Werte
	Datenverkehr gewährt. Weitere Informationen finden Sie unter Verwaltung von EC2-Sicherheitsgruppen in Umgebungen mit mehreren Instanzen. Sie können eine einzelne Zeichenfolge mit kommagetrennten Werten angeben, die bestehende Amazon EC2-Sicherheitsgruppen-IDs oder Verweise auf AWS::EC2::SecurityGroup Ressourcen enthalten, die in der Vorlage erstellt wurden. Sie müssen mindestens einen Wert für diese Option angeben, wenn `DisableDefaultEC2SecurityGroup` für diesen Namespace auf `true` gesetzt ist.		

Name	Beschreibung	Standard	Zulässige Werte
SSHSource Restriction	Wird zum Sperren des SSH-Zugriffs auf eine Umgebung verwendet. Beispielsweise können Sie den SSH-Zugriff auf die EC2-Instances sperren, sodass nur ein Bastion-Host auf die Instances im privaten Subnetz zugreifen kann. Die Zeichenfolge hat folgendes Format: *protocol*, *fromPort*, *toPort*, *source_restriction* *protocol* Das Protokoll für die Regel zum eingehenden Datenverkehr. *fromPort* Die Nummer des Start-Ports. *toPort* Die Nummer des End-Ports. *source_restriction* Der CIDR-Bereich (Classless Inter-Domain Routing) oder die Sicherheitsgruppe, durch die der Datenverkehr geleitet werden muss. Geben Sie die Sicherheitsgruppe mit der Sicherheitsgruppen-ID an. Um eine Sicherheitsgruppe aus einem anderen Konto anzugeben, geben Sie die AWS Konto-ID vor der Sicherheitsgruppen-ID ein,	Keine	

Name	Beschreibung	Standard	Zulässige Werte
	getrennt durch einen Schrägstrich. Das andere Konto muss sich in derselben AWS Region befinden. Beachten Sie die Syntax:*aws-account-id* **/***security-group-id* . Beispiel: **123456789012 /sg-99999999** Beispiele: • `tcp, 22, 22, 54.240.196.185/32` • `tcp, 22, 22, my-security-group-id` • `tcp, 22, 22, 123456789012/their-security-group-id`		

Name	Beschreibung	Standard	Zulässige Werte
BlockDeviceMappings	Fügen Sie zusätzliche Amazon-EBS-Volumes oder Instance-Speicher-Volumes auf allen Instances in der Auto-Scaling-Gruppe hinzu. **Important** Diese Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unter Migrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.	Keine	• Größe – muss zwischen 500 und 16 384 GiB liegen • Durchsatz – muss zwischen 125 und 1 000 Mebibyte pro Sekunde (MiB/s) liegen

Name	Beschreibung	Standard	Zulässige Werte
	Wenn Sie Instance-Speicher-Volumes zuweisen, ordnen Sie dem Gerätenamen nur einem Volume-Namen zu. Wir empfehlen jedoch, dass Sie bei der Zuweisung von Amazon-EBS-Volumes zusätzlich einige oder alle der folgenden Felder angeben (jedes Feld muss durch einen Doppelpunkt getrennt sein): • Snapshot-ID • Größe in GB • Bei Beenden löschen (`true` oder `false`) • Speichertyp (nur für `gp3`, `gp2`, `standard`, `st1`, `sc1`, oder `io1`) • IOPS (nur für `gp3` oder `io1`) • Durchsatz (nur für `gp3`) In folgendem Beispiel werden drei Amazon-EBS-Volumes angefügt, und zwar ein leeres gp2-Volume mit 100 GB und einem Snapshot, ein leeres io1-Volume mit 20 GB und 2 000 bereitgestellten IOPS und das Instance-Speicher-Volume `ephemeral0` . Sofern der Instance-Typ dies unterstützt, können mehrere Instance-Speicher-Volumes hinzugefügt werden. `/dev/sdj=:100:true:gp2,/dev/sdh=snap-51eef269,/dev/sdi=:20:true:io1:2000,/dev/sdb=ephemeral0`		

Name	Beschreibung	Standard	Zulässige Werte
RootVolumeType	Volume-Typ (Magnetic-, Allzweck-SSD- oder bereitgestellte IOPS-SSD-Volumes) des Amazon-EBS-Stamm-Volumes, das an die EC2-Instances der Umgebung angefügt ist. **Important** Diese Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten. Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unter Migrieren Sie Ihre Elastic Beanstalk Beanstalk-Umgebung auf Launch-Templates.	Variiert je nach Plattform.	`standard` für Magnetspeicher. `gp2` oder `gp3` für universelle SSD. `io1` für bereitgestellte IOPS-SSD.

Name	Beschreibung	Standard	Zulässige Werte
RootVolumeSize	Die Speicherkapazität des Amazon-EBS-Stamm-Volumes in GB. Dies ist erforderlich, wenn Sie für `RootVolumeType` bereitgestellte IOPS-SSD angeben. Beispiel, `"64"`.	Je nach Plattform für Magnetspeicher und Allzweck-SSD unterschiedlich. Kein Standardwert für bereitgestellte IOPS-SSD.	`10` bis `16384` GB für Allzweck-SSD und bereitgestellte IOPS-SSD. `8` bis `1024` GB für Magnetspeicher.
RootVolumeIOPS	Die gewünschten input/output Operationen pro Sekunde (IOPS) für ein bereitgestelltes IOPS-SSD-Root-Volume oder für ein SSD-Root-Volume für allgemeine Zwecke`gp3`. Das maximale Verhältnis von IOPS zur Volume-Größe liegt bei 500 zu 1. Beispielsweise muss ein Volume mit 3 000 IOPS mindestens eine Größe von 6 GiB haben.	Keine	`100` auf `20000` für bereitgestellte IOPS-SSD-Root-Volumes mit io1. `3000` auf `16000` für universelle `gp3` SSD-Root-Volumes.
RootVolumeThroughput	Der gewünschte Durchsatz von Mebibytes pro Sekunde (MIB/s) des Amazon-EBS-Stamm-Volumes, das an die EC2-Instance der Umgebung angefügt ist. **Note** Diese Option trifft nur auf `gp3`-Speichertypen zu.	Keine	`125` auf `1000`

aws:autoscaling:scheduledaction

Konfigurieren Sie die geplanten Aktionen der Auto-Scaling-Gruppe der Umgebung. Geben Sie für jede Aktion neben dem Optionsnamen, dem Namespace und dem Wert für jede Einstellung einen `resource_name` an. Beispiele finden Sie unter Der Namespace aws:autoscaling:scheduledaction.

Namespace: **`aws:autoscaling:scheduledaction`**

Name	Beschreibung	Standard	Zulässige Werte
StartTime	Wählen Sie für einmalige Aktionen das Datum und die Uhrzeit aus, zu der die Aktion ausgeführt werden soll. Legen Sie für wiederkehrende Aktionen fest, wann die Aktion aktiviert wird.	Keine	Ein eindeutiger ISO-8601-Zeitstempel für alle geplanten Skalierungen
EndTime	Ein Datum und eine Uhrzeit in der future (in der UTC/GMT Zeitzone), zu dem die geplante Skalierungsaktion nicht mehr wiederholt werden soll. Wenn Sie keinen angeben EndTime, wird die Aktion entsprechend dem `Recurrence` Ausdruck wiederholt. Beispiel: `2015-04-28T04:07:2Z` Wenn eine geplante Aktion endet, wird Amazon-EC2-Auto-Scaling nicht automatisch auf die vorherigen Einstellungen zurückgesetzt. Konfigurieren Sie daher gegebenenfalls eine zweite geplante Aktion, um zu den ursprünglichen Einstellungen zurückzukehren.	Keine	Ein eindeutiger ISO-8601-Zeitstempel für alle geplanten Skalierungen
MaxSize	Der Maximalwert für Instances , der beim Ausführen der Aktion angewendet werden soll	Keine	`0` auf `10000`

Name	Beschreibung	Standard	Zulässige Werte
MinSize	Der Minimalwert für Instances , der beim Ausführen der Aktion angewendet werden soll	Keine	`0` auf `10000`
DesiredCapacity	Legen Sie die gewünschte Anfangskapazität für die Auto-Scaling-Gruppe ein. Nachdem die geplante Aktion angewendet wurde, wird die gewünschte Kapazität basierend auf den Einstellungen angepasst.	Keine	`0` auf `10000`
Recurrence	Gibt die Häufigkeit an, mit der eine geplante Aktion ausgeführt werden soll. Falls Sie keine Wiederholung angeben, wird die Skalierung nur einmal (gemäß) ausgeführt `StartTime` .	Keine	Ein Cron-Ausdruck
Suspend	Setzen Sie die Option auf `true` fest, um eine wiederkehrende geplante Aktion vorübergehend zu deaktivieren.	`false`	`true` `false`

aws:autoscaling:trigger

Konfigurieren Sie Skalierungsauslöser für die Auto-Scaling-Gruppe der Umgebung.

> **Note**
>
> Drei Optionen in diesem Namensraum bestimmen, wie lange die Metrik für einen Auslöser außerhalb der definierten Grenzen bleiben kann, bevor der Auslöser initiiert wird. Diese Optionen stehen in folgendem Zusammenhang:
> `BreachDuration = Period * EvaluationPeriods`
> Die Standardwerte für diese Optionen (5, 5 bzw. 1) erfüllen diese Gleichung. Wenn Sie inkonsistente Werte angeben, ändert Elastic Beanstalk möglicherweise einen der Werte so, dass die Gleichung weiterhin erfüllt ist.

Namespace: **`aws:autoscaling:trigger`**

Name	Beschreibung	Standard	Zulässige Werte
BreachDuration	Der Zeitraum, für den der Grenzwert einer Metrik überschritten werden darf (wie unter `UpperThreshold` und `LowerThreshold` definiert), bevor der Auslöser ausgelöst wird.	`5`	`1` auf `600`
LowerBreachScaleIncrement	Gibt an, wie viele Amazon-EC2-Instances im Rahmen einer Skalierung entfernt werden sollen.	`-1`	
LowerThreshold	Falls die Messung diesen Wert der Verletzungsdauer unterschreitet, wird ein Auslöser ausgelöst.	`2000000`	`0` auf `20000000`

MeasureName	Für den Auto-Scaling-Auslöser verwendete Metrik.
	
	Note
	`HealthyHostCount`, `UnhealthyHostCount` und `TargetResponseTime` sind nur für Umgebungen mit einem dedizierten Load Balancer anwendbar. Dies sind keine gültigen Metrikwerte für Umgebungen, die mit einem gemeinsam genutzten Load Balancer konfiguriert sind. Weitere Informationen zu Load-Balancer-Typen finden Sie unter Load Balancer Ihrer Elastic-Beanstalk-Umgebung.
		`NetworkOut`	`CPUUtilization`
			
			`NetworkIn`
			
			`NetworkOut`
			
			`DiskWriteOps`
			
			`DiskReadBytes`
			
			`DiskReadOps`
			
			`DiskWriteBytes`
			
			`Latency`
			
			`RequestCount`
			
			`HealthyHostCount`

Name	Beschreibung	Standard	Zulässige Werte

			`UnhealthyHostCount`
			
			`TargetResponseTime`

Period	Gibt an, wie oft Amazon die Metriken für Ihren Trigger CloudWatch misst. Der Wert ist die Anzahl der Minuten zwischen zwei aufeinanderfolgenden Zeiträumen.	`5`	`1` auf `600`
EvaluationPeriods	Die Anzahl der aufeinanderfolgenden Auswertungszeiträumen, die verwendet, um festzustellen, ob eine Verletzung auftritt.	`1`	`1` auf `600`

Statistic	Die Statistik, die der Auslöser verwendet, z. B. `Average`.	`Average`	`Minimum`
			
			`Maximum`
			
			`Sum`
			
			`Average`

Name	Beschreibung	Standard	Zulässige Werte

Unit	Die Maßeinheit für die Auslöserm essung, z. B. `Bytes`.	`Bytes`	`Seconds`
			`Percent`
			`Bytes`
			`Bits`
			`Count`
			`Bytes/Second`
			`Bits/Second`
			`Count/Second`
			`None`

UpperBreachScaleIn crement	Gibt an, wie viele Amazon-EC2-Instances im Rahmen einer Skalierun g hinzugefügt werden sollen.	1	
UpperThreshold	Falls die Messung diesen Wert der Verletzungsdauer überschreitet, wird ein Auslöser ausgelöst.	`6000000`	`0` auf `20000000`

aws:autoscaling:updatepolicy:rollingupdate

Konfigurieren Sie fortlaufende Aktualisierungen für die Auto-Scaling-Gruppe der Umgebung.

Namespace: **`aws:autoscaling:updatepolicy:rollingupdate`**

Name	Beschreibung	Standard	Zulässige Werte
MaxBatchSize	Anzahl der Instances , die in jedem fortlaufe nden Aktualisierungs-Batch enthalten sind.	Ein Drittel der Mindestgröße der Auto-Scaling-Gruppe, gerundet	1 auf `10000`

Name	Beschreibung	Standard	Zulässige Werte
		auf die nächsthöhere Ganzzahl.	
MinInstancesInService	Mindestanzahl der Instances, die in der Auto-Scaling-Gruppe aktiv sein müssen, wenn andere Instances beendet werden.	Die Mindestgröße der Auto-Scaling-Gruppe oder eine Instance weniger als die maximale Größe der Auto-Scaling-Gruppe – je nachdem, welcher Wert der niedrigere ist.	`0` auf 9999

Name	Beschreibung	Standard	Zulässige Werte

RollingUpdateEnabled	Wenn für diesen Parameter `true` festgelegt ist, sind fortlaufende Aktualisierungen für eine Umgebung aktiviert. Fortlaufende Aktualisierungen sind sinnvoll, wenn Sie häufig kleinere Aktualisierungen für die Elastic-Beanstalk-Softwareanwendung ohne Ausfallzeit der Anwendung ausführen möchten.
	
	Wird dieser Wert auf "true" gesetzt, aktiviert das System automatisch die Optionen `MaxBatchSize` , `MinInstancesInService` und `PauseTime` . Wenn Sie für eine dieser Optionen einen Wert angeben, wird auch die Option `RollingUpdateEnabled` automatisch auf den Wert `true` gesetzt. Mit dem Wert `false` lassen sich fortlaufe
		`false`	`true`
			
			`false`

Name	Beschreibung	Standard	Zulässige Werte
	nde Aktualisierungen deaktivieren. **Note** Wenn Sie die Elastic-Beanstalk-Konsole oder EB-CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.		

Name	Beschreibung	Standard	Zulässige Werte

RollingUpdateType	Das umfasst drei Arten: zeitbasierte fortlaufende Updates, zustandsbasierte fortlaufende Updates und unveränderliche Updates.
	
	Zeitbasierte, fortlaufende Updates werden PauseTime zwischen den Batches angewendet. Bei zustandsbasierten fortlaufenden Aktualisierungen wird gewartet, bis die neuen Instances die Zustandsprüfungen bestanden haben, bevor mit dem nächsten Batch begonnen wird. Bei unveränderlichen Aktualisierungen wird eine komplette Instance-Gruppe in einer neuen Auto-Scaling-Gruppe gestartet.
	
	Note
	Wenn Sie die Elastic-Beanstalk-
		`Time`	`Time`
			
			`Health`
			
			`Immutable`

Name	Beschreibung	Standard	Zulässige Werte
	Konsole oder EB-CLI verwenden , um eine Umgebung zu erstellen , können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.		
PauseTime	Gibt den Zeitraum (In Sekunden, Minuten, oder Stunden) an, den der Elastic-Beanstalk-Service nach erfolgter Aktualisierung eines Instance-Batchs wartet, bevor mit dem nächsten Batch begonnen wird.	Wird automatisch auf Basis von Instance-Typ und Container berechnet.	`PT0S`* (0 Sekunden) bis `PT1H` (1 Stunde)

Name	Beschreibung	Standard	Zulässige Werte
Timeout	Gibt den maximalen Zeitraum (In Minuten oder Stunden) an, der gewartet wird, bis alle Instances in einem Batch die Zustandsprüfungen bestehen, bevor die Aktualisierung abgebrochen wird.	`PT30M` (30 Minuten)	`PT5M`* (5 Minuten) bis `PT1H` (1 Stunde) * Format für die ISO8601Dauer: `PT#H#M#S` wobei jedes # für die Anzahl der Stunden, Minuten bzw. and/or Sekunden steht.

aws:ec2:instances

Konfigurieren Sie die Instances Ihrer Umgebung, einschließlich Spot-Optionen. Dieser Namespace ergänzt `aws:autoscaling:launchconfiguration` und `aws:autoscaling:asg`.

Weitere Informationen finden Sie unter the section called “Auto Scaling-Gruppe”.

Namespace: **`aws:ec2:instances`**

Name	Beschreibung	Standa	Zulässige Werte

EnableSpot	Aktivieren Sie Spot-Instance-Anforderungen für Ihre Umgebung. Wenn `false`, werden einige Optionen in diesem Namespace nicht wirksam.
	
	Important
	Diese Optionseinstellung kann dazu führen, dass Elastic Beanstalk eine bestehende Umgebung mit Startkonfigurationen migriert, um Vorlagen zu starten. Dazu sind
		`false`	`true`
			
			`false`

Name	Beschreibung	Standa	Zulässige Werte
	die erforderlichen Berechtigungen erforderlich, um Startvorlagen zu verwalten . Diese Berechtigungen sind in unserer verwalteten Richtlinie enthalten. Wenn Sie benutzerdefinierte Richtlinien anstelle unserer verwalteten Richtlinien verwenden, schlagen die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Sie Ihre Umgebungskonfiguration aktualisieren. Weitere Informationen und andere Überlegungen finden Sie unterMigrieren Sie Ihre Elastic Beanstalk Beanstalk -Umgebung auf Launch-Templates.		

Name	Beschreibung	Standa	Zulässige Werte
InstanceTypes	Eine durch Kommas getrennte Liste von Instance-Typen, die in Ihrer Umgebung verwendet werden sollen (zum Beispiel, `t2.micro, t3.micro`). Wenn `EnableSpot` ist `true` und auf gesetzt `SpotAllocationStrategy` ist`capacity-optimized-prioritized` , bestimmt die in dieser Option angegebene Werteliste die Priorität des Instance-Typs für die Spot-Instance-Zuweisungsstrategie. Wenn Spot-Instances deaktiviert sind (`EnableSpot` ist `false`), wird nur der erste Instance-Typ in der Liste verwendet. Der erste Instance-Typ in der Liste in dieser Option entspricht dem Wert der `InstanceType` -Option im `aws:autoscaling:launchconfiguration` -Namespace. Wir empfehlen nicht, die letztere Option zu verwenden, da sie veraltet ist. Wenn Sie beide angeben, wird der erste Instance-Typ in der Liste in der `InstanceTypes` -Option verwendet und `InstanceType` wird ignoriert. Die verfügbaren Instance-Typen hängen von den verwendeten Availability Zones und Region ab. Wenn Sie ein Subnetz wählen, bestimmt die Availability Zone, die dieses Subnetz	Eine Liste mit zwei Instanc Typen. Variier je nach Konto und Regior	Ein bis vierzig EC2-Instance-Typen. Wir empfehlen mindestens zwei. Variiert je nach Konto, Region und Availability Zone. Sie können eine Liste von Amazon-EC2-Instance-Typen erhalten, die nach diesen Werten gefiltert werden. Weitere Informationen finden Sie unter Verfügbare Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch. Die Instance-Typen müssen alle Teil derselben Architektur sein (`arm64`,`x86_64`,`i386`). `SupportedArchitectures` ist auch Teil dieses Namespace. Wenn Sie Werte angeben für `SupportedArchitectures` , müssen Werte, die Sie für `InstanceTypes` eingeben, zu einer einzigen der von

Name	Beschreibung	Standa	Zulässige Werte
	enthält, die verfügbaren Instance-Typen. • Elastic Beanstalk unterstützt keine Amazon EC2 Mac-Instance-Typen. • Weitere Informationen zu Amazon EC2 EC2-Instance-Familien und -Typen finden Sie unter Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch. • Weitere Informationen zu den verfügbaren Instance-Typen in allen Regionen finden Sie unter Verfügbare Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch. **Note** Einige ältere AWS Konten bieten Elastic Beanstalk möglicherweise Standard-Instance-Typen, die Spot-Instances nicht unterstützen (z. B. t1.micro). Wenn Sie Spot-Instance-Anforderungen aktivieren und eine Fehlermeldung zu einem Instance-Typ erhalten, der keine Spot-Flotten unterstützt, müssen Sie Instance-Typen konfigurieren, die Spot-Flotten unterstützen. Verwenden Sie zur Auswahl von Spot-Instance-Typen Spot Instance Advisor.		Ihnen für `Supported Architectures` bereitgestellten Architekturen gehören.

Name	Beschreibung	Standa	Zulässige Werte
	Wenn Sie die Umgebungskonfiguration aktualisieren und einen oder mehrere Instance-Typen aus der `InstanceTypes` -Option entfernen, beendet Elastic Beanstalk alle Amazon-EC2-Instances, die auf einem der entfernten Instance-Typen ausgeführt werden. Die Auto-Scaling-Gruppe Ihrer Umgebung startet dann nach Bedarf neue Instances, um die gewünschte Kapazität zu vervollständigen, unter Verwendung der aktuell angegebenen Instance-Typen.		

SpotAllocationStrategy	Gibt die Strategie zur Zuweisung von Spot-Instances an, die bestimmt, wie Spot-Instances aus den verfügbaren Spot-Kapazitätspools zugewiesen werden.
	
	Wenn auf gesetzt`capacity-optimized-prioritized` , `InstanceTypes` legt die Reihenfolge der Werte die Priorität des Instance-Typs für die Zuteilung fest.
	
	Diese Option ist nur relevant, wenn `EnableSpot` `true` ist.
		`capac`
		`optim`
			`capacity-optimized`
			
			`price-capacity-optimized`
			
			`capacity-optimized-prioritized`
			
			`lowest-price`

Name	Beschreibung	Standa	Zulässige Werte
SpotFleet OnDemandBase	Die Mindestanzahl von On-Demand-Instances, die von der Auto Scaling-Gruppe bereitgestellt werden, bevor Spot-Instances bei der Skalierung Ihrer Umgebung in Betracht gezogen werden. Diese Option ist nur relevant, wenn `EnableSpot` `true` ist.	`0`	Option `0` bis `MaxSize` Option im `aws:autoscaling:asg` -Namespace
SpotFleet OnDemandAboveBasePercentage	Der Prozentsatz der On-Demand-Instances als Teil der zusätzlichen Kapazität, die von der Auto-Scaling-Gruppe über die `SpotOnDemandBase` -Instances hinaus bereitgestellt wird. Diese Option ist nur relevant, wenn `EnableSpot` `true` ist.	`0` für eine Umgeb mit einer einzelr Instanc `70` für eine Umgeb mit Lastau eich	`0` auf `100`

Name	Beschreibung	Standa	Zulässige Werte

SpotMaxPrice	Der Höchstpreis pro Einheitsstunde in USD, den Sie für eine Spot-Instance zu zahlen bereit sind. Empfehlungen zu Höchstpreisoptionen für Spot-Instances finden Sie in der Preisentwicklung für Spot-Instances im Amazon EC2 EC2-Benutzerhandbuch.
	
	Diese Option ist nur relevant, wenn `EnableSpot` `true` ist.
		On-Demand-Preis für jeden Instanc Typ. Der Wert der Option ist in diesem Fall `null`.	`0.001` auf `20.0`
			
			`null`

Name	Beschreibung	Standa	Zulässige Werte

SupportedArchitectures	Eine durch Kommas getrennte Liste von Amazon-EC2-Instance-Typen, die in Ihrer Umgebung verwendet werden sollen.
	
	Elastic Beanstalk unterstützt Instance-Typen, die auf den folgenden Prozessorarchitekturen basieren:
	
	• AWS Graviton 64-Bit-Arm-Architektur (arm64)
	• 64-Bit Architektur (x86_64)
	• 32-Bit Architektur (i386)
	
	Weitere Informationen zu Prozessorarchitektur und Amazon EC2-Instance-Typen finden Sie unter the section called “EC2 Amazon-Instance-Typen”.
		Keine	`arm64`
			
			`x86_64`
			
			`i386`
			
			Note
			Die 32-Bit-Architektur `i386` wird von den meisten Elastic Beanstalk-Plattformen nicht unterstützt. Wir empfehlen, dass Sie die `x86_64` oder stattdessen `arm64` Architekturtypen auswählen.

aws:ec2:vpc

Konfigurieren Sie Ihre Umgebung, um Ressourcen in einer benutzerdefinierten Amazon Virtual Private Cloud (Amazon-VPC) zu starten. Falls Sie keine Einstellungen in diesem Namespace konfigurieren, startet Elastic Beanstalk die Ressourcen in der Standard-VPC.

Namespace: **`aws:ec2:vpc`**

Name	Beschreibung	Standard	Zulässige Werte
VPCId	Die ID der Amazon-VPC.	Keine	

Name	Beschreibung	Standard	Zulässige Werte
Subnets	Das Subnetz oder die Subnetze IDs der Auto Scaling Scaling-Gruppe. Wenn Sie mehrere Subnetze haben, geben Sie den Wert als einzelne, durch Kommas getrennte Zeichenfolge des IDs Subnetzes an (z. B.). `"subnet-11111111,subnet-22222222"`	Keine	
ELBSubnets	Das Subnetz oder IDs die Subnetze für den Elastic Load Balancer. Wenn Sie mehrere Subnetze haben, geben Sie den Wert als einzelne, durch Kommas getrennte Zeichenfolge des IDs Subnetzes an (z. B.). `"subnet-11111111,subnet-22222222"`	Keine	

ELBScheme	Geben Sie `internal` an, wenn ein interner Load Balancer in der Amazon-VPC erstellt werden soll, damit nur innerhalb der Amazon-VPC auf die Elastic-Beanstalk-Anwendung zugegriffen werden kann. Wenn Sie einen anderen Wert als `public` oder `internal` angeben, wird dieser von Elastic Beanstalk ignoriert.	`public`	`public`
			
			`internal`

DBSubnets	Enthält die Subnetze IDs der Datenbank. Dieser Parameter wird nur verwendet, wenn eine Amazon-RDS-DB-Instance als Teil der Anwendung hinzugefügt werden soll. Wenn Sie mehrere Subnetze haben, geben Sie den Wert als einzelne, durch Kommas getrennte Zeichenfolge des Subnetzes an IDs (z. B.). `"subnet-11111111,subnet-22222222"`	Keine	

Name	Beschreibung	Standard	Zulässige Werte

AssociatePublicIpAddress	Gibt an, ob die Instances mit öffentlichen IP-Adressen in der Amazon-VPC gestartet werden sollen. Instances mit öffentlichen IP-Adressen erfordern kein NAT-Gerät, um mit dem Internet zu kommunizieren. Sie müssen den Wert auf `true` setzen, wenn der Load Balancer und die Instances in ein einziges öffentliches Subnetz einbezogen werden sollen.
	
	Diese Option hat keine Auswirkungen auf eine Single-Instance-Umgebung, die stets eine einzelne Amazon-EC2-Instance mit einer Elastic-IP-Adresse aufweist. Die Option ist relevant für skalierbare Umgebungen mit Lastausgleich.
		Keine	`true`
			
			`false`

aws:elasticbeanstalk:application

Konfigurieren Sie einen Pfad für die Zustandsprüfung der Anwendung. Weitere Informationen finden Sie unter Grundlegende Zustandsberichte.

Namespace: **`aws:elasticbeanstalk:application`**

Name	Beschreibung	Standa	Zulässige Werte

URL für die Zustandsprüfung der Anwendung	Der Pfad, an den Zustandsprüfungsanfragen gesendet werden. Wenn dieser Pfad nicht festgelegt ist, versucht der Load Balancer, eine TCP-Verbindung über Port 80 herzustellen, um den Zustand Ihrer Anwendung zu überprüfen. Geben Sie als erstes Zeichen im Pfad `/` an, damit eine HTTP GET-Anforderung an diesen Pfad gesendet wird. Sie können vor dem Pfad auch ein Protokoll (HTTP, HTTPS, TCP oder SSL) und einen	Keine	Gültige Werte sind:
			
			`/` (HTTP GET zum Stammpfad)
			
			`/health`
			
			`HTTPS:443/`
			
			`HTTPS:443/health`

Name	Beschreibung	Standa	Zulässige Werte
	Port zur Überprüfung der HTTPS-Verbindung einbinden oder einen anderen als den Standard-Port verwenden. **Note** Wenn Sie die Elastic-Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert.		

Die EB-CLI und Elastic-Beanstalk-Konsole wenden die für die vorhergehenden Optionen empfohlenen Werte an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter Empfohlene Werte.

aws:elasticbeanstalk:application:environment

Konfigurieren Sie Umgebungseigenschaften für die Anwendung.

Namespace: **`aws:elasticbeanstalk:application:environment`**

Name	Beschreibung	Standard	Zulässige Werte
Name einer beliebigen Umgebungsvariable	Dient der Übergabe von Schlüssel-Wert-Paaren.	Keine	Wert einer beliebigen Umgebungsvariable.

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

aws:elasticbeanstalk:application:environmentsecrets

Konfigurieren Sie Umgebungsvariablen, die als Umgebungsgeheimnisse für Ihre Anwendung dienen. In Umgebungsgeheimnissen werden AWS Secrets Manager Geheimnisse oder AWS Systems Manager Parameter Store-Parameter gespeichert.

Namespace: **`aws:elasticbeanstalk:application:environmentsecrets`**

Name	Beschreibung	Standa	Zulässige Werte
OptionName	Gibt den Namen der Umgebungs variablen an, die den geheimen Speicher- oder Parameterspeicherwert enthalten soll.	Keine	Name einer beliebigen Umgebungsvariable
Wert	Gibt den ARN für den Wert an, der in AWS Secrets Manager oder AWS Systems Manager Parameter Store gespeichert ist. Beim Instance-Bootstrapping initiiert Elastic Beanstalk die Umgebungsvariable auf den Wert, der in dieser ARN-Ressource gespeichert ist. **Note** Stellen Sie sicher, dass die EC2-Instance-Profilrolle Ihrer Umgebung über die erforderl ichen Berechtigungen verfügt, um auf das Geheimnis und den Parameter zuzugreifen. ARNs Weitere Informationen finden Sie unter Erforderliche IAM-Berechtigungen.	Keine	Gültiger ARN-Wert für einen AWS Secrets Manager geheimen oder AWS Systems Manager Parameter Store-Par ameterwert.

Weitere Informationen finden Sie unter Geheimnisse als Umgebungsvariablen konfigurieren.

aws:elasticbeanstalk:cloudwatch:logs

Konfigurieren Sie das Instance-Protokoll-Streaming für Ihre Anwendung.

Namespace: **`aws:elasticbeanstalk:cloudwatch:logs`**

Name	Beschreibung	Standard	Zulässige Werte
StreamLogs	Gibt an, ob Gruppen in CloudWatch Logs für Proxy- und Bereitstellungsprotokolle erstellt und Logs von jeder Instanz in Ihrer Umgebung gestreamt werden sollen.	`false`	`true` `false`
DeleteOnTerminate	Gibt an, ob die Protokollgruppen gelöscht werden sollen, wenn die Umgebung beendet wird. Lautet der Wert `false`, werden die Protokolle für die unter `RetentionInDays` angegebene Dauer gespeichert.	`false`	`true` `false`
RetentionInDays	Gibt die Anzahl der Tage an, die Protokollereignisse gespeichert werden, bevor sie ablaufen.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

aws:elasticbeanstalk:cloudwatch:logs:health

Konfigurieren Sie das Umgebungszustand-Protokoll-Streaming für Ihre Anwendung.

Namespace: **`aws:elasticbeanstalk:cloudwatch:logs:health`**

Name	Beschreibung	Standard	Zulässige Werte
HealthStreamingEnabled	Gibt für Umgebungen mit aktivierter erweiterter Zustandsberichterstattung an, ob eine Gruppe in CloudWatch Logs for Environment Health erstellt und Integritätsdaten der Elastic Beanstalk Beanstalk-Umgebung archiviert werden sollen. Weitere Informationen zur Aktivierung des erweiterten Zustands finden Sie unter `aws:elasticbeanstalk:health reporting:system`.	`false`	`true` `false`
DeleteOnTerminate	Gibt an, ob die Protokollgruppe gelöscht werden soll, wenn die Umgebung beendet wird. Bei `false` werden die Zustandsdaten `RetentionInDays` Tage gespeichert.	`false`	`true` `false`
RetentionInDays	Die Anzahl der Tage, wie lange archivierte Zustandsdaten aufbewahrt werden, bevor sie ablaufen.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

aws:elasticbeanstalk:command

Konfigurieren Sie die Bereitstellungsrichtlinie für den Anwendungscode. Weitere Informationen finden Sie unter the section called “Optionen für die Bereitstellung”.

Namespace: **`aws:elasticbeanstalk:command`**

Name	Beschreibung	Standard	Zulässige Werte

DeploymentPolicy	Wählen Sie eine Bereitstellungsrichtlinie für die Bereitstellung von Anwendungsversionen aus.
	
	Note
	Wenn Sie die Elastic-Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert.
		`AllAtOnce`	`AllAtOnce`
			
			`Rolling`
			
			`RollingWithAdditionalBatch`
			
			`Immutable`
			
			`TrafficSplitting`

Timeout	Gibt an, wie viele Sekunden gewartet wird, bis eine Instance die Ausführung von Befehlen abgeschlossen hat. Elastic Beanstalk fügt dem `Timeout`-Wert intern 240 Sekunden (vier Minuten) hinzu. Beispiel: Der effektive Timeout liegt standardmäßig bei 840 Sekunden (600 + 240) oder 14 Minuten.	`600`	1 auf `3600`

BatchSizeType	Der Zahlentyp, der in angegeben ist. BatchSize
	
	Note
	Wenn Sie die Elastic-Beanstalk-Konsole oder EB-CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer
		`Percentage`	`Percentage`
			
			`Fixed`

Name	Beschreibung	Standard	Zulässige Werte
	Konfigurationsdatei festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.		
BatchSize	Der Prozentsatz oder die feste Anzahl der Amazon-EC2-Instances in der Auto-Scaling-Gruppe, für die gleichzeitig Bereitstellungen ausgeführt werden. Gültige Werte variieren je nach BatchSizeTypeverwendeter Einstellung. **Note** Wenn Sie die Elastic-Beanstalk-Konsole oder EB-CLI verwenden , um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.	`100`	1 auf `100` (`Percentage`). 1zu aws:autoscaling:asg::() MaxSize `Fixed`

IgnoreHealthCheck	Damit lässt sich festlegen, dass eine Bereitstellung nicht aufgrund fehlgeschlagener Zustandsprüfungen abgebrochen wird.	`false`	`true`
			
			`false`

aws:elasticbeanstalk:environment

Konfigurieren Sie die Architektur und die Servicerolle für die Umgebung.

Namespace: **`aws:elasticbeanstalk:environment`**

Name	Beschreibung	Standard	Zulässige Werte

EnvironmentType	Legen Sie `SingleInstance` fest, damit eine EC2-Instance ohne Load Balancer gestartet wird.	`LoadBalanced`	`SingleInstance`
			
			`LoadBalanced`

ServiceRole	Der Name einer IAM-Rolle, die Elastic Beanstalk zur Verwaltung der Umgebungsressourcen verwendet. Geben Sie den Namen einer Rolle (optional mit einem benutzerdefinierten Pfad als Präfix) oder ihren ARN an. Beispiele: • `aws-elasticbeanstalk-service-role` • *`custom-path`* `/`*`custom-role`* • `arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role` **Note** Wenn Sie die Elastic-Beanstalk-Konsole oder EB-CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.	Keine	IAM-Rollennamen, Pfad/Name oder ARN

Name	Beschreibung	Standard	Zulässige Werte

LoadBalancerType	Der Typ des Load Balancers für Ihre Umgebung. Weitere Informationen finden Sie unter the section called “Load Balancer”.	`classic`	`classic`
			`application`
			`network`

LoadBalancerIsShared	Gibt an, ob der Load Balancer der Umgebung dediziert oder freigegeben ist. Diese Option kann nur für einen Application Load Balancer festgelegt werden. Sie kann nach der Erstellung der Umgebung nicht geändert werden. Wenn `false`, verfügt die Umgebung über einen eigenen dedizierten Load Balancer, der von Elastic Beanstalk erstellt und verwaltet wird. Wenn `true`, verwendet die Umgebung einen freigegebenen, von Ihnen erstellten Load Balancer, der in der Option `SharedLoadBalancer` des Namespace aws:elbv2:loadbalancer angegeben ist.	`false`	`true` `false`

aws:elasticbeanstalk:environment:process:default

Konfigurieren Sie den Standardprozess für die Umgebung.

Namespace: **`aws:elasticbeanstalk:environment:process:default`**

Name	Beschreibung	Standard	Zulässige Werte
DeregistrationDelay	Die Zeit in Sekunden, die gewartet werden muss, bis aktive Anforderungen	`20`	`0` auf `3600`

Name	Beschreibung	Standard	Zulässige Werte
	abgeschlossen sind, bevor eine Abmeldung erfolgt.		
HealthCheckInterval	Gibt an, wie häufig (in Sekunden) Elastic-Load-Balancing-Zustandsprüfungen für die Amazon-EC2-Instances der Anwendung ausführt.	Bei Classic oder Application Load Balancer: `15` Bei Network Load Balancer: `30`	Bei Classic oder Application Load Balancer: `5` bis `300` Bei Network Load Balancer: `10`, `30`
HealthCheckPath	Pfad, an den HTTP-Anfragen für Zustandsprüfungen gesendet werden.	`/`	Ein routingfähiger Pfad.
HealthCheckTimeout	Gibt an, wie lange (in Sekunden) während einer Zustandsprüfung auf eine Antwort gewartet wird. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	`5`	`1` auf `60`
HealthyThresholdCount	Die Anzahl aufeinander erfolgender erfolgreicher Anforderungen vor Elastic-Load-Balancing ändert den Instance-Zustand.	Bei Classic oder Application Load Balancer: `3` Bei Network Load Balancer: `5`	`2` auf `10`

Name	Beschreibung	Standard	Zulässige Werte
MatcherHTTPCode	Eine durch Komma getrennte Liste von HTTP-Codes (s), die anzeigt, dass eine Instance fehlerfrei ist. Diese Option ist nur für Umgebungen mit einem Applicati on Load Balancer bestimmt.	`200`	Bei Application Load Balancer: `200` bis `499` Bei Network Load Balancer: `200` bis `399`
Port	Port, den der Prozess überwacht.	`80`	1 auf `65535`

Protocol	Gibt das vom Prozess verwendete Protokoll an.
	
	Bei einem Applicati on Load Balancer können Sie diese Option nur auf `HTTP` oder `HTTPS` festlegen.
	
	Bei einem Network Load Balancer können Sie diese Option nur auf festlegen `TCP`.
		Bei Classic oder Application Load Balancer: `HTTP`
		
		Bei Network Load Balancer: `TCP`
			`TCP`
			
			`HTTP`
			
			`HTTPS`

Name	Beschreibung	Standard	Zulässige Werte

StickinessEnabled	Legen Sie den Wert auf "true" fest, um Sticky Sessions zu aktivieren.
	
	Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.
		`'false'`	`'false'`
			
			`'true'`

StickinessLBCookieDuration	Gibt die Lebensdauer des Sticky Session-Cookies in Sekunden an. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	`86400` (ein Tag)	1 auf `604800`
StickinessType	Legen Sie den Wert auf `lb_cookie` fest, um Cookies für Sticky Sessions zu nutzen. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	`lb_cookie`	`lb_cookie`

Name	Beschreibung	Standard	Zulässige Werte
UnhealthyThresholdCount	Die Anzahl aufeinander erfolgender erfolgloser Anforderungen vor Elastic-Load-Balancing ändert den Instance-Zustand.	`5`	`2` auf `10`

aws:elasticbeanstalk:environment:process:process_name

Konfigurieren Sie zusätzliche Prozesse für die Umgebung.

Namespace: **`aws:elasticbeanstalk:environment:process:`*`process_name`***

Name	Beschreibung	Standard	Zulässige Werte
DeregistrationDelay	Die Zeit in Sekunden, die gewartet werden muss, bis aktive Anforderungen abgeschlossen sind, bevor eine Abmeldung erfolgt.	`20`	`0` auf `3600`
HealthCheckInterval	Gibt an, wie häufig (in Sekunden) Elastic-Load-Balancing Zustandsprüfungen für die Amazon-EC2-Instances der Anwendung ausführt.	Bei Classic oder Application Load Balancer: `15` Bei Network Load Balancer: `30`	Bei Classic oder Application Load Balancer: `5` bis `300` Bei Network Load Balancer: `10`, `30`
HealthCheckPath	Pfad, an den HTTP-Anfragen für Zustandsprüfungen gesendet werden.	`/`	Ein routingfähiger Pfad.

Name	Beschreibung	Standard	Zulässige Werte
HealthCheckTimeout	Gibt an, wie lange (in Sekunden) während einer Zustandsprüfung auf eine Antwort gewartet wird. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	5	1 auf 60
HealthyThresholdCount	Die Anzahl aufeinander erfolgender erfolgreicher Anforderungen vor Elastic-Load-Balancing ändert den Instance-Zustand.	Bei Classic oder Application Load Balancer: 3 Bei Network Load Balancer: 5	2 auf 10
MatcherHTTPCode	Eine durch Komma getrennte Liste von HTTP-Code(s), die anzeigt, dass eine Instance fehlerfrei ist. Diese Option ist nur für Umgebungen mit einem Application Load Balancer bestimmt.	200	Bei Application Load Balancer: 200 bis 499 Bei Network Load Balancer: 200 bis 399
Port	Der Port, den der Prozess überwacht.	80	1 auf 65535

Name	Beschreibung	Standard	Zulässige Werte

Protocol	Gibt das vom Prozess verwendete Protokoll an.
	
	Bei einem Application Load Balancer können Sie diese Option nur auf `HTTP` oder `HTTPS` festlegen.
	
	Bei einem Network Load Balancer können Sie diese Option nur auf festlegen `TCP`.
		Bei Classic oder Application Load Balancer: `HTTP`
		
		Bei Network Load Balancer: `TCP`
			`TCP`
			
			`HTTP`
			
			`HTTPS`

StickinessEnabled	Legen Sie den Wert auf "true" fest, um Sticky Sessions zu aktivieren.
	
	Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.
		`'false'`	`'false'`
			
			`'true'`

Name	Beschreibung	Standard	Zulässige Werte
StickinessLBCookieDuration	Gibt die Lebensdauer des Sticky Session-Cookies in Sekunden an. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	`86400` (ein Tag)	1 auf `604800`
StickinessType	Legen Sie den Wert auf `lb_cookie` fest, um Cookies für Sticky Sessions zu nutzen. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	`lb_cookie`	`lb_cookie`
UnhealthyThresholdCount	Die Anzahl aufeinanderfolgender erfolgloser Anforderungen vor Elastic-Load-Balancing ändert den Instance-Zustand.	`5`	2 auf `10`

aws:elasticbeanstalk:environment:proxy:staticfiles

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereits, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

> **Note**
>
> Dieser Namespace gilt für Plattformzweige, die auf Amazon Linux 2 und höher basieren. Wenn Ihre Umgebung eine Plattformversion verwendet, die auf Amazon-Linux-AMI (vor Amazon-Linux-2) basiert, finden Sie unter the section called “Plattformspezifische Optionen” Informationen zu plattformspezifischen Namespaces für statische Dateien.

Namespace: **`aws:elasticbeanstalk:environment:proxy:staticfiles`**

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitstellt. Starten Sie den Wert mit `/`. Beispiel: Geben Sie `/images` an, um Dateien unter *`subdomain`* `.elasticbeanstalk.com/images` bereitzustellen.	Der Name des Verzeichnisses, das die Dateien enthält. Beispiel: Geben Sie `staticimages` an, um Dateien aus einem Verzeichnis mit dem Namen `staticimages` auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

aws:elasticbeanstalk:healthreporting:system

Konfigurieren Sie erweiterte Zustandsberichte für die Umgebung.

Namespace: **`aws:elasticbeanstalk:healthreporting:system`**

Name	Beschreibung	Standard	Zulässige Werte

SystemType	Das System für Zustandsberichte (grundlegend oder erweitert). Erweiterte Zustandsberichte erfordern eine Servicerolle und eine Plattformversion mit Version 2 oder höher.	`basic`	`basic`
			
			`enhanced`

Name	Beschreibung	Standard	Zulässige Werte
	Note Wenn Sie die Elastic-Beanstalk-Konsole oder EB-CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.		
ConfigDocument	Ein JSON-Dokument, das die Umgebungs- und Instanzmetriken beschreibt, in denen veröffentlicht werden soll. CloudWatch	Keine	
EnhancedHealthAuthEnabled	Aktiviert die Autorisierung für die interne API, die von Elastic Beanstalk für die Übermittlung erweiterter Integritätsinformationen aus Ihren Umgebungs-Instances an den Elastic-Beanstalk-Service verwendet wird. Weitere Informationen finden Sie unter the section called “Rollen in erweiterten Zustandsberichten”. **Note** Diese Option kann nur auf erweiterte Integritätsberichte angewendet werden (z. B. bei Festlegung von `SystemType` auf `enhanced`).	`true`	`true` `false`

Name	Beschreibung	Standard	Zulässige Werte

HealthCheckSuccessThreshold	Setzt den Grenzwert herab, sodass Instances die Zustandsprüfungen bestehen.
	
	Note
	Wenn Sie die Elastic-Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert.
		`Ok`	`Ok`
			
			`Warning`
			
			`Degraded`
			
			`Severe`

aws:elasticbeanstalk:hostmanager

Konfigurieren Sie die EC2-Instances der Umgebung für den Upload von rotierten Protokollen in Amazon-S3.

Namespace: **`aws:elasticbeanstalk:hostmanager`**

Name	Beschreibung	Standard	Zulässige Werte

LogPublicationControl	Kopieren Sie die Protokolldateien der Amazon-EC2-Instances Ihrer Anwendung in den Amazon-S3-Bucket dieser Anwendung.	`false`	`true`
			
			`false`

aws:elasticbeanstalk:managedactions

Konfigurieren Sie verwaltete Plattformaktualisierungen für die Umgebung.

Namespace: **`aws:elasticbeanstalk:managedactions`**

Name	Beschreibung	Standard	Zulässige Werte

ManagedActionsEnabled	Aktivieren Sie verwaltete Plattformaktualisierungen.
	
	Wenn Sie diesen Parameter auf `true` setzen, müssen Sie auch einen Wert für `PreferredStartTime` und `UpdateLevel` angeben.
		`false`	`true`
			
			`false`

PreferredStartTime	Konfigurieren Sie ein Wartungsfenster für verwaltete Aktionen in UTC (koordinierter Weltzeit). Beispiel, `"Tue:09:00"` .	Keine	Tag und Uhrzeit im Format *`day`*:*`hour`*:*`minute`* .
ServiceRoleForManagedUpdates	Der Name einer IAM-Rolle, die Elastic Beanstalk verwendet , um verwaltete Plattform-Updates für Ihre Umgebung durchzuführen. Sie können entweder dieselbe Rolle verwenden, die Sie für die `ServiceRole` -Option des `aws:elasticbeanstalk:environment` -Namespaces angegeben haben, oder die serviceverknüpfte Rolle für verwaltete Updates Ihres Kontos. Wenn im zweiten Fall das Konto noch nicht über eine serviceve	Keine	Identisch mit `ServiceRole` oder `AWSServiceRoleForElasticBeanstalkManagedUpdates`

Name	Beschreibung	Standard	Zulässige Werte
	rknüpfte Rolle für verwaltete Updates verfügt, wird sie von Elastic Beanstalk erstellt.		

aws:elasticbeanstalk:managedactions:platformupdate

Konfigurieren Sie verwaltete Plattformaktualisierungen für die Umgebung.

Namespace: **`aws:elasticbeanstalk:managedactions:platformupdate`**

Name	Beschreibung	Standard	Zulässige Werte
UpdateLevel	Die höchste Aktualisierungsebene, die mit verwalteten Plattformaktualisierungen möglich ist. Plattformen sind versioniert. *`major`* *`minor`*. *`patch`*. Beispielsweise gibt die Plattformversion 2.0.8 die Hauptversion 2, die Nebenversion 0 und die Patch-Version 8 an.	Keine	`patch` nur für Patch-Versionsaktualisierungen. `minor` für sowohl Neben- als auch Patch-Versionsaktualisierungen.

InstanceRefreshEnabled	Damit aktivieren Sie den wöchentlichen Instance-Ersatz.
	
	Der Parameter `ManagedActionsEnabled` muss auf den Wert `true` festgelegt sein.
		`false`	`true`
			
			`false`

aws:elasticbeanstalk:monitoring

Konfigurieren Sie die Umgebung so, dass EC2-Instances mit nicht bestandenen Zustandsprüfungen beendet werden.

Namespace: **`aws:elasticbeanstalk:monitoring`**

Name	Beschreibung	Standard	Zulässige Werte

Automatically Terminate Unhealthy Instances	Wenn eine Instance die Zustandsprüfungen nicht besteht, wird sie beendet.
	
	Note
	Diese Option wurde nur in Legacy-Umgebungen unterstützt. Sie bestimmt den Zustand einer Instance basierend auf ihrer Erreichbarkeit und auf anderen Instance-basierten Metriken. Elastic Beanstalk bietet keine Möglichkeit zum automatischen Beenden von Instances anhand des Anwendungszustands.
		`true`	`true`
			
			`false`

aws:elasticbeanstalk:sns:topics

Konfigurieren Sie Benachrichtigungen für die Umgebung.

Namespace: **`aws:elasticbeanstalk:sns:topics`**

Name	Beschreibung	Standard	Zulässige Werte
Notification Endpoint	Gibt den Endpunkt an, über den Sie Benachrichtigungen zu wichtigen Ereignissen für die Anwendung erhalten möchten. **Note** Wenn Sie die Elastic-Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert.	Keine	

Notification Protocol	Gibt das Protokoll an, über das Benachrichtigungen an Ihren Endpunkt gesendet werden.	`email`	`http`
			
			`https`
			
			`email`
			
			`email-json`
			
			`sqs`

Notification Topic ARN	Der Amazon-Ressourcenname (ARN) für das abonnierte Thema.	Keine	
Notification Topic Name	Der Name des abonnierten Themas.	Keine	

aws:elasticbeanstalk:sqsd

Konfigurieren Sie die Amazon-SQS-Warteschlange für eine Worker-Umgebung.

Namespace: **`aws:elasticbeanstalk:sqsd`**

Name	Beschreibung	Standard	Zulässige Werte
WorkerQueueURL	Die URL der Warteschlange, aus der Nachrichten vom Daemon der Worker-Umgebungsebene gelesen werden. **Note** Wenn Sie keinen Wert angeben, handelt es sich bei der Warteschlange, die Elastic Beanstalk automatisch erstellt, um eine Standard-Amazon-SQS-Warteschlange. Wenn Sie einen Wert angeben, können Sie die URL einer Standard- oder einer FIFO-Amazon-SQS-Warteschlange angeben. Beachten Sie, dass bei Angabe einer FIFO-Warteschlange regelmäßige Aufgaben nicht unterstützt werden.	Wird automatisch generiert.	Wenn Sie keinen Wert angeben, erstellt Elastic Beanstalk automatisch eine Warteschlange.
HttpPath	Gibt den relativen Pfad zur Anwendung an, an die HTTP POST-Nachrichten gesendet werden.	/	

Name	Beschreibung	Standard	Zulässige Werte

MimeType	Gibt den MIME-Typ der Nachricht an, die mit der HTTP POST-Anforderung gesendet wird.	`application/json`	`application/json`
			
			`application/x-www-form-urlencoded`
			
			`application/xml`
			
			`text/plain`
			
			Benutzerdefinierter MIME-Typ.

HttpConnections	Gibt die maximale Anzahl gleichzeitiger Verbindungen zu den Anwendungen auf einer Amazon-EC2-Instance an. **Note** Wenn Sie die Elastic-Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert.	`50`	1 auf `100`
ConnectTimeout	Gibt die Zeitspanne (in Sekunden) an, die auf erfolgreiche Verbindungserstellungen zu einer Anwendung gewartet wird.	`5`	1 auf `60`

Name	Beschreibung	Standard	Zulässige Werte
InactivityTimeout	Gibt die Zeitspanne (in Sekunden) an, die bei einer bestehenden Verbindung zu einer Anwendung auf eine Antwort gewartet wird. Die Nachricht wird solange erneut verarbeitet, bis der Daemon die Antwort "200 (OK)" von der Anwendung der Worker-Umgebungsebene erhält oder der `RetentionPeriod` -Wert abgelaufen ist.	299	1 auf 36000
VisibilityTimeout	Gibt die Zeitspanne (in Sekunden) an, die eine eingehende Nachricht aus der Amazon-SQS-Warteschlange zur Verarbeitung gesperrt ist. Nach Ablauf des konfigurierten Zeitraums wird die Nachricht in der Warteschlange wieder sichtbar und kann von einem anderen Daemon gelesen werden.	300	0 auf 43200
ErrorVisibilityTimeout	Gibt die Zeitspanne (in Sekunden) an, die verstreichen muss, bevor Elastic Beanstalk eine Nachricht nach einem fehlgeschlagenen Verarbeitungsversuch mit einem expliziten Fehler an die Amazon-SQS-Warteschlange zurückgibt.	2 Sekunden	0 bis 43200 Sekunden
RetentionPeriod	Gibt die Zeitspanne (in Sekunden) an, die eine Nachricht gültig ist und aktiv verarbeitet wird.	345600	60 auf 1209600

Name	Beschreibung	Standard	Zulässige Werte
MaxRetries	Gibt die maximale Anzahl der Wiederholungsversuche an, mit denen Elastic Beanstalk versuchen kann, die Nachricht zur Verarbeitung an die entsprechende Web-Anwendung zu senden, bevor die Nachricht in die Warteschlange für unzustellbare Nachrichten verschoben wird.	`10`	1 auf `100`

aws:elasticbeanstalk:trafficsplitting

Konfigurieren Sie Bereitstellungen für die Datenverkehrsaufteilung für Ihre Umgebung.

Dieser Namespace wird angewendet, wenn Sie die `DeploymentPolicy`-Option des Namespace aws:elasticbeanstalk:command auf `TrafficSplitting` festlegen. Weitere Informationen zu Bereitstellungsrichtlinien finden Sie unter the section called “Optionen für die Bereitstellung”.

Namespace: **`aws:elasticbeanstalk:trafficsplitting`**

Name	Beschreibung	Standard	Zulässige Werte
NewVersionPercent	Der anfängliche Prozentsatz des eingehenden Clientdatenverkehrs, den Elastic Beanstalk auf Umgebungs-Instances verlagert, auf denen die von Ihnen bereitgestellte neue Anwendungsversion ausgeführt wird.	`10`	1 auf `100`
EvaluationTime	Der Zeitraum (in Minuten), den Elastic Beanstalk nach einer anfänglich fehlerfreien Bereitstellung wartet, bevor der gesamte eingehende Clientdatenverkehr auf die von Ihnen bereitgestellte neue	`5`	`3` auf `600`

Name	Beschreibung	Standard	Zulässige Werte
	Anwendungsversion verschoben wird.		

aws:elasticbeanstalk:xray

Führen Sie den AWS X-Ray Daemon aus, um Trace-Informationen aus Ihrer integrierten X-Ray-Anwendung weiterzuleiten.

Namespace: **`aws:elasticbeanstalk:xray`**

Name	Beschreibung	Standard	Zulässige Werte
`XRayEnabled`	Auf `true` setzen, um den X-Ray-Daemon auf den Instances in Ihrer Umgebung auszuführen.	`false`	`true` `false`

aws:elb:healthcheck

Konfigurieren Sie Zustandsprüfungen für den Classic Load Balancer.

Namespace: **`aws:elb:healthcheck`**

Name	Beschreibung	Standard	Zulässige Werte
HealthyThreshold	Die Anzahl aufeinanderfolgender erfolgreicher Anforderungen vor Elastic-Load-Balancing ändert den Instance-Zustand.	`3`	2 auf `10`
Interval	Das Intervall, in dem Elastic-Load-Balancing den Zustand der Amazon-EC2-Instances der Anwendung überprüft.	`10`	5 auf `300`
Timeout	Die Zeitspanne in Sekunden, die Elastic-Load-Balancing auf eine Antwort wartet, bevor die Instance als nicht reagierend eingestuft wird.	`5`	2 auf `60`

Name	Beschreibung	Standard	Zulässige Werte
Unhealthy Threshold	Die Anzahl aufeinanderfolgender erfolgloser Anforderungen vor Elastic-Load-Balancing ändert den Instance-Zustand.	`5`	`2` auf `10`
(veraltet) Target	Das Ziel einer Backend-Instance, an das Zustandsprüfungen gesendet werden. Verwenden Sie stattdessen `Application Healthcheck URL` im Namespace `aws:elasticbeanstalk:application`.	`TCP:80`	Ziel im Format *`PROTOCOL`*`:`*`PORT`*`/`*`PATH`

aws:elb:loadbalancer

Konfigurieren Sie den Classic Load Balancer der Umgebung.

Mehrere der Optionen in diesem Namespace werden zugunsten von listenerspezifischen Optionen im aws:elb:listener-Namespace nicht mehr unterstützt. Mit diesen nicht mehr unterstützten Optionen können nur zwei Listener (einen sicheren und einen nicht sicheren) auf Standard-Ports konfiguriert werden.

Namespace: **`aws:elb:loadbalancer`**

Name	Beschreibung	Standard	Zulässige Werte

CrossZone	Konfigurieren Sie den Load Balancer so, dass der Datenverkehr gleichmäßig auf alle Instances in allen Availability Zones (und nicht nur in den einzelnen Zonen) weitergeleitet wird.
	
	Note
	Wenn Sie die Elastic-Beanstalk-Konsole oder EB-CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in
		`false`	`true`
			
			`false`

Name	Beschreibung	Standard	Zulässige Werte
	einer Konfigurationsdatei festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.		
SecurityGroups	Ordnen Sie dem Load Balancer eine oder mehrere der von Ihnen erstellten Sicherheitsgruppen zu. Erforderlich, wenn `DisableDefaultEC2SecurityGroup` (aws:autoscaling:launchconfiguration) auf gesetzt ist `true`. Umgebungen mit Lastenausgleich, die sich von der standardmäßigen Elastic Beanstalk EC2-Sicherheitsgruppe abgemeldet haben, müssen diese Option für eine oder mehrere Sicherheitsgruppen bereitstellen. Weitere Informationen finden Sie unter Verwaltung von EC2-Sicherheitsgruppen.	Keine	Eine oder mehrere Sicherheitsgruppen. IDs

Name	Beschreibung	Standard	Zulässige Werte
ManagedSecurityGroup	Ordnen Sie dem Load Balancer der Umgebung eine vorhandene Sicherheitsgruppe zu, anstatt eine neue zu erstellen. Wenn Sie diese Einstellung nutzen möchten, müssen Sie die Einstellung `SecurityGroups` in diesem Namespace aktualisieren, damit die ID Ihrer Sicherheitsgruppe einbezogen wird, und die ID der automatisch generierten Sicherheitsgruppe löschen (falls eine generiert wurde). Um den Datenverkehr vom Load Balancer zu den EC2-Instances zu ermöglichen, fügt Elastic Beanstalk eine Regel, die eingehenden Datenverkehr von der verwalteten Sicherheitsgruppe zulässt, zur Sicherheitsgruppe der Instance hinzu.	Keine	Eine Sicherheitsgruppen-ID.

(veraltet) LoadBalancerHTTPPort	Der zu überwachende Port für den nicht sicheren Listener.	`80`	`OFF`
			
			`80`

(veraltet) LoadBalancerPortProtocol	Das zu verwendende Protokoll für den nicht sicheren Listener.	`HTTP`	`HTTP`
			
			`TCP`

(veraltet) LoadBalancerHTTPSPort	Den zu überwachenden Port für den sicheren Listener.	`OFF`	`OFF`
			
			`443`
			
			`8443`

(veraltet) LoadBalancerSSLPortProtocol	Das zu verwendende Protokoll für den sicheren Listener.	`HTTPS`	`HTTPS`
			
			`SSL`

Name	Beschreibung	Standard	Zulässige Werte
(veraltet) SSLCertificateId	Der Amazon-Ressourcenname (ARN) eines SSL-Zertifikats, das an den sicheren Listener gebunden werden soll.	Keine	

aws:elb:listener

Konfigurieren Sie den Standard-Listener (Port 80) für einen Classic Load Balancer.

Namespace: **`aws:elb:listener`**

Name	Beschreibung	Standard	Zulässige Werte
ListenerProtocol	Gibt das vom Listener verwendete Protokoll an.	`HTTP`	`HTTP` `TCP`
InstancePort	Port, der von diesem Listener für die Kommunikation mit den EC2-Instances verwendet wird.	80	1 auf 65535
InstanceProtocol	Protokoll, das von diesem Listener für die Kommunikation mit den EC2-Instances verwendet wird. Die Internetprotokoll-Ebene muss mit übereinstimmen `ListenerProtocol` . Außerdem muss die Sicherheitsebene mit allen anderen Listenern identisch sein, die denselben `InstancePort` nutzen wie dieser Listener. Wenn `ListenerProtocol` beispielsweise auf `HTTPS` (Anwendungsebene, über eine sichere Verbindung) festgelegt ist, können Sie `InstanceProtocol` auf	`HTTP`, wenn `ListenerProtocol` den Wert `HTTP` hat. `TCP`, wenn `ListenerProtocol` den Wert `TCP` hat.	`HTTP` oder `HTTPS`, wenn `ListenerProtocol` den Wert `HTTP` oder `HTTPS` hat. `TCP` oder `SSL`, wenn `ListenerProtocol` den Wert

Name	Beschreibung	Standard	Zulässige Werte
	HTTP (ebenfalls Anwendungsebene, aber unsichere Verbindung) einstellen. Falls Sie zudem `InstancePort` auf `80` festgelegt haben, müssen Sie `InstanceProtocol` in allen anderen Listenern, bei denen HTTP auf `InstancePort` festgelegt ist, auf `80` einstellen.		TCP oder SSL hat.
PolicyNames	Eine durch Komma getrennte Liste mit Richtliniennamen, die für den Port dieses Listeners gelten. Wir empfehlen, stattdessen die LoadBalancerPorts Option des aws:elb:policies Namespaces zu verwenden.	Keine	
ListenerEnabled	Gibt an, ob dieser Listener aktiviert ist. Wenn Sie den Wert `false` festlegen, wird der Listener nicht in den Load Balancer einbezogen.	`true`	`true` `false`

aws:elb:listener:listener_port

Konfigurieren Sie zusätzliche Listener für einen Classic Load Balancer.

Namespace: **`aws:elb:listener:`*`listener_port`***

Name	Beschreibung	Standard	Zulässige Werte
ListenerProtocol	Gibt das vom Listener verwendete Protokoll an.	`HTTP`	`HTTP` `HTTPS` `TCP` `SSL`

Name	Beschreibung	Standard	Zulässige Werte
InstancePort	Port, der von diesem Listener für die Kommunikation mit den EC2-Instances verwendet wird.	Das Gleiche wie *listener port* .	1 auf 65535
InstanceProtocol	Protokoll, das von diesem Listener für die Kommunikation mit den EC2-Instances verwendet wird. Die Internetprotokoll-Ebene muss mit übereinstimmen `ListenerProtocol` . Außerdem muss die Sicherheitsebene mit allen anderen Listenern identisch sein, die denselben `InstancePort` nutzen wie dieser Listener. Wenn `ListenerProtocol` beispielsweise auf `HTTPS` (Anwendungsebene, über eine sichere Verbindung) festgelegt ist, können Sie `InstanceProtocol` auf `HTTP` (ebenfalls Anwendungsebene, aber unsichere Verbindung) einstellen. Falls Sie zudem `InstancePort` auf `80` festgelegt haben, müssen Sie `InstanceProtocol` in allen anderen Listenern, bei denen `HTTP` auf `InstancePort` festgelegt ist, auf `80` einstellen.	`HTTP`, wenn `ListenerProtocol` den Wert `HTTP` oder `HTTPS` hat. `TCP`, wenn `ListenerProtocol` den Wert `TCP` oder `SSL` hat.	`HTTP` oder `HTTPS`, wenn `ListenerProtocol` den Wert `HTTP` oder `HTTPS` hat. `TCP` oder `SSL`, wenn `ListenerProtocol` den Wert `TCP` oder `SSL` hat.
PolicyNames	Eine durch Komma getrennte Liste mit Richtliniennamen, die für den Port dieses Listeners gelten. Wir empfehlen Ihnen, stattdessen die LoadBalancerPorts Option des aws:elb:policies Namespaces zu verwenden.	Keine	

Name	Beschreibung	Standard	Zulässige Werte
SSLCertificateId	Der Amazon-Ressourcenname (ARN) eines SSL-Zertifikats, das an den Listener gebunden werden soll.	Keine	
ListenerEnabled	Gibt an, ob dieser Listener aktiviert ist. Wenn Sie den Wert `false` festlegen, wird der Listener nicht in den Load Balancer einbezogen.	`true`, wenn eine andere Option festgelegt ist; ansonsten `false`.	`true` `false`

aws:elb:policies

Ändern Sie bei einem Classic Load Balancer die Sticky Sessions-Standardrichtlinien sowie die globalen Load-Balancer-Richtlinien.

Namespace: **`aws:elb:policies`**

Name	Beschreibung	Standard	Zulässige Werte

ConnectionDrainingEnabled	Gibt an, ob der Load Balancer bestehende Verbindungen zu instabil gewordenen oder abgemeldeten Instances halten soll, um laufende Anforderungen abzuschließen.
	
	Note
	Wenn Sie die Elastic-Beanstalk-Konsole oder EB-CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei
		`false`	`true`
			
			`false`

Name	Beschreibung	Standard	Zulässige Werte
	festlegen. Die Konsole und EB-CLI überschreiben diese Option mit einem empfohlenen Wert.		
ConnectionDraining Timeout	Gibt an, wie viele Sekunden der Load Balancer beim Connection Draining bestehende Verbindungen zu einer Instance maximal halten soll, bevor das Beenden der Verbindungen erzwungen wird. **Note** Wenn Sie die Elastic-Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert.	`20`	1 auf `3600`
ConnectionSettingIdleTimeout	Die Anzahl der Sekunden, die der Load Balancer wartet, dass Daten über die Verbindung gesendet oder empfangen werden. Wenn nach Verstreichen dieses Zeitraums keine Daten versendet oder empfangen wurden, schließt der Load Balancer die Verbindung.	`60`	1 auf `3600`

Name	Beschreibung	Standard	Zulässige Werte
LoadBalancerPorts	Eine durch Komma getrennte Liste der Listener-Ports, für die die Standardr ichtlinie (`AWSEB-ELB-StickinessPolicy`) gilt.	Keine	Mit `:all` können Sie alle Listener-Ports angeben.
Stickiness Cookie Expiration	Gibt die Zeitspanne (in Sekunden) an, die ein Cookie gültig ist. Von diesem Parameter wird die Standardrichtlinie () verwendet (`AWSEB-ELB-StickinessPolicy`).	`0`	`0` auf `1000000`
Stickiness Policy	Damit wird eine Benutzersitzung an eine bestimmte Server-Instance gebunden, sodass alle vom Benutzer während der Sitzung gesendeten Anforderungen an die gleiche Server-Instance übermittelt werden. Von diesem Parameter wird die Standardrichtlinie () verwendet (`AWSEB-ELB-StickinessPolicy`).	`false`	`true false`

aws:elb:policies:policy_name

Erstellen Sie zusätzliche Load Balancer-Richtlinien für einen Classic Load Balancer.

Namespace: **`aws:elb:policies:`*`policy_name`***

Name	Beschreibung	Standard	Zulässige Werte
CookieName	Der Name des (von der Anwendung generierten) Cookies, mit dem die Sitzungsgültigkeitsdauer einer `AppCookie StickinessPolicyType` -Richtlinie gesteuert wird. Diese Richtlinie kann	Keine	

Name	Beschreibung	Standard	Zulässige Werte
	nur HTTP/HTTPS Listenern zugeordnet werden.		
InstancePorts	Eine durch Komma getrennte Liste der Instance-Ports, für die diese Richtlinie gilt.	Keine	Eine Liste mit Ports oder `:all`
LoadBalancerPorts	Eine durch Komma getrennte Liste der Listener-Ports, für die diese Richtlinie gilt.	Keine	Eine Liste mit Ports oder `:all`
ProxyProtocol	Gibt bei einer `ProxyProtocolPolicyType` -Richtlinie an, ob die IP-Adresse und der Port der ursprünglichen Anforderung in TCP-Nachrichten eingebunden werden sollen. Diese Richtlinie kann nur TCP/SSL Listenern zugeordnet werden.	Keine	`true false`
PublicKey	Der Inhalt eines öffentlichen Schlüssels, der von einer `PublicKeyPolicyType` -Richtlinie zur Authentifizierung der Backend-Server oder der Server verwendet wird. Diese Richtlinie kann nicht direkt für Backend-Server oder Listener eingesetzt werden, sondern muss Teil einer `BackendServerAuthenticationPolicyType` -Richtlinie sein.	Keine	

Name	Beschreibung	Standard	Zulässige Werte
PublicKeyPolicyNames	Eine durch Komma getrennte Liste der Richtliniennamen (aus den `PublicKeyPolicyType` -Richtlinien) für eine `BackendServerAuthenticationPolicyType` -Richtlinie, mit der die Authentifizierung am Backend-Server oder an den Servern gesteuert wird. Diese Richtlinie kann nur mit Backend-Servern verknüpft werden, die HTTPS/SSL verwenden.	Keine	
SSLProtocols	Eine durch Komma getrennte Liste der SSL-Protokolle, die für eine `SSLNegotiationPolicyType` -Richtlinie aktiviert werden sollen, in der die vom Load Balancer akzeptierten Verschlüsselungsverfahren und Protokolle definiert werden. Diese Richtlinie kann nur HTTPS/SSL Listenern zugeordnet werden.	Keine	
SSLReferencePolicy	Der Name einer vordefinierten Sicherheitsrichtlinie, die den bewährten AWS Sicherheitsmethoden entspricht und die Sie für eine `SSLNegotiationPolicyType` Richtlinie aktivieren möchten, die die Verschlüsselungen und Protokolle definiert, die vom Load Balancer akzeptiert werden. Diese Richtlinie kann nur Listenern zugeordnet werden. HTTPS/SSL	Keine	
Stickiness Cookie Expiration	Gibt die Zeitspanne (in Sekunden) an, die ein Cookie gültig ist.	`0`	`0` auf `1000000`

Name	Beschreibung	Standard	Zulässige Werte
Stickiness Policy	Damit wird eine Benutzersitzung an eine bestimmte Server-Instance gebunden, sodass alle vom Benutzer während der Sitzung gesendeten Anforderungen an die gleiche Server-Instance übermittelt werden.	`false`	`true false`

aws:elbv2:listener:default

Konfigurieren Sie den Standard-Listener (Port 80) auf einem Application Load Balancer oder einem Network Load Balancer.

Dieser Namespace gilt nicht für eine Umgebung, die einen freigegebenen Load Balancer verwendet. Freigegebene Load Balancer besitzen keinen Standard-Listener.

Namespace: **`aws:elbv2:listener:default`**

Name	Beschreibung	Standard	Zulässige Werte
DefaultProcess	Der Name des Prozesses, an den der Datenverkehr weitergeleitet wird, falls keine Regel zutrifft.	`default`	Ein Prozessname.

ListenerEnabled	Legen Sie den Wert auf `false` fest, um den Listener zu deaktivieren. Mit dieser Option können Sie den Standard-Listener auf Port 80 deaktivieren.	`true`	`true`
			
			`false`

Name	Beschreibung	Standard	Zulässige Werte
Protocol	Das Protokoll des zu verarbeitenden Datenverkehrs.	Bei Application Load Balancer: `HTTP` Bei Network Load Balancer: `TCP`	Bei Application Load Balancer: `HTTP`, `HTTPS` Bei Network Load Balancer: `TCP`
Rules	Eine Liste der Regeln, die für diesen Listener gelten sollen. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	Keine	Eine durch Komma getrennte Liste der Regelnamen.
SSLCertificateArns	Der Amazon-Ressourcenname (ARN) des SSL-Zertifikats, das an den Listener gebunden werden soll. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	Keine	Der ARN eines in IAM oder ACM gespeicherten Zertifikats.

Name	Beschreibung	Standard	Zulässige Werte
SSLPolicy	Geben Sie eine Sicherheitsrichtlinie an, die für den Listener gelten soll. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	Keine (ELB-Standard)	Der Name einer Load Balancer-Sicherheitsrichtlinie.

aws:elbv2:listener:listener_port

Konfigurieren Sie zusätzliche Listener auf einem Application Load Balancer oder einem Network Load Balancer.

> **Note**
>
> Für einen gemeinsam genutzten Application Load Balancer können Sie nur die `Rule` Option angeben. Die übrigen Optionen können nicht auf freigegebene Load Balancer angewendet werden.

Namespace: **`aws:elbv2:listener:`*`listener_port`***

Name	Beschreibung	Standard	Zulässige Werte
DefaultProcess	Der Name des Prozesses, an den der Datenverkehr weitergeleitet wird, falls keine Regel zutrifft.	`default`	Ein Prozessname.
ListenerEnabled	Legen Sie den Wert auf `false` fest, um	`true`	`true`

Name	Beschreibung	Standard	Zulässige Werte
	den Listener zu deaktivieren. Mit dieser Option können Sie den Standard-Listener auf Port 80 deaktivieren.		`false`
Protocol	Das Protokoll des zu verarbeitenden Datenverkehrs.	Bei Application Load Balancer: `HTTP` Bei Network Load Balancer: `TCP`	Bei Application Load Balancer: `HTTP`, `HTTPS` Bei Network Load Balancer: `TCP`
Rules	Liste der Regeln, die für diesen Listener gelten sollen. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar. Wenn Ihre Umgebung einen freigegebenen Application Load Balancer verwendet und Sie diese Option für keinen Listener angeben, ordnet Elastic Beanstalk die Regel `default` automatisch einem Port 80-Listener zu.	Keine	Eine durch Komma getrennte Liste der Regelnamen.

Name	Beschreibung	Standard	Zulässige Werte
SSLCertificateArns	Der Amazon-Ressourcenname (ARN) des SSL-Zertifikats, das an den Listener gebunden werden soll. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	Keine	Der ARN eines in IAM oder ACM gespeicherten Zertifikats.
SSLPolicy	Geben Sie eine Sicherheitsrichtlinie an, die für den Listener gelten soll. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	Keine (ELB-Standard)	Der Name einer Load-Balancer-Sicherheitsrichtlinie.

aws:elbv2:listenerrule:rule_name

Definieren Sie Listener-Regeln für einen Application Load Balancer. Wenn eine Anforderung dem Host-Namen oder Pfaden in einer Regel entspricht, leitet der Load Balancer sie an den angegebenen Prozess weiter. Wenn Sie eine Regel verwenden möchten, fügen Sie sie dem Listener mit der Option `Rules` im `aws:elbv2:listener:listener_port`-Namespace zu.

> **Note**
>
> Dieser Namespace ist nicht auf Umgebungen mit einem Network Load Balancer anwendbar.

Namespace: **`aws:elbv2:listenerrule:`*`rule_name`***

Name	Beschreibung	Standard	Zulässige Werte
HostHeaders	Eine Liste mit den zuzuordnenden Host-Namen. Beispiel, `my.example.com` .	Dedizierter Load Balancer: Keiner Freigegebener Load Balancer: der CNAME der Umgebung	Jeder Name kann bis zu 128 Zeichen umfassen. Ein Muster kann sowohl Groß- als auch Kleinbuchstaben, Zahlen, Bindestriche (–) und bis zu drei Platzhalterzeichen enthalten (* entspricht null oder mehr Zeichen; ? entspricht genau einem Zeichen). Sie können mehrere Namen angeben, die jeweils durch ein Komma getrennt sind. Application Load Balancer unterstützt bis zu fünf kombinierte `HostHeader` - und `PathPattern` -Regeln. Weitere Informationen finden Sie unter Hostbedingungen im Benutzerhandbuch für Application Load Balancers.
PathPatterns	Die Pfadmuster sollten übereinstimmen (z. B. `/img/*`).	Keine	Jedes Muster kann bis zu 128 Zeichen enthalten. Ein Muster

Name	Beschreibung	Standard	Zulässige Werte
	Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar .		kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (–) und bis zu drei Platzhalterzeichen enthalten (* entspricht null oder mehr Zeichen; ? entspricht genau einem Zeichen). Sie können mehrere durch Komma getrennte Pfadmuster hinzufügen. Application Load Balancer unterstützt bis zu fünf kombinierte `HostHeader` - und `PathPattern` -Regeln. Weitere Informationen finden Sie unter Pfadbedingungen im Benutzerhandbuch für Application Load Balancers.

Name	Beschreibung	Standard	Zulässige Werte
Priority	Die Priorität dieser Regel, wenn mehrere Regeln zutreffen. Die niedrigere Zahl hat Vorrang. Zwei Regeln dürfen nicht dieselbe Priorität haben. Im Fall eines freigegebenen Load Balancer behandelt Elastic Beanstalk Regelprioritäten in freigebenden Umgebungen als relativ und ordnet sie während der Erstellung absoluten Prioritäten zu.	1	1 auf `1000`
Process	Der Name des Prozesses, an den der Datenverkehr weitergeleitet wird, wenn diese Regel auf die Anforderung zutrifft.	`default`	Ein Prozessname.

aws:elbv2:loadbalancer

Konfigurieren Sie einen Application Load Balancer.

Im Fall eines freigegebenen Load Balancer sind nur die Optionen `SecurityGroups` und `SharedLoadBalancer` gültig.

> **Note**
>
> Dieser Namespace ist nicht auf Umgebungen mit einem Network Load Balancer anwendbar.

Namespace: **`aws:elbv2:loadbalancer`**

Name	Beschreibung	Standard	Zulässige Werte
AccessLogsS3Bucket	Amazon-S3-Bucket, in dem Zugriffsprotokolle gespeichert werden. Der Bucket muss in derselben Region sein wie die Umgebung und	Keine	Ein Bucket-Name.

Name	Beschreibung	Standard	Zulässige Werte
	Schreibzugriff für den Load Balancer gewähren.		
AccessLogsS3Enabled	Damit aktivieren Sie die Speicherung von Zugriffsprotokollen.	`false`	`true` `false`
AccessLogsS3Prefix	Ein Präfix wird den Zugriffsprotokollnamen vorangestellt. Standardmäßig lädt der Load Balancer Protokolle in ein Verzeichnis hoch, das AWSLogs in dem von Ihnen angegebenen Bucket benannt ist. Geben Sie ein Präfix an, um das AWSLogs Verzeichnis in einem anderen Verzeichnis zu platzieren.	Keine	
IdleTimeout	Die Zeit in Sekunden, die auf den Abschluss einer Anforderung gewartet wird, bevor die Verbindungen zu Client und Instance beendet werden.	Keine	1 auf `3600`

Name	Beschreibung	Standard	Zulässige Werte
IpAddressType	Gibt die Konfiguration des IP-Adressformats für den Load Balancer der Umgebung an. Verwenden Sie die DualStack-Option, um das Protokoll zusammen mit dem IPv6 Protokoll zu aktivieren. IPv4 Nur Application Load Balancers und Network Load Balancer unterstützen die Dualstack-Option. Einzelinstanzumgebungen und Umgebungen, die Classic Load Balancers verwenden, unterstützen sie nicht. Sie müssen die VPC der Umgebung und alle VPC-Subnetze mit IPv6 CIDR-Blöcken verknüpfen, um die Dualstack-Option zu unterstützen. Weitere Informationen finden Sie unter the section called “Konfiguration von Dual-Stack”.	IPv4	IPv4, Dualstack

Name	Beschreibung	Standard	Zulässige Werte
ManagedSecurityGroup	Ordnen Sie dem Load Balancer der Umgebung eine vorhandene Sicherheitsgruppe zu, anstatt eine neue zu erstellen. Wenn Sie diese Einstellung nutzen möchten, müssen Sie die Einstellung `SecurityGroups` in diesem Namespace aktualisieren, damit die ID Ihrer Sicherheitsgruppe einbezogen und die ID der automatisch generierten Sicherheitsgruppe (sofern vorhanden) gelöscht wird. Um den Datenverkehr vom Load Balancer zu den EC2-Instances zu ermöglichen, fügt Elastic Beanstalk eine Regel, die eingehenden Datenverkehr von der verwalteten Sicherheitsgruppe zulässt, zur Sicherheitsgruppe der Instance hinzu.	Die Sicherheitsgruppe die Elastic Beanstalk für den Load Balancer erstellt.	Eine Sicherheitsgruppen-ID.

Name	Beschreibung	Standard	Zulässige Werte
SecurityGroups	Eine Liste der Sicherheitsgruppen, die dem Load Balancer zugeordnet werden sollen. Erforderlich, wenn `DisableDefaultEC2SecurityGroup` (aws:autoscaling:launchconfiguration) auf gesetzt ist. `true` Umgebungen mit Lastenausgleich, die sich von der standardmäßigen Elastic Beanstalk EC2-Sicherheitsgruppe abgemeldet haben, müssen diese Option für eine oder mehrere Sicherheitsgruppen bereitstellen. Weitere Informationen finden Sie unter Verwaltung von EC2-Sicherheitsgruppen. Wenn Sie diesen Wert für einen freigegebenen Load Balancer nicht angeben, prüft Elastic Beanstalk, ob bereits eine vorhandene Sicherheitsgruppe, die von ihm verwaltet wird, dem Load Balancer angefügt ist. Wenn keine Verbindung zum Load Balancer besteht, erstellt Elastic Beanstalk eine Sicherheitsgruppe und hängt diese an den Load Balancer an. Elastic Beanstalk löscht diese Sicherheitsgruppe, wenn die letzte Umgebung, die den Load Balancer teilt, beendet wird. Die Load Balancer-Sicherheitsgruppen werden verwendet, um	Die Sicherheitsgruppe die Elastic Beanstal für den Load Balancer erstellt.	Durch Kommas getrennte Liste der Sicherheitsgruppen. IDs

Name	Beschreibung	Standard	Zulässige Werte
	die Regel für den eingehenden Datenverkehr für die Amazon-EC2-Instance-Sicherheitsgruppe einzurichten.		

Name	Beschreibung	Standard	Zulässige Werte
SharedLoadBalancer	Der Amazon-Ressourcenname (ARN) eines freigegebenen Load Balancers. Diese Option ist nur für einen Application Load Balancer relevant. Sie muss festgelegt werden, wenn die Option `LoadBalancerIsShared` des Namespace aws:elasticbeanstalk:environment auf `true` festgelegt ist. Sie können den ARN des freigegebenen Load Balancer nach Erstellung der Umgebung nicht ändern. Kriterien für einen gültigen Wert: • Es muss sich um einen gültigen, aktiven Load Balancer in der AWS Region handeln, in der sich die Umgebung befindet. • Es muss sich in derselben Amazon Virtual Private Cloud (Amazon-VPC) wie die Umgebung befinden. • Es darf sich nicht um einen Load Balancer handeln, der von Elastic Beanstalk als dedizierter Load Balancer für eine andere Umgebung erstellt wurde. Sie können diese dedizierten Load Balancer anhand des Präfixes identifizieren `awseb-`. Beispiel: `arn:aws:elasticloadbalancing:us-east`	Keine	ARN eines gültigen Load Balancers, der alle hier beschriebenen Kriterien erfüllt.

Name	Beschreibung	Standard	Zulässige Werte
	`-2:123456789012:loadbalancer/app/FrontEndLB/0dbf78d8ad96abbc`		

aws:rds:dbinstance

Konfigurieren Sie eine angeschlossene Amazon-RDS-DB-Instance.

Namespace: **`aws:rds:dbinstance`**

Name	Beschreibung	Standard	Zulässige Werte
DBAllocatedStorage	Die in Gigabyte (GB) angegebene Größe des zugewiesenen Datenbankspeichers.	MySQL: `5` Oracle: `10` sqlserver-se: `200` sqlserver-ex: `30` sqlserver-web: `30`	MySQL: `5-1024` Oracle: `10-1024` sqlserver: nicht änderbar

DBDeletionPolicy	Gibt an, ob ein Snapshot der DB-Instance beibehalten, gelöscht oder erstellt werden soll, wenn eine Umgebung beendet wird.
	Diese Option funktioniert in Verbindung mit `HasCoupledDatabase`, ebenfalls eine Option dieses Namespaces.
		`Delete`	`Delete`
			`Retain`
			`Snapshot`

Name	Beschreibung	Standard	Zulässige Werte
	Warning Das Löschen einer DB-Instanz führt zu einem permanenten Datenverlust.		

DBEngine	Der Name der Datenbank-Engine, die für diese Instance verwendet werden soll.	`mysql`	`mysql`
			`oracle-se1`
			`sqlserver-ex`
			`sqlserver-web`
			`sqlserver-se`
			`postgres`

DBEngineVersion	Die Versionsnummer des Datenbank-Engines.	`5.5`	
DBInstanceClass	Der Typ der Datenbank-Instance.	`db.t2.micro` (`db.m1.large` für eine Umgebung, die nicht in einer Amazon-VPC ausgeführt wird)	Weitere Informationen finden Sie unter DB-Instance-Klasse im Amazon-Relational-Database Service-Benutzerhandbuch.

Name	Beschreibung	Standard	Zulässige Werte
DBPassword	Der Name des Masterbenutzerkennworts dieser DB-Instance.	Keine	
DBSnapshotIdentifier	Kennung für den DB-Snapshot, der zur Wiederherstellung verwendet werden soll.	Keine	
DBUser	Der Name des Masterbenutzers dieser DB-Instance.	ebroot	

Name	Beschreibung	Standard	Zulässige Werte

HasCoupledDatabase	Gibt an, ob eine DB-Instance mit Ihrer Umgebung gekoppelt ist. Wenn auf `true` umgeschaltet, erstellt Elastic Beanstalk eine neue DB-Instance, die mit Ihrer Umgebung gekoppelt ist. Wenn auf `false` umgeschaltet, beginnt Elastic Beanstalk mit der Entkopplung der DB-Instance von Ihrer Umgebung.
	
	Diese Option funktioniert in Verbindung mit `DBDeletionPolicy` , ebenfalls eine Option dieses Namespaces.
	
	Note
	Hinweis: Wenn Sie diesen Wert nach dem Entkoppeln der vorherigen Datenbank wieder auf `true` setzen, erstellt Elastic Beanstalk eine neue Datenbank mit den vorherigen Datenbank optionseinstellungen. Um jedoch die Sicherheit Ihrer Umgebung zu gewährleisten, werden die vorhandenen `DBUser`- und `DBPassword` -Einstellungen nicht beibehalten. Sie müssen `DBUser` und `DBPassword` erneut angeben.
		`false`	`true`
			
			`false`

Name	Beschreibung	Standard	Zulässige Werte

MultiAZDatabase	Gibt an, ob für diese DB-Instance eine Multi-AZ-Bereitstellung erfolgen soll. Weitere Informationen zu Multi-AZ-Bereitstellungen mit Amazon Relational Database Service (RDS), finden Sie unter Regionen und Availability Zones im Amazon-Relational-Database-Service-Benutzerhandbuch.	`false`	`true`
			
			`false`

Plattformspezifische Optionen

Einige Elastic-Beanstalk-Plattformen definieren Options-Namespaces, die spezifisch für die Plattform sind. Diese Namespaces und ihre Optionen sind unterhalb jeder Plattform aufgeführt.

> **Note**
>
> Ehemals wurden in Leiterplattenversionen, die auf Amazon-Linux-AMI (vor Amazon-Linux-2) basierten, die folgenden zwei Features und ihre jeweiligen Namespaces als plattformspezifische Features betrachtet und hier pro Plattform aufgelistet:
>
> - Proxy-Konfiguration für statische Dateien – `aws:elasticbeanstalk:environment:proxy:staticfiles`
> - AWS X-Ray Unterstützung — `aws:elasticbeanstalk:xray`
>
> In Amazon-Linux-2-Plattformversionen implementiert Elastic Beanstalk diese Features auf konsistente Weise auf allen unterstützenden Plattformen. Der zugehörige Namespace wird nun auf der Seite the section called “Allgemeine Optionen” aufgelistet. Sie werden auf dieser Seite nur noch für Plattformen erwähnt, deren Namespaces anders benannt sind.

Plattformen

- Docker-Plattformoptionen
- Go-Plattformoptionen
- Java-SE-Plattform – Optionen

- Optionen für Java mit Tomcat-Plattform
- .NET Core auf Linux-Plattformoptionen
- .NET-Plattformoptionen
- Node.js-Plattformoptionen
- PHP-Plattformoptionen
- Python-Plattformoptionen
- Ruby-Plattformoptionen

Docker-Plattformoptionen

Die folgenden für Docker spezifischen Konfigurationsoptionen gelten für Docker- und Preconfigured Docker-Plattformen.

> **Note**
>
> Diese Konfigurationsoptionen gelten nicht für
>
> - Docker-Plattform (Amazon Linux 2) mit Docker Compose
> - Multicontainer Docker-Plattform (Amazon Linux AMI AL1) — diese Plattform wurde eingestellt

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name	Beschreibung	Standard	Zulässige Werte
ProxyServer	Gibt den Webserver an, der als Proxy verwendet werden soll.	`nginx`	`nginx` `none`— Nur Amazon Linux AM und Docker w/ DC

Go-Plattformoptionen

Amazon-Linux-AMI (vor Amazon-Linux-2)-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:container:golang:staticfiles`**

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereits, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitstellt Beispiel: Verwenden Sie `/images`, um Dateien unter *`subdomain`* `.elasticbeanstalk.com/images` bereitzustellen.	Der Name des Verzeichnisses, das die Dateien enthält Beispiel: Verwenden Sie `staticimages` , um Dateien aus einem Verzeichnis namens `staticimages` auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

Java-SE-Plattform – Optionen

Amazon-Linux-AMI (vor Amazon-Linux-2)-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:container:java:staticfiles`**

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereits, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitstellt Beispiel: Verwenden Sie `/images`, um Dateien unter *`subdomain`* `.elasticbeanstalk.com/images` bereitzustellen.	Der Name des Verzeichnisses, das die Dateien enthält Beispiel: Verwenden Sie `staticimages` , um Dateien aus einem Verzeichnis namens `staticimages` auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

Optionen für Java mit Tomcat-Plattform

Namespace: **`aws:elasticbeanstalk:application:environment`**

Name	Beschreibung	Standard	Zulässige Werte
JDBC_CONNECTION_STRING	Die Verbindungszeichenfolge für eine externe Datenbank.	–	–

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Namespace: **`aws:elasticbeanstalk:container:tomcat:jvmoptions`**

Name	Beschreibung	Standard	Zulässige Werte
JVM Options	Übergeben von Befehlszeilen-Optionen an die JVM beim Startup.	–	–
Xmx	Maximale JVM-Heap-Größen.	`256m`	–
XX:MaxPermSize	Bereich des JVM-Heap, der zum Speichern von Klassendefinitionen und zugehörige Metadaten verwendet wird.	`64m`	–

Name	Beschreibung	Standard	Zulässige Werte
	Note Diese Option gilt nur für Java-Versionen vor Java 8 und wird auf Plattformen von Elastic Beanstalk Tomcat basierend auf Amazon Linux 2 und später nicht unterstützt.		
Xms	Ursprüngliche JVM-Heap-Größen.	`256m`	–
optionName	Geben Sie neben den von der Tomcat-Plattform definierten JVM-Optionen beliebige weitere JVM-Optionen an.	–	–

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name	Beschreibung	Standard	Zulässige Werte
GzipCompression	Festlegen auf `false`, um die Antwortkomprimierung zu deaktivieren. Nur für Amazon Linux AMI-Plattformversionen gültig (vor Amazon Linux 2).	`true`	`true` `false`

ProxyServer	Legen Sie die Proxys fest, der für die Instances Ihrer Umgebung verwendet werden soll. Wenn Sie für diese Option `apache` festlegen, verwendet Elastic Beanstalk Apache 2.4.
	Legen Sie sie auf `apache/2.2` fest, wenn Ihre Anwendung aufgrund einer inkompatiblen Proxy-Konfigurationseinstellung für die Migration von Apache 2.2 nicht bereit ist. Dieser Wert ist nur für Amazon Linux AMI-
		`nginx` (Amazon-Linux-2)
		`apache` (Amazon-Linux-AMI)
			`apache`
			`apache/2.2` – nur Amazon-Linux-AMI
			`nginx`

Name	Beschreibung	Standard	Zulässige Werte
	Plattformversionen gültig (vor Amazon Linux 2). Legen Sie sie auf `nginx` fest, um nginx zu verwenden. Dies ist die Standardeinstellung ab Amazon-Linux-2-Plattformversionen. Weitere Informationen finden Sie unter Konfigurieren des Proxy-Server.		

.NET Core auf Linux-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name	Beschreibung	Standard	Zulässige Werte

ProxyServer	Gibt den Webserver an, der als Proxy verwendet werden soll.	`nginx`	`nginx`
			
			`none`

.NET-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:container:dotnet:apppool`**

Name	Beschreibung	Standard	Zulässige Werte

Target Runtime	Wählen Sie die Version von .NET-Framework für Ihre Anwendung.	`4.0`	`2.0`
			
			`4.0`

Enable 32-bit Applications	Legen Sie `True` fest, um 32-Bit-Anwendungen auszuführen.	`False`	`True`
			
			`False`

Node.js-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name	Beschreibung	Standard	Zulässige Werte

ProxyServer	Legen Sie die Proxys fest, der für die Instances Ihrer Umgebung verwendet werden soll.	`nginx`	`apache`
			
			`nginx`

Amazon-Linux-AMI (vor Amazon-Linux-2)-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:container:nodejs`**

Name	Beschreibung	Standard	Zulässige Werte
NodeCommand	Befehl zum Starten der Node.js-Anwendung. Wenn eine leere Zeichenfolge angegeben wird, wird `app.js`, dann `server.js` , dann `npm start` (in dieser Reihenfolge) verwendet.	""	–
NodeVersion	Version von Node.js. Beispiel, `4.4.6` Unterstützte Node.js-Versionen variieren zwischen den Node.js-Plattformversionen. Eine Liste der aktuell unterstützten Versionen finden Sie unter Node.js im Dokument AWS Elastic Beanstalk -Plattformen. **Note** Wenn die von Ihnen verwendete Version von Node.js von der Plattform nicht mehr unterstützt wird, müssen Sie die Versionseinstellung vor der Ausführung einer Plattform	variiert	variiert

Name	Beschreibung	Standard	Zulässige Werte
	aktualisierung ändern oder löschen. Dieser Fall kann eintreten, wenn eine Schwachstelle für eine oder mehrere Versionen von Node.js identifiziert wird. In diesem Fall schlägt der Versuch NodeVersionfehl, auf eine neue Version der Plattform zu aktualisieren, die die konfigurierte Version nicht unterstützt. Um zu vermeiden, dass eine neue Umgebung erstellt werden muss, ändern Sie die NodeVersionKonfigurationsoption in eine Version von Node.js, die sowohl von der alten als auch von der neuen Plattformversion unterstützt wird, oder entfernen Sie die Optionseinstellung und führen Sie dann das Plattformupdate durch.		
GzipCompression	Gibt an, ob die gzip-Komprimierung aktiviert ist. Wenn auf gesetzt ProxyServer istnone, ist die Gzip-Komprimierung deaktiviert.	`false`	`true` `false`

ProxyServer	Gibt an, welcher Webserver verwendet werden soll, um Proxy-Verbindungen zu Node.js zu verwenden. Wenn auf gesetzt ProxyServer istnone, werden statische Dateizuordnungen nicht wirksam und die Gzip-Komprimierung ist deaktiviert.	`nginx`	`apache`
			`nginx`
			`none`

Namespace: **`aws:elasticbeanstalk:container:nodejs:staticfiles`**

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereits, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

> **Note**
>
> Statische Dateieinstellungen werden nicht angewendet, wenn `aws:elasticbeanstalk:container:nodejs::ProxyFiles` auf none festgelegt ist.

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitstellt Beispiel: Verwenden Sie `/images`, um Dateien unter *`subdomain`* `.elasticbeanstalk.com/images` bereitzustellen.	Der Name des Verzeichnisses, das die Dateien enthält Beispiel: Verwenden Sie `staticimages`, um Dateien aus einem Verzeichnis namens `staticimages` auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

PHP-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:container:php:phpini`**

Name	Beschreibung	Standard	Zulässige Werte
document_root	Geben Sie das untergeordnete Verzeichnis Ihres Projekts an, das als der öffentlich zugängliche Webstamm behandelt wird.	`/`	Es wird eine leere Zeichenfolge als `/` behandelt oder geben Sie eine Zeichenfo

Name	Beschreibung	Standard	Zulässige Werte
			lge an, die wie folgt beginnt: `/`
memory_limit	Menge des Speichers, der der PHP-Umgebung zugeordnet ist.	`256M`	–

zlib.output_compression	Gibt an, ob PHP die Komprimierung für die Ausgabe nutzen soll.	`Off`	`On`
			`Off`
			`true`
			`false`

allow_url_fopen	Gibt an, wenn die Datei-Features von PHP Daten aus Remote-Standorten abrufen können, wie Websites oder FTP-Server.	`On`	`On`
			`Off`
			`true`
			`false`

display_errors	Gibt an, ob Fehlermeldungen Teil der Ausgabe sein sollen.	`Off`	`On` `Off`
max_execution_time	Gibt die maximale Zeitspanne (in Sekunden) an, die ein Skript ausgeführt werden kann, bevor es von der Umgebung beendet wird.	`60`	`0` auf `9223372036854775807` (PHP_INT_MAX)
composer_options	Legt benutzerdefinierte Optionen fest, die bei der Installation von Abhängigkeiten mit Composer über den Befehl verwendet werden sollen. composer.phar install Weitere Informationen finden Sie unter install auf der Website getcomposer.org.	–	–

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name	Beschreibung	Standard	Zulässige Werte

ProxyServer	Legen Sie die Proxys fest, der für die Instances Ihrer Umgebung verwendet werden soll.	`nginx`	`apache`
			
			`nginx`

> **Note**
>
> Weitere Informationen zur PHP-Plattform finden Sie unter Verwenden der Elastic Beanstalk PHP-Plattform.

Python-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:application:environment`**

Name	Beschreibung	Standard	Zulässige Werte
DJANGO_SETTINGS_MODULE	Gibt an, welche Einstellungsdatei verwendet werden soll.	–	–

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Namespace: **`aws:elasticbeanstalk:container:python`**

Name	Beschreibung	Standard	Zulässige Werte
WSGIPath	Die Datei, die die WSGI-Anwendung enthält. Diese Datei muss über eine aufrufbare `application` verfügen.	Auf Amazon-Linux-2-Python-Plattformversionen:	–

Name	Beschreibung	Standard	Zulässige Werte
		`application` Auf Amazon-Linux-AMI-Python-Plattformversionen: `application.py`	
NumProcesses	Die Anzahl der Daemon-Prozesse, die für die Prozessgruppe gestartet werden sollen, wenn WSGI-Anwendungen ausgeführt werden.	1	–
NumThreads	Die Anzahl der Threads, die erstellt werden sollen, um Anfragen in jedem Daemon-Prozess innerhalb der Prozessgruppe zu verarbeiten, wenn WSGI-Anwendungen ausgeführt werden.	15	–

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name	Beschreibung	Standard	Zulässige Werte

ProxyServer	Legen Sie die Proxys fest, der für die Instances Ihrer Umgebung verwendet werden soll.	`nginx`	`apache`
			
			`nginx`

Amazon-Linux-AMI (vor Amazon-Linux-2)-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:container:python:staticfiles`**

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad

erhält, stellt er die Datei direkt bereits, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

Standardmäßig stellt der Proxy-Server in einer Python-Umgebung alle Dateien im Ordner `static` unter dem Pfad `/static` bereit.

Namespace: **`aws:elasticbeanstalk:container:python:staticfiles`**

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitstellt Beispiel: Verwenden Sie `/images`, um Dateien unter *`subdomain`* `.eleasticbeanstalk.com/images` bereitzustellen.	Der Name des Verzeichnisses, das die Dateien enthält Beispiel: Verwenden Sie `staticimages` , um Dateien aus einem Verzeichnis namens `staticimages` auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

Ruby-Plattformoptionen

Namespace: **`aws:elasticbeanstalk:application:environment`**

Name	Beschreibung	Standard	Zulässige Werte

RAILS_SKIP_MIGRATIONS	Gibt an, ob `` `rake db:migrate` `` im Namen der Anwendungen der Benutzer ausgeführt werden oder ob dies übersprungen werden soll. Dies gilt nur für Rails-3-Anwendungen.	`false`	`true`
			
			`false`

RAILS_SKIP_ASSET_COMPILATION	Gibt an, ob der Container `` `rake assets:precompile` `` im Namen der Anwendungen der Benutzer ausgeführt werden oder ob dies übersprungen	`false`	`true`
			
			`false`

Name	Beschreibung	Standard	Zulässige Werte
	werden soll. Dies gilt auch für Rails-3-Anwendungen.		
BUNDLE_WITHOUT	Eine durch Doppelpunkte (`:`) getrennte Liste mit Gruppen, die bei der Installation von Abhängigkeiten aus einer "Gemfile"-Datei ignoriert werden sollen.	`test:development`	–
RACK_ENV	Gibt an, in welcher Umgebungsphase eine Anwendung ausgeführt werden kann. Beispiele für gängige Umgebungen sind Entwicklung, Produktion, Test.	`production`	–

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Benutzerdefinierte Optionen

Verwenden Sie den `aws:elasticbeanstalk:customoption`-Namespace, um Optionen und Werte zu definieren, die in `Resources`-Blocks in anderen Konfigurationsdateien gelesen werden können. Verwenden Sie benutzerdefinierte Optionen zur Erfassung von benutzerspezifischen Einstellungen in einer einzelnen Konfigurationsdatei.

Vielleicht haben Sie eine komplexe Konfigurationsdatei, die eine Ressource definiert, die vom Benutzer konfiguriert werden kann, der die Umgebung startet. Wenn Sie mit `Fn::GetOptionSetting` den Wert für eine benutzerdefinierte Option abrufen, können Sie die Definition dieser Option in eine andere Konfigurationsdatei legen, wo sie einfacher entdeckt und vom Benutzer geändert werden kann.

Da es sich um Konfigurationsoptionen handelt, können benutzerdefinierte Optionen außerdem auf API-Ebene festgelegt werden, um Werte in einer Konfigurationsdatei zu überschreiben. Informationen finden Sie unter Precedence.

Benutzerdefinierte Optionen sind genau wie alle anderen Optionen definiert:

```
option_settings:
  aws:elasticbeanstalk:customoption:
```

```
    option name: option value
```

Die folgende Konfigurationsdatei erstellt beispielsweise eine Option mit dem Namen `ELBAlarmEmail` und legt den Wert auf `someone@example.com` fest:

```
option_settings:
  aws:elasticbeanstalk:customoption:
    ELBAlarmEmail: someone@example.com
```

In anderen Fällen definiert eine Konfigurationsdatei ein SNS-Thema, das die Option mit `Fn::GetOptionSetting` liest, um den Wert des `Endpoint`-Attributs zu füllen:

```
Resources:
  MySNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint:
            Fn::GetOptionSetting:
              OptionName: ELBAlarmEmail
              DefaultValue: nobody@example.com
          Protocol: email
```

Weitere Beispielausschnitte mit `Fn::GetOptionSetting` finden Sie unter Hinzufügen und Anpassen von Elastic Beanstalk-Umgebungsressourcen.

Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`)

Sie können dem Quellcode Ihrer Webanwendung AWS Elastic Beanstalk Konfigurationsdateien (`.ebextensions`) hinzufügen, um Ihre Umgebung zu konfigurieren und die darin enthaltenen AWS Ressourcen anzupassen. Konfigurationsdateien sind Dokumente im YAML- oder JSON-Format mit einer `.config` Dateierweiterung, die Sie in einem Ordner mit dem Namen Ihres Anwendungsquellpakets `.ebextensions` ablegen und dort bereitstellen.

Example.ebextensions/ .config network-load-balancer

In diesem Beispiel wird eine einfache Konfigurationsänderung vorgenommen. Es wird eine Konfigurationsoption geändert, um den Typ des Load Balancers Ihrer Umgebung auf „Network Load Balancer“ festzulegen.

```
option_settings:
  aws:elasticbeanstalk:environment:
    LoadBalancerType: network
```

Wir empfehlen die Verwendung von YAML für Konfigurationsdateien, da das Format besser lesbar ist als JSON. YAML unterstützt Kommentare, mehrzeilige Befehle, verschiedene Alternativen für Anführungszeichen und vieles mehr. Sie können jedoch alle Konfigurationsänderungen in Elastic Beanstalk-Konfigurationsdateien gleichermaßen mit YAML oder JSON vornehmen.

> **Tipp**
>
> Für die Entwicklung und die Testphase neuer Konfigurationsdateien starten Sie eine bereinigte Umgebung, auf der die Standardanwendung ausgeführt wird, und stellen diese dort bereit. Schlecht formatierte Konfigurationsdateien führen dazu, dass eine neue Umgebung nicht gestartet werden kann und nicht wiederherstellbar ist.

Im Abschnitt `option_settings` der Konfigurationsdatei werden die Werte für Konfigurationsoptionen definiert. Mit den Konfigurationsoptionen können Sie Ihre Elastic Beanstalk Beanstalk-Umgebung, die darin AWS enthaltenen Ressourcen und die Software, mit der Ihre Anwendung ausgeführt wird, konfigurieren. Konfigurationsdateien sind nur eine von mehreren Möglichkeiten zum Festlegen von Konfigurationsoptionen.

`Resources`In diesem Abschnitt können Sie die Ressourcen in der Umgebung Ihrer Anwendung weiter anpassen und zusätzliche AWS Ressourcen definieren, die über die in den Konfigurationsoptionen bereitgestellten Funktionen hinausgehen. Sie können alle Ressourcen hinzufügen und konfigurieren AWS CloudFormation, die von Elastic Beanstalk unterstützt werden, um Umgebungen zu erstellen.

In den anderen Abschnitten einer Konfigurationsdatei (`packages`,,`sources`,`files`, `users` `groups` `commands`container_commands, und`services`) können Sie die EC2 Instances konfigurieren, die in Ihrer Umgebung gestartet werden. Jedes Mal, wenn ein Server in der Umgebung gestartet wird, führt Elastic Beanstalk die in diesen Abschnitten definierten Vorgänge aus, um das Betriebs- und das Speichersystem für die Anwendung vorzubereiten.

Beispiele für häufig verwendete .ebextensions finden Sie im Elastic Beanstalk-Konfigurationsdateien-Repository.

Voraussetzungen

- Standort — Elastic Beanstalk verarbeitet alle in Ihrem `.ebextensions` Deployment vorhandenen Ordner. Wir empfehlen jedoch, dass Sie alle Ihre Konfigurationsdateien in einem einzigen Ordner mit dem Namen `.ebextensions` „Root“ im Stammverzeichnis Ihres Quellpakets ablegen. Verzeichnisse, die mit einem Punkt beginnen, können von Dateibrowsern ausgeblendet werden. Stellen Sie daher sicher, dass das Verzeichnis beim Erstellen des Quell-Bundles angelegt wird. Weitere Informationen finden Sie unter Ein Elastic Beanstalk Beanstalk-Anwendungsquellpaket erstellen.
- Naming (Namensgebung) – Konfigurationsdateien müssen die Dateiendung `.config` haben.
- Formatting (Format) – Konfigurationsdateien müssen den YAML- oder JSON-Spezifikationen entsprechen.

  Bei Verwendung von YAML immer Leerzeichen verwenden, um Schlüssel in den verschiedenen Verschachtelungsebenen einzurücken. Weitere Informationen zu YAML finden Sie unter YAML Ain't Markup Language (YAML™) Version 1.1.
- Uniqueness (Eindeutigkeit) – Verwenden Sie jeden Schlüssel in jeder Konfigurationsdatei nur einmal.

> **Warnung**
>
> Wenn Sie einen Schlüssel (z. B. `option_settings`) innerhalb einer Konfigurationsdatei mehrfach verwenden, wird einer der Abschnitte gelöscht. Kombinieren Sie doppelte Abschnitte in einem Abschnitt oder platzieren Sie diese in separaten Konfigurationsdateien.

Der Prozess für die Bereitstellung kann je nach dem Client, der für die Umgebungsverwaltung verwendet wird, unterschiedlich sein. Ausführliche Informationen finden Sie in den folgenden Abschnitten:

- Elastic Beanstalk-Konsole
- EB CLI
- AWS CLI

Themen

- Optionseinstellungen

- Anpassen von Software auf Linux-Servern
- Anpassen von Software auf Windows-Servern
- Hinzufügen und Anpassen von Elastic Beanstalk-Umgebungsressourcen

Optionseinstellungen

Mit dem `option_settings`-Schlüssel können Sie die Elastic Beanstalk-Konfiguration ändern und Variablen definieren, die von der Anwendung über Umgebungsvariablen abgerufen werden. Einige Namespaces bieten die Möglichkeit, die Anzahl der Parameter zu erhöhen und Parameternamen anzugeben. Eine Liste der Namespaces und Konfigurationsoptionen finden Sie unter Konfigurationsoptionen.

Optionseinstellungen können während der Umgebungserstellung oder einer -aktualisierung auch direkt in eine Umgebung übernommen werden. Werden Einstellungen direkt für die Umgebung bereitgestellt, überschreibt dies die Einstellungen für dieselben Optionen in Konfigurationsdateien. Wenn Sie Einstellungen aus der Konfiguration einer Umgebung entfernen, werden die Einstellungen in Konfigurationsdateien wirksam. Details dazu finden Sie unter Precedence.

Syntax

Die Standardsyntax für Optionseinstellungen besteht aus einem Array von Objekten, von denen jedes die Schlüssel `namespace`, `option_name` und `value` aufweist.

```
option_settings:
  - namespace:  namespace
    option_name:  option name
    value:  option value
  - namespace:  namespace
    option_name:  option name
    value:  option value
```

Der `namespace`-Schlüssel ist optional. Falls Sie keinen Namespace angeben, wird der Standardwert `aws:elasticbeanstalk:application:environment`verwendet:

```
option_settings:
  - option_name:  option name
    value:  option value
  - option_name:  option name
    value:  option value
```

Elastic Beanstalk unterstützt auch eine Syntax-Kurznotation für Optionseinstellungen, mit der Sie Optionen als Schlüssel-Wert-Paare unter dem Namespace angeben können:

```
option_settings:
  namespace:
    option name: option value
    option name: option value
```

Beispiele

Im folgenden Beispiel wird eine spezifische Option für die Tomcat-Plattform im Namespace `aws:elasticbeanstalk:container:tomcat:jvmoptions` und in der Umgebungseigenschaft `MYPARAMETER` eingerichtet.

In YAML-Standardformat:

Example.ebextensions/options.config

```
option_settings:
  - namespace:  aws:elasticbeanstalk:container:tomcat:jvmoptions
    option_name:  Xmx
    value:  256m
  - option_name: MYPARAMETER
    value: parametervalue
```

In der Kurznotation:

Example.ebextensions/options.config

```
option_settings:
  aws:elasticbeanstalk:container:tomcat:jvmoptions:
    Xmx: 256m
  aws:elasticbeanstalk:application:environment:
    MYPARAMETER: parametervalue
```

Im JSON-Format:

Example.ebextensions/options.config

```
{
```

```
  "option_settings": [
    {
      "namespace": "aws:elasticbeanstalk:container:tomcat:jvmoptions",
      "option_name": "Xmx",
      "value": "256m"
    },
    {
      "option_name": "MYPARAMETER",
      "value": "parametervalue"
    }
  ]
}
```

Anpassen von Software auf Linux-Servern

In diesem Abschnitt wird beschrieben, welche Art von Informationen Sie in eine Konfigurationsdatei aufnehmen können, um die Software auf Ihren EC2 Linux-Instances anzupassen. Allgemeine Informationen zum Anpassen und Konfigurieren der Elastic Beanstalk-Umgebungen finden Sie unter Konfigurieren von Elastic-Beanstalk-Umgebungen. Informationen zum Anpassen der Software auf Ihren EC2 Instanzen, auf denen Windows ausgeführt wird, finden Sie unterAnpassen von Software auf Windows-Servern.

Möglicherweise muss die Software, von der Ihre Anwendung abhängig ist, angepasst und konfiguriert werden. Sie können Befehle hinzufügen, die während der Instance-Bereitstellung ausgeführt werden sollen, Linux-Benutzer und -Gruppen definieren und Dateien auf Ihre Umgebungs-Instances herunterladen oder dort direkt erstellen. Bei diesen Dateien kann es sich entweder um von der Anwendung benötigte Abhängigkeiten (z. B. zusätzliche Pakete aus dem yum-Repository) oder um Konfigurationsdateien handeln (wie z. B. ein Ersatz für eine Proxykonfigurationsdatei zum Überschreiben bestimmter Standardeinstellungen von Elastic Beanstalk).

Hinweise

- Wir empfehlen dringend, auf Amazon Linux 2-Plattformen Buildfile zu verwenden, anstatt Dateien und Befehle in .ebextensions-Konfigurationsdateien bereitzustellen. Procfile und Plattform-Hooks möglichst zum Konfigurieren und Ausführen von benutzerdefiniertem Code auf Ihren Umgebungs-Instances während der Instance-Bereitstellung. Details zu diesen Mechanismen finden Sie unter the section called “Erweitern von Linux-Plattformen”.
- Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und

stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Konfigurationsdateien unterstützen die folgenden Schlüssel, die sich auf die Linux-Server auswirken, auf denen die Anwendung ausgeführt wird.

Schlüssel

- Pakete
- Gruppen
- Benutzer
- Quellen
- Dateien
- Befehle
- Dienstleistungen
- Container-Befehle
- Beispiel: Verwendung benutzerdefinierter CloudWatch Amazon-Metriken

Schlüssel werden in der hier aufgeführten Reihenfolge verarbeitet.

Beobachten Sie die Ereignisse in der Umgebung, während Sie Konfigurationsdateien entwickeln und testen. Elastic Beanstalk ignoriert eine Konfigurationsdatei, die Validierungsfehler enthält, z. B. einen ungültigen Schlüssel, und verarbeitet keinen der in der betreffenden Datei enthaltenen Schlüssel. Wenn dies geschieht, fügt Elastic Beanstalk eine Warnung in das Ereignisprotokoll ein.

Pakete

Mit dem Schlüssel `packages` können Sie vorgefertigte Anwendungen und Komponenten herunterladen und installieren.

Syntax

```
packages:
  name of package manager:
    package name: version
    ...
```

```
  name of package manager:
    package name: version
    ...
  ...
```

Sie können unter jedem Paketmanager-Schlüssel mehrere Pakete angeben.

Unterstützte Paketformate

Elastic Beanstalk unterstützt derzeit die folgenden Paketmanager: yum, rubygems, python und rpm. Pakete werden in der folgenden Reihenfolge verarbeitet: rpm, yum und dann rubygems und python. Es gibt keine Reihenfolge zwischen rubygems und python. Innerhalb jedes Paketmanagers ist die Reihenfolge der Paketinstallation nicht garantiert. Verwenden Sie einen Paketmanager, der vom Betriebssystem unterstützt wird.

> **Note**
>
> Elastic Beanstalk unterstützt zwei zugrundeliegende Paketmanager für Python, pip und easy_install. In der Syntax der Konfigurationsdatei müssen Sie den Namen des Paketmanagers als `python` angeben. Wenn Sie eine Konfigurationsdatei verwenden, um einen Python-Paketmanager anzugeben, verwendet Elastic Beanstalk Python 2.7. Wenn Ihre Anwendung sich auf eine andere Version von Python stützt, können Sie die zu installierenden Pakete in einer `requirements.txt`-Datei angeben. Weitere Informationen finden Sie unter Angeben von Abhängigkeiten mithilfe einer Anforderungsdatei auf Elastic Beanstalk.

Angeben von Versionen

Innerhalb jedes Paketmanagers ist jedes Paket als ein Paketname und eine Liste der Versionen angegeben. Die Version kann eine Zeichenfolge, eine Liste von Versionen oder eine leere Zeichenfolge oder Liste sein. Eine leere Zeichenfolge oder Liste gibt an, dass Sie die neueste Version verwenden möchten. Bei RPM-Manager wird die Version als ein Pfad zu einer Datei auf einem Datenträger oder eine URL angegeben. Relative Pfade werden nicht unterstützt.

Wenn Sie eine Version eines Pakets angeben, versucht Elastic Beanstalk, diese Version zu installieren, auch wenn auf der Instance bereits eine neuere Version des Pakets installiert ist. Wenn eine neuere Version bereits installiert ist, schlägt die Bereitstellung fehl. Einige Paketmanager unterstützen mehrere Versionen, andere wiederum nicht. In der Dokumentation zu Ihrem Paketmanager finden Sie weitere Informationen. Wenn Sie keine Version angeben und bereits

eine Version des Pakets installiert ist, installiert Elastic Beanstalk keine neue Version. Es wird angenommen, dass Sie die bestehende Version behalten und verwenden möchten.

Beispielausschnitt

Der folgende Codeausschnitt gibt eine Versions-URL für RPM an, fordert die neueste Version von yum und Version 0.10.2 von chef von rubygems an.

```
packages:
  yum:
    libmemcached: []
    ruby-devel: []
    gcc: []
  rpm:
    epel: http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
  rubygems:
    chef: '0.10.2'
```

Gruppen

Sie können den `groups` Schlüssel verwenden, um Linux/UNIX Gruppen zu erstellen und Gruppen zuzuweisen IDs. Um eine Gruppe zu erstellen, fügen Sie ein neues Schlüssel-Wert-Paar hinzu, das einer optionalen Gruppen-ID einen neuen Gruppennamen zuordnet. Der Gruppenschlüssel kann einen oder mehrere Gruppennamen enthalten. In der folgenden Tabelle werden die verfügbaren Schlüssel aufgeführt.

Syntax

```
groups:
  name of group: {}
  name of group:
    gid: "group id"
```

Optionen

`gid`

Eine Gruppen-ID-Nummer.

Wenn eine Gruppen-ID angegeben ist und die Gruppe nach Name bereits vorhanden ist, schlägt die Gruppenerstellung fehl. Wenn eine andere Gruppe die angegebene Gruppen-ID aufweist, lehnt das Betriebssystem die Gruppenerstellung möglicherweise ab.

Beispielausschnitt

Der folgende Codeausschnitt gibt eine Gruppe mit dem Namen groupOne an, ohne eine Gruppen-ID zuzuweisen, und eine Gruppe mit dem Namen groupTwo, die den Gruppen-ID-Wert 45 angegeben hat.

```
groups:
  groupOne: {}
  groupTwo:
    gid: "45"
```

Benutzer

Sie können den `users` Schlüssel verwenden, um Linux/UNIX Benutzer auf der EC2 Instanz zu erstellen.

Syntax

```
users:
  name of user:
    groups:
      - name of group
    uid: "id of the user"
    homeDir: "user's home directory"
```

Optionen

`uid`

Eine Benutzer-ID. Der Erstellungsprozess ist nicht erfolgreich, wenn der Benutzername mit einer anderen Benutzer-ID vorhanden ist. Wenn die Benutzer-ID bereits einem vorhandenen Benutzer zugewiesen ist, lehnt das Betriebssystem die Erstellungsanforderung möglicherweise ab.

`groups`

Eine Liste von Gruppennamen. Der Benutzer wird jeder Gruppe in der Liste hinzugefügt.

`homeDir`

Das Stammverzeichnis des Benutzers.

Benutzer werden als nicht interaktive Systembenutzer mit der Shell `/sbin/nologin` erstellt. Dies ist beabsichtigt und kann nicht geändert werden.

Beispielausschnitt

```
users:
  myuser:
    groups:
      - group1
      - group2
    uid: "50"
    homeDir: "/tmp"
```

Quellen

Sie können den `sources` Schlüssel verwenden, um eine Archivdatei von einer öffentlichen URL herunterzuladen und sie in ein Zielverzeichnis auf der EC2 Instanz zu entpacken.

Syntax

```
sources:
  target directory: location of archive file
```

Unterstützte Formate

Unterstützte Formate sind tar, tar+gzip, tar+bz2 und zip. Sie können auf externe Speicherort wie Amazon Simple Storage Service (Amazon S3) verweisen (z. B. `https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject`), sofern die URL öffentlich zugänglich ist.

Beispielausschnitt

Im folgenden Beispiel wird eine öffentliche ZIP-Datei von einem Amazon S3-Bucket heruntergeladen und in `/etc/myapp` entpackt:

```
sources:
  /etc/myapp: https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject
```

> **Note**
>
> Mehrere Extraktionen sollten nicht den gleichen Zielpfad wiederverwenden. Das Extrahieren einer anderen Quelle in denselben Zielpfad ersetzt den Inhalt, statt an den Inhalt angefügt zu werden.

Dateien

Sie können den `files` Schlüssel verwenden, um Dateien auf der EC2 Instanz zu erstellen. Der Inhalt kann entweder in die Konfigurationsdatei eingebettet sein oder von einer URL abgerufen werden. Die Dateien werden in lexikalischer Reihenfolge auf den Datenträger geschrieben.

Wenn private Dateien mit dem Schlüssel `files` von Amazon S3 heruntergeladen werden sollen, stellen Sie ein Instance-Profil für die Autorisierung bereit.

Wenn der von Ihnen angegebene Dateipfad bereits auf der Instance vorhanden ist, wird die vorhandene Datei beibehalten, wobei die Erweiterung `.bak` an ihren Namen angehängt wird.

Syntax

```
files:
  "target file location on disk":
     mode: "six-digit octal value"
     owner: name of owning user for file
     group: name of owning group for file
     source: URL
     authentication: authentication name:

  "target file location on disk":
     mode: "six-digit octal value"
     owner: name of owning user for file
     group: name of owning group for file
     content: |
      # this is my
      # file content
     encoding: encoding format
     authentication: authentication name:
```

Optionen

`content`

Zeichenfolgeninhalt zum Hinzufügen zur Datei. Geben Sie `content` oder `source` an, jedoch nicht beides.

`source`

URL einer Datei zum Herunterladen. Geben Sie `content` oder `source` an, jedoch nicht beides.

`encoding`

Das Codierungsformat der angegebene Zeichenfolge mit der `content`-Option.

Zulässige Werte: `plain` | `base64`

`group`

Linux-Gruppe, der die Datei gehört.

`owner`

Linux-Benutzer, dem die Datei gehört.

`mode`

Ein sechsstelliger Oktalwert, der für den Modus für diese Datei steht. Wird für Windows-Systeme nicht unterstützt. Verwenden Sie die ersten drei Ziffern für Symlinks und die letzten drei Ziffern für das Einrichten von Berechtigungen. Um einen Symlink zu erstellen, geben Sie `120`*`xxx`* an, wobei `xxx` die Berechtigungen der Zieldatei definiert. Um Berechtigungen für eine Datei anzugeben, verwenden Sie die letzten drei Ziffern, z. B. `000644`.

`authentication`

Der Name einer CloudFormation -Authentifizierungsmethode, die verwendet werden soll. Mit dem Ressourcenschlüssel können Sie Authentifizierungsmethoden zu den Metadaten der Auto Scaling-Gruppe hinzufügen. Ein Beispiel finden Sie unten.

Beispielausschnitt

```
files:
  "/home/ec2-user/myfile" :
    mode: "000755"
    owner: root
    group: root
    source: http://foo.bar/myfile

  "/home/ec2-user/myfile2" :
    mode: "000755"
    owner: root
    group: root
    content: |
      this is my
```

```
      file content
```

Beispiel mit einem Symlink. Dies erzeugt einen Link `/tmp/myfile2.txt`, der auf die Datei `/tmp/myfile1.txt` verweist.

```
files:
  "/tmp/myfile2.txt" :
    mode: "120400"
    content: "/tmp/myfile1.txt"
```

In dem folgenden Beispiel wird der Ressourcenschlüssel zum Hinzufügen einer Authentifizierungsmethode mit dem Namen S3Auth und zum Herunterladen einer privaten Datei aus einem Amazon S3-Bucket verwendet:

```
Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          buckets: ["amzn-s3-demo-bucket2"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
              DefaultValue: "aws-elasticbeanstalk-ec2-role"

files:
  "/tmp/data.json" :
    mode: "000755"
    owner: root
    group: root
    authentication: "S3Auth"
    source: https://elasticbeanstalk-us-west-2-123456789012.s3-us-west-2.amazonaws.com/data.json
```

Befehle

Sie können den `commands` Schlüssel verwenden, um Befehle auf der EC2 Instanz auszuführen. Die Befehle werden ausgeführt, bevor die Anwendung und der Webserver eingerichtet sind und die Anwendungsversionsdatei extrahiert wird.

Die angegebenen Befehle werden als Root-Benutzer ausgeführt und in alphabetischer Reihenfolge dem Namen nach verarbeitet. Befehle werden standardmäßig im Stammverzeichnis ausgeführt. Um die Befehle aus einem anderen Verzeichnis auszuführen, verwenden Sie die cwd-Option.

Um Probleme mit Befehlen zu beheben, schlagen Sie deren Ausgabe in den Instance-Protokollen nach.

Syntax

```
commands:
  command name:
    command: command to run
    cwd: working directory
    env:
      variable name: variable value
    test: conditions for command
    ignoreErrors: true
```

Optionen

`command`

Ein Array (Blocksequenzsammlung in YAML-Syntax) oder eine Zeichenfolge, das bzw. die den auszuführenden Befehl angibt. Einige wichtige Hinweise:

- Wenn Sie eine Zeichenfolge verwenden, müssen Sie die Zeichenfolge nicht in Anführungszeichen einschließen. Wenn Sie Anführungszeichen verwenden, muss literal verwendeten Anführungszeichen desselben Typs ein Escape-Zeichen vorangestellt werden.
- Wenn Sie ein Array verwenden, müssen Sie Leerzeichen kein Escape-Zeichen voranstellen und Befehlsparameter nicht in Anführungszeichen angeben. Jedes Arrayelement ist ein einzelnes Befehlsargument. Verwenden Sie ein Array nicht, um mehrere Befehle anzugeben.

Die folgenden Beispiele sind gleichwertig:

```
commands:
  command1:
    command: git commit -m "This is a comment."
  command2:
    command: "git commit -m \"This is a comment.\""
  command3:
    command: 'git commit -m "This is a comment."'
  command4:
```

```
    command:
      - git
      - commit
      - -m
      - This is a comment.
```

Wenn Sie mehrere Befehle angeben, verwenden Sie ein literales Blockskalar, wie im folgenden Beispiel gezeigt.

```
commands:
  command block:
    command: |
      git commit -m "This is a comment."
      git push
```

`env`

(Optional) Legt Umgebungsvariablen für den Befehl fest. Diese Eigenschaft überschreibt die vorhandene Umgebung, anstatt sie anzuhängen.

`cwd`

(Optional) Das Arbeitsverzeichnis. Sofern nicht definiert, werden Befehle aus dem Stammverzeichnis (/) ausgeführt.

`test`

(Optional) Ein Befehl, der den Wert `true` (Beendigungscode 0) zurückgeben muss, damit Elastic Beanstalk den im Schlüssel `command` enthaltenen Befehl, wie ein Shell-Skript, verarbeitet.

`ignoreErrors`

(Optional) Ein boolescher Wert, der bestimmt, ob andere Befehle ausgeführt werden sollen, wenn der im Schlüssel `command` enthaltene Befehl fehlschlägt (gibt einen Wert ungleich null zurück). Legen Sie diesen Wert auf `true` fest, damit weiterhin Befehle ausgeführt werden, selbst wenn der Befehl fehlschlägt. Legen Sie den Wert auf `false` fest, um anzugeben, dass keine Befehle mehr ausgeführt werden sollen, wenn der Befehl fehlschlägt. Der Standardwert ist `false`.

Beispielausschnitt

Der folgende Beispielausschnitt führt ein Python-Skript aus.

```
commands:
```

```
  python_install:
    command: myscript.py
    cwd: /home/ec2-user
    env:
      myvarname: myvarvalue
    test: "[ -x /usr/bin/python ]"
```

Dienstleistungen

Mit dem Schlüssel `services` definieren Sie, welche Services beim Start der Instance gestartet oder gestoppt werden sollen. Zudem lassen sich mit dem Schlüssel `services` auch Abhängigkeiten von Quellen, Paketen und Dateien angeben, sodass Elastic Beanstalk im Falle eines Neustarts aufgrund von installierten Dateien den Service neu startet.

Syntax

```
services:
  sysvinit:
    name of service:
      enabled: "true"
      ensureRunning: "true"
      files:
        - "file name"
      sources:
        - "directory"
      packages:
        name of package manager:
          "package name[: version]"
      commands:
        - "name of command"
```

Optionen

`ensureRunning`

Legen Sie den Wert auf `true` fest, um sicherzustellen, dass der Service ausgeführt wird, nachdem Elastic Beanstalk abgeschlossen wurde.

Legen Sie den Wert auf `false` fest, um sicherzustellen, dass der Service nicht ausgeführt wird, nachdem Elastic Beanstalk abgeschlossen wurde.

Lassen Sie diesen Schlüssel aus, um keine Änderungen am Servicestatus vorzunehmen.

`enabled`

Legen Sie den Wert auf `true` fest, um sicherzustellen, dass der Service beim Systemstart automatisch gestartet wird.

Legen Sie den Wert auf `false` fest, um sicherzustellen, dass der Service beim Systemstart nicht automatisch gestartet wird.

Lassen Sie diesen Schlüssel aus, um keine Änderungen an dieser Eigenschaft vorzunehmen.

`files`

Eine Liste von Dateien. Wenn Elastic Beanstalk eine Datei direkt über den Dateienblock ändert, wird der Service neu gestartet.

`sources`

Eine Liste von Verzeichnissen. Wenn Elastic Beanstalk ein Archiv in eines dieser Verzeichnisse erweitert, wird der Service neu gestartet.

`packages`

Eine Zuordnung des Paket-Managers zu einer Liste von Paketnamen. Wenn Elastic Beanstalk eines dieser Pakete installiert oder aktualisiert, wird der Service neu gestartet.

`commands`

Eine Liste von Befehlsnamen. Wenn Elastic Beanstalk den angegebenen Befehl ausführt, wird der Service neu gestartet.

Beispielausschnitt

Es folgt ein Beispiel für einen Ausschnitt:

```
services:
  sysvinit:
    myservice:
      enabled: true
      ensureRunning: true
```

Container-Befehle

Mit dem Schlüssel `container_commands` können Sie Befehle ausführen, die sich auf den Anwendungsquellcode auswirken. Container-Befehle werden ausgeführt, nachdem die Anwendung

und der Webserver eingerichtet sind und die Anwendungsversionsdatei extrahiert wurde, aber bevor die Anwendungsversion bereitgestellt wird. Nicht-Container-Befehle und andere Anpassungen werden vor der Extraktion des Anwendungsquellcodes ausgeführt.

Die angegebenen Befehle werden als Root-Benutzer ausgeführt und in alphabetischer Reihenfolge dem Namen nach verarbeitet. Container-Befehle werden aus dem Staging-Verzeichnis ausgeführt, aus dem der Quellcode vor der Bereitstellung auf dem Anwendungsserver extrahiert wird. Alle Änderungen, die Sie mithilfe eines Container-Befehls am Quellcode im Staging-Verzeichnis vornehmen, werden bei der Bereitstellung der Quelle am endgültigen Speicherort einbezogen.

> **Note**
>
> Die Ausgabe Ihrer Container-Befehle wird im `cfn-init-cmd.log`-Instance-Protokoll protokolliert. Weitere Informationen zum Abrufen und Anzeigen von Instance-Logs finden Sie unter Logs von EC2 Amazon-Instances anzeigen.

Verwenden Sie den Befehl `leader_only`, wenn der Befehl nur auf einer einzelnen Instance ausgeführt werden soll, oder konfigurieren Sie `test` so, dass der Befehl nur ausgeführt wird, sofern der Testbefehl den Wert `true` ergibt. Die Ausführung von Container-Befehlen des Leader-only-Typs erfolgt nur während der Umgebungserstellung und -bereitstellung, wohingegen andere Befehle und Serveranpassungen bei jeder Instance-Bereitstellung oder -Aktualisierung ausgeführt werden. Container-Befehle des Leader-only-Typs werden bei Startkonfigurationsänderungen nicht ausgeführt, so z. B. eine Änderung der AMI-ID oder des Instance-Typs.

Syntax

```
container_commands:
  name of container_command:
    command: "command to run"
    leader_only: true
  name of container_command:
    command: "command to run"
```

Optionen

`command`

Eine auszuführende Zeichenfolge bzw. ein Array von Zeichenfolgen.

`env`

(Optional) Legen Sie Umgebungsvariablen vor der Befehlsausführung fest, um alle vorhandenen Werte zu überschreiben.

`cwd`

(Optional) Das Arbeitsverzeichnis. Standardmäßig ist dies das Staging-Verzeichnis der nicht komprimierten Anwendung.

`leader_only`

(Optional) Führen Sie den Befehl nur auf einer einzelnen, von Elastic Beanstalk ausgewählten Instance aus. Container-Befehle des Leader-only-Typs werden vor anderen Container-Befehlen ausgeführt. Ein Befehl kann vom Typ Leader-only oder `test` sein, aber nicht beides (`leader_only` hat Vorrang).

`test`

(Optional) Führen Sie einen Testbefehl aus, der `true` zurückgeben muss, damit der Container-Befehl ausgeführt wird. Ein Befehl kann vom Typ Leader-only oder `test` sein, aber nicht beides (`leader_only` hat Vorrang).

`ignoreErrors`

(Optional) Bereitstellungen schlagen nicht fehl, sofern der Container-Befehl einen anderen Wert als 0 (Erfolg) zurückgibt. Legen Sie den Wert auf `true` fest, um dies zu aktivieren.

Beispielausschnitt

Es folgt ein Beispiel für einen Ausschnitt.

```
container_commands:
  collectstatic:
    command: "django-admin.py collectstatic --noinput"
  01syncdb:
    command: "django-admin.py syncdb --noinput"
    leader_only: true
  02migrate:
    command: "django-admin.py migrate"
    leader_only: true
  99customize:
    command: "scripts/customize.sh"
```

Beispiel: Verwendung benutzerdefinierter CloudWatch Amazon-Metriken

Dieses Thema enthält ein Konfigurationsbeispiel, das Elastic Beanstalk-Metriken mit Amazon CloudWatch Agent für Plattformen integriert, die auf Amazon Linux 2 und höher basieren. Das Konfigurationsbeispiel verwendet Dateien und Befehle in einer `.ebextensions` Konfigurationsdatei.

Amazon CloudWatch ist ein Webservice, mit dem Sie verschiedene Metriken überwachen, verwalten und veröffentlichen sowie Alarmaktionen auf der Grundlage von Daten aus Metriken konfigurieren können. Sie können benutzerdefinierte Metriken für Ihren eigenen Gebrauch definieren, und Elastic Beanstalk überträgt diese Metriken an Amazon. CloudWatch Sobald Amazon Ihre benutzerdefinierten Metriken CloudWatch enthält, können Sie diese in der CloudWatch Amazon-Konsole anzeigen.

Der CloudWatch Amazon-Agent

Der CloudWatch Amazon-Agent ermöglicht die Erfassung von CloudWatch Metriken und Protokollen sowohl von EC2 Amazon-Instances als auch von lokalen Servern über Betriebssysteme hinweg. Der Agent unterstützt Metriken, die auf Systemebene erfasst wurden. Er unterstützt auch die benutzerdefinierte Protokoll- und Metrikenerfassung aus Ihren Anwendungen und Services. Weitere Informationen über den CloudWatch Amazon-Agenten finden Sie im CloudWatch Amazon-Benutzerhandbuch unter Erfassung von Metriken und Protokollen mit dem CloudWatch Agenten.

> **Note**
>
> Elastic Beanstalk Enhanced Health Reporting bietet native Unterstützung für die Veröffentlichung einer Vielzahl von Instanz- und Umgebungsmetriken für. CloudWatch Details dazu finden Sie unter Veröffentlichen CloudWatch benutzerdefinierter Amazon-Metriken für eine Umgebung.

Themen

- .Ebextensions-Konfigurationsdatei
- Berechtigungen
- Metriken in der CloudWatch Konsole anzeigen

.Ebextensions-Konfigurationsdatei

In diesem Beispiel werden Dateien und Befehle in einer .ebextensions-Konfigurationsdatei verwendet, um den CloudWatch Amazon-Agenten auf der Amazon Linux 2-Plattform zu konfigurieren

und auszuführen. Der Agent ist in Amazon Linux 2 enthalten. Wenn Sie ein anderes Betriebssystem verwenden, sind möglicherweise zusätzliche Schritte zur Installation des Agenten erforderlich. Weitere Informationen finden Sie unter Installation des CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.

Damit Sie dieses Beispiel einsetzen können, speichern Sie es in einer Datei mit dem Namen `cloudwatch.config` in einem Verzeichnis mit dem Namen `.ebextensions` im Top-Level-Verzeichnis des Projekts. Dann stellen Sie die Anwendung über die Elastic Beanstalk-Konsole (binden Sie das Verzeichnis .ebextensions in das Quell-Bundle ein) oder die EB CLI bereit.

Weitere Informationen zu Konfigurationsdateien finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

Diese Datei hat zwei Bereiche:

- `files` – In diesem Abschnitt wird die Agentenkonfigurationsdatei hinzugefügt. Es gibt an, welche Metriken und Protokolle der Agent an Amazon senden soll CloudWatch. In diesem Beispiel senden wir nur die mem_used_percent-Metrik. Eine vollständige Liste der vom CloudWatch Amazon-Agenten unterstützten Metriken auf Systemebene finden Sie unter Vom CloudWatch Agenten gesammelte Metriken im CloudWatch Amazon-Benutzerhandbuch.
- `container_commands` – Dieser Abschnitt enthält den Befehl, der den Agenten startet und die Konfigurationsdatei als Parameter übergibt. Weitere Einzelheiten zu `container_commands` finden Sie unter Container-Befehle.

.ebextensions/cloudwatch.config

```
files:
  "/opt/aws/amazon-cloudwatch-agent/bin/config.json":
    mode: "000600"
    owner: root
    group: root
    content: |
      {
        "agent": {
          "metrics_collection_interval": 60,
          "run_as_user": "root"
        },
        "metrics": {
          "namespace": "System/Linux",
          "append_dimensions": {
```

```
            "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
          },
          "metrics_collected": {
            "mem": {
              "measurement": [
                "mem_used_percent"
              ]
            }
          }
        }
      }
container_commands:
  start_cloudwatch_agent:
    command: /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a
 append-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json
```

Berechtigungen

Die Instances in Ihrer Umgebung benötigen die richtigen IAM-Berechtigungen, um benutzerdefinierte CloudWatch Amazon-Metriken mithilfe des CloudWatch Amazon-Agenten zu veröffentlichen. Sie erteilen den Umgebungs-Instances die Berechtigungen, indem Sie diese zum Instance-Profil der Umgebung hinzufügen. Die Berechtigungen für das Instance-Profil können Sie vor oder nach der Anwendungsbereitstellung hinzufügen.

Um Berechtigungen zur Veröffentlichung CloudWatch von Metriken zu erteilen

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen.
3. Wählen Sie die Instance-Profilrolle der Umgebung aus. Wenn Sie eine Umgebung mit der Elastic Beanstalk-Konsole oder der EB CLI erstellen, ist das standardmäßig `aws-elasticbeanstalk-ec2-role`.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Unter Berechtigungsrichtlinien im Abschnitt Berechtigungen wählen Sie Richtlinien anfügen aus.
6. Wählen Sie unter Berechtigungen anhängen die AWS verwaltete Richtlinie aus CloudWatchAgentServerPolicy. Klicken Sie dann auf Attach Policy (Richtlinie anfügen).

Weitere Informationen zum Verwalten von Richtlinien finden Sie unter Arbeiten mit Richtlinien im IAM-Benutzerhandbuch.

Metriken in der CloudWatch Konsole anzeigen

Nachdem Sie die CloudWatch Konfigurationsdatei in Ihrer Umgebung bereitgestellt haben, überprüfen Sie die CloudWatchAmazon-Konsole, um Ihre Messwerte einzusehen. Benutzerdefinierte Metriken werden im CWAgentNamespace gespeichert.

Weitere Informationen finden Sie unter Verfügbare Messwerte anzeigen im CloudWatch Amazon-Benutzerhandbuch.

Anpassen von Software auf Windows-Servern

Möglicherweise muss die Software, von der Ihre Anwendung abhängig ist, angepasst und konfiguriert werden. Bei diesen Dateien kann es sich um von der Anwendung benötigte Abhängigkeiten handeln, z. B. zusätzliche Pakete oder Services, die ausgeführt werden müssen. Allgemeine Informationen zum Anpassen und Konfigurieren der Elastic Beanstalk-Umgebungen finden Sie unter Konfigurieren von Elastic-Beanstalk-Umgebungen.

> **Note**
>
> Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Konfigurationsdateien unterstützen die folgenden Schlüssel, die sich auf die Windows-Server auswirken, auf denen die Anwendung ausgeführt wird.

Schlüssel

- Pakete
- Quellen
- Dateien
- Befehle
- Dienstleistungen
- Container-Befehle

Schlüssel werden in der hier aufgeführten Reihenfolge verarbeitet.

> **Note**
>
> Ältere Versionen (ohne Versionsangabe) von .NET-Plattformversionen können die Konfigurationsdateien nicht in der richtigen Reihenfolge verarbeiten. Weitere Informationen finden Sie unter Migration über die Hauptversionen der Elastic Beanstalk-Windows Server-Plattform hinweg.

Beobachten Sie die Ereignisse in der Umgebung, während Sie Konfigurationsdateien entwickeln und testen. Elastic Beanstalk ignoriert eine Konfigurationsdatei, die Validierungsfehler enthält, z. B. einen ungültigen Schlüssel, und verarbeitet keinen der in der betreffenden Datei enthaltenen Schlüssel. Wenn dies geschieht, fügt Elastic Beanstalk eine Warnung in das Ereignisprotokoll ein.

Pakete

Mit dem Schlüssel `packages` können Sie vorgefertigte Anwendungen und Komponenten herunterladen und installieren.

In Windows-Umgebungen unterstützt Elastic Beanstalk das Herunterladen und Installieren von MSI-Paketen. (Linux-Umgebungen unterstützen zusätzliche Paketmanager. Details dazu finden Sie unter Pakete auf der Seite Anpassen von Software auf Linux-Servern.)

Sie können auf einen beliebigen externen Speicherort verweisen, z. B. ein Amazon Simple Storage Service (Amazon S3)-Objekt, sofern die URL öffentlich zugänglich ist.

Wenn Sie mehrere `msi:` Pakete angeben, wird deren Installationsreihenfolge nicht garantiert.

Syntax

Geben Sie einen Namen Ihrer Wahl als Paketnamen an, und eine URL zu einem MSI-Dateispeicherort als den Wert. Sie können mehrere Pakete unter dem `msi:`-Schlüssel angeben.

```
packages:
  msi:
    package name: package url
    ...
```

Beispiele

Im folgenden Beispiel wird eine URL zum Herunterladen von mysql von `https://dev.mysql.com/` angegeben.

```
packages:
  msi:
    mysql: https://dev.mysql.com/get/Downloads/Connector-Net/mysql-connector-
net-8.0.11.msi
```

Im folgenden Beispiel wird ein Amazon S3-Objekt als MSI-Dateispeicherort angegeben.

```
packages:
  msi:
    mymsi: https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject.msi
```

Quellen

Mit dem Schlüssel `sources` können Sie eine Archivdatei über eine öffentliche URL herunterladen und diese auf der EC2-Instance in einem Zielverzeichnis entpacken.

Syntax

```
sources:
  target directory: location of archive file
```

Unterstützte Formate

In Windows-Umgebungen unterstützt Elastic Beanstalk das ZIP-Format. (Linux-Umgebungen unterstützen zusätzliche Formate. Details dazu finden Sie unter Quellen auf der Seite Anpassen von Software auf Linux-Servern.)

Sie können auf einen beliebigen externen Speicherort verweisen, z. B. ein Amazon Simple Storage Service (Amazon S3)-Objekt, sofern die URL öffentlich zugänglich ist.

Beispiel

Das folgende Beispiel lädt eine öffentliche .zip-Datei von einem Amazon S3 herunter und entpackt sie in. `c:/myproject/myapp`

```
sources:
  "c:/myproject/myapp": https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject.zip
```

Dateien

Mit dem Schlüssel `files` können Sie Dateien auf der EC2-Instance erstellen. Die Inhalte können entweder in der Konfigurationsdatei enthalten sein oder über eine URL abgerufen werden. Die Dateien werden in lexikalischer Reihenfolge auf den Datenträger geschrieben. Wenn private Dateien von Amazon S3 heruntergeladen werden sollen, stellen Sie ein Instance-Profil für die Autorisierung bereit.

Syntax

```
files:
  "target file location on disk":
    source: URL
    authentication: authentication name:

  "target file location on disk":
    content: |
      this is my content
    encoding: encoding format
```

Optionen

`content`

(Optional) Eine Zeichenfolge.

`source`

(Optional) Die URL, über die diese Datei geladen wird. Diese Option kann nicht mit dem Inhaltsschlüssel angegeben werden.

`encoding`

(Optional) Das Codierungsformat. Diese Option wird nur für einen bereitgestellten Inhaltsschlüsselwert verwendet. Der Standardwert ist `plain`.

Zulässige Werte: `plain` | `base64`

`authentication`

(Optional) Der Name der zu verwendenden CloudFormation -Authentifizierungsmethode. Mit dem Ressourcenschlüssel können Sie Authentifizierungsmethoden zu den Metadaten der Auto Scaling-Gruppe hinzufügen.

Beispiele

Das folgende Beispiel zeigt die beiden Möglichkeiten, Dateiinhalt bereitzustellen: von einer URL oder inline in der Konfigurationsdatei.

```
files:
  "c:\\targetdirectory\\targetfile.txt":
    source: http://foo.bar/myfile

  "c:/targetdirectory/targetfile.txt":
    content: |
      # this is my file
      # with content
```

> **Note**
>
> Falls im Dateipfad ein Backslash (\) verwendet wird, muss diesem – wie im vorigen Beispiel – ein weiterer Backslash (das Escape-Zeichen) vorangestellt werden.

Das folgende Beispiel verwendet den Resources-Schlüssel, um eine Authentifizierungsmethode namens S3Auth hinzuzufügen und verwendet ihn, um eine private Datei von einem Amazon S3 herunterzuladen:

```
files:
  "c:\\targetdirectory\\targetfile.zip":
    source: https://elasticbeanstalk-us-east-2-123456789012.s3.amazonaws.com/prefix/
myfile.zip
    authentication: S3Auth

Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          s: ["amzn-s3-demo-bucket"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
```

```
            DefaultValue: "aws-elasticbeanstalk-ec2-role"
```

Befehle

Mit dem Schlüssel `commands` können Sie Befehle auf der EC2-Instance ausführen. Die Befehle werden in alphabetischer Reihenfolge nach Name verarbeitet und ausgeführt, bevor die Anwendung und der Webserver eingerichtet sind und die Anwendungsversionsdatei extrahiert wird.

Die angegebenen Befehle werden als Administrator-Benutzer ausgeführt.

Um Probleme mit Befehlen zu beheben, schlagen Sie deren Ausgabe in den Instance-Protokollen nach.

Syntax

```
commands:
  command name:
    command: command to run
```

Optionen

`command`

Entweder ein Array oder eine Zeichenfolge, das bzw. die den auszuführenden Befehl angibt. Wenn Sie ein Array verwenden, müssen Sie Leerzeichen kein Escape-Zeichen voranstellen und Befehlsparameter nicht in Anführungszeichen angeben.

`cwd`

(Optional) Das Arbeitsverzeichnis. Standardmäßig sucht Elastic Beanstalk nach dem Verzeichnispfad des Projekts. Falls dieser nicht gefunden wird, gilt `c:\Windows\System32` als Standardwert.

`env`

(Optional) Legt Umgebungsvariablen für den Befehl fest. Diese Eigenschaft überschreibt die vorhandene Umgebung, anstatt sie anzuhängen.

`ignoreErrors`

(Optional) Ein boolescher Wert, der bestimmt, ob andere Befehle ausgeführt werden sollen, wenn der im Schlüssel `command` enthaltene Befehl fehlschlägt (gibt einen Wert ungleich Null zurück). Legen Sie diesen Wert auf `true` fest, damit weiterhin Befehle ausgeführt werden, selbst wenn der

Befehl fehlschlägt. Legen Sie den Wert auf `false` fest, um anzugeben, dass keine Befehle mehr ausgeführt werden sollen, wenn der Befehl fehlschlägt. Der Standardwert ist `false`.

`test`

(Optional) Ein Befehl, der den Wert `true` (Beendigungscode 0) zurückgeben muss, damit Elastic Beanstalk den im Schlüssel `command` enthaltenen Befehl verarbeitet.

`waitAfterCompletion`

(Optional) Wartezeit in Sekunden, nachdem ein Befehl abgeschlossen ist, bevor der nächste Befehl ausgeführt wird. Die Standard-Wert ist **`60`** Sekunden. Sie können auch angeben. **`forever`**

> **Important**
>
> Systemneustarts während der Bereitstellung werden nicht unterstützt. Wenn das System aufgrund eines Befehls neu gestartet wird, schlägt die Instanzinitialisierung fehl, sodass die Bereitstellung fehlschlägt.
> Um das Problem zu umgehen, können Sie `.ebextensions configuration`damit Neustarts nach Abschluss der Bereitstellung planen.

Beispiel

Im folgenden Beispiel wird die Ausgabe des Befehls `set` in der angegebenen Datei gespeichert. Wenn ein weiterer Befehl vorhanden ist, führt Elastic Beanstalk diesen unverzüglich nach Abschluss dieses Befehls aus. Falls dieser Befehl einen Neustart erforderlich macht, startet Elastic Beanstalk die Instance direkt nach Abschluss des Befehls neu.

```
commands:
  test:
    command: set > c:\\myapp\\set.txt
    waitAfterCompletion: 0
```

Dienstleistungen

Mit dem Schlüssel `services` definieren Sie, welche Services beim Start der Instance gestartet oder gestoppt werden sollen. Zudem lassen sich mit dem Schlüssel `services` auch Abhängigkeiten von Quellen, Paketen und Dateien angeben, sodass Elastic Beanstalk im Falle eines Neustarts aufgrund von installierten Dateien den Service neu startet.

Syntax

```
services:
  windows:
    name of service:
      files:
        - "file name"
      sources:
        - "directory"
      packages:
        name of package manager:
            "package name[: version]"
      commands:
        - "name of command"
```

Optionen

`ensureRunning`

(Optional) Legen Sie den Wert auf `true` fest, um sicherzustellen, dass der Service ausgeführt wird, nachdem Elastic Beanstalk abgeschlossen wurde.

Legen Sie den Wert auf `false` fest, um sicherzustellen, dass der Service nicht ausgeführt wird, nachdem Elastic Beanstalk abgeschlossen wurde.

Lassen Sie diesen Schlüssel aus, um keine Änderungen am Servicestatus vorzunehmen.

`enabled`

(Optional) Legen Sie den Wert auf `true` fest, um sicherzustellen, dass der Service beim Systemstart automatisch gestartet wird.

Legen Sie den Wert auf `false` fest, um sicherzustellen, dass der Service beim Systemstart nicht automatisch gestartet wird.

Lassen Sie diesen Schlüssel aus, um keine Änderungen an dieser Eigenschaft vorzunehmen.

`files`

Eine Liste von Dateien. Wenn Elastic Beanstalk eine Datei direkt über den Dateienblock ändert, wird der Service neu gestartet.

`sources`

Eine Liste von Verzeichnissen. Wenn Elastic Beanstalk ein Archiv in eines dieser Verzeichnisse erweitert, wird der Service neu gestartet.

`packages`

Eine Zuordnung des Paket-Managers zu einer Liste von Paketnamen. Wenn Elastic Beanstalk eines dieser Pakete installiert oder aktualisiert, wird der Service neu gestartet.

`commands`

Eine Liste von Befehlsnamen. Wenn Elastic Beanstalk den angegebenen Befehl ausführt, wird der Service neu gestartet.

Beispiel

```
services:
  windows:
    myservice:
      enabled: true
      ensureRunning: true
```

Container-Befehle

Mit dem Schlüssel `container_commands` können Sie Befehle ausführen, die sich auf den Anwendungsquellcode auswirken. Container-Befehle werden ausgeführt, nachdem die Anwendung und der Webserver eingerichtet sind und die Anwendungsversionsdatei extrahiert wurde, aber bevor die Anwendungsversion bereitgestellt wird. Nicht-Container-Befehle und andere Anpassungen werden vor der Extraktion des Anwendungsquellcodes ausgeführt.

Container-Befehle werden aus dem Staging-Verzeichnis ausgeführt, aus dem der Quellcode vor der Bereitstellung auf dem Anwendungsserver extrahiert wird. Alle Änderungen, die Sie mithilfe eines Container-Befehls am Quellcode im Staging-Verzeichnis vornehmen, werden bei der Bereitstellung der Quelle am endgültigen Speicherort einbezogen.

Um Probleme mit Container-Befehlen zu beheben, schlagen Sie deren Ausgabe in den Instance-Protokollen nach.

Verwenden Sie den Befehl `leader_only`, wenn der Befehl nur auf einer einzelnen Instance ausgeführt werden soll, oder konfigurieren Sie `test` so, dass der Befehl nur ausgeführt wird, sofern der Testbefehl den Wert `true` ergibt. Die Ausführung von Container-Befehlen des Leader-only-

Typs erfolgt nur während der Umgebungserstellung und -bereitstellung, wohingegen andere Befehle und Serveranpassungen bei jeder Instance-Bereitstellung oder -Aktualisierung ausgeführt werden. Container-Befehle des Leader-only-Typs werden bei Startkonfigurationsänderungen nicht ausgeführt, so z. B. eine Änderung der AMI-ID oder des Instance-Typs.

Syntax

```
container_commands:
  name of container_command:
    command: command to run
```

Optionen

`command`

Eine auszuführende Zeichenfolge bzw. ein Array von Zeichenfolgen.

`env`

(Optional) Legen Sie Umgebungsvariablen vor der Befehlsausführung fest, um alle vorhandenen Werte zu überschreiben.

`cwd`

(Optional) Das Arbeitsverzeichnis. Standardmäßig ist dies das Staging-Verzeichnis der nicht komprimierten Anwendung.

`leader_only`

(Optional) Führen Sie den Befehl nur auf einer einzelnen, von Elastic Beanstalk ausgewählten Instance aus. Container-Befehle des Leader-only-Typs werden vor anderen Container-Befehlen ausgeführt. Ein Befehl kann vom Typ Leader-only oder `test` sein, aber nicht beides (`leader_only` hat Vorrang).

`test`

(Optional) Führen Sie einen Testbefehl aus, der `true` zurückgeben muss, damit der Container-Befehl ausgeführt wird. Ein Befehl kann vom Typ Leader-only oder `test` sein, aber nicht beides (`leader_only` hat Vorrang).

`ignoreErrors`

(Optional) Bereitstellungen schlagen nicht fehl, sofern der Container-Befehl einen anderen Wert als 0 (Erfolg) zurückgibt. Legen Sie den Wert auf `true` fest, um dies zu aktivieren.

`waitAfterCompletion`

(Optional) Wartezeit in Sekunden, nachdem ein Befehl abgeschlossen ist, bevor der nächste Befehl ausgeführt wird. Die Standard-Wert ist **`60`** Sekunden. Sie können auch angeben. **`forever`**

> **Important**
>
> Systemneustarts während der Bereitstellung werden nicht unterstützt. Wenn das System aufgrund eines Befehls neu gestartet wird, schlägt die Instanzinitialisierung fehl, sodass die Bereitstellung fehlschlägt.
> Um das Problem zu umgehen, können Sie `.ebextensions configuration`damit Neustarts nach Abschluss der Bereitstellung planen.

Beispiel

Im folgenden Beispiel wird die Ausgabe des Befehls `set` in der angegebenen Datei gespeichert. Elastic Beanstalk führt den Befehl auf einer Instance aus und startet diese direkt nach Abschluss des Befehls neu.

```
container_commands:
  foo:
    command: set > c:\\myapp\\set.txt
    leader_only: true
    waitAfterCompletion: 0
```

Hinzufügen und Anpassen von Elastic Beanstalk-Umgebungsressourcen

Sie können Ihre Umgebungsressourcen, die Teil Ihrer Elastic Beanstalk-Umgebung sind, anpassen. Beispielsweise möchten Sie vielleicht eine Amazon SQS SQS-Warteschlange und einen Alarm zur Warteschlangentiefe hinzufügen, oder Sie möchten vielleicht einen ElastiCache Amazon-Cluster hinzufügen. Indem Sie Ihrem Quell-Bundle eine Konfigurationsdatei beifügen, können Sie die Umgebung einfach gleichzeitig mit der Bereitstellung der Anwendungsversion anpassen.

Sie können den `Resources` Schlüssel in einer Konfigurationsdatei verwenden, um AWS Ressourcen in Ihrer Umgebung zu erstellen und anzupassen. In Konfigurationsdateien definierte Ressourcen werden der CloudFormation Vorlage hinzugefügt, die zum Starten Ihrer Umgebung verwendet wird. Alle CloudFormation Ressourcentypen werden unterstützt.

> **Note**
>
> Wenn Sie eine Ressource hinzufügen, die nicht von Elastic Beanstalk verwaltet wird, achten Sie darauf, eine Benutzerrichtlinie mit den entsprechenden Berechtigungen für Ihre AWS Identity and Access Management (IAM-) Benutzer hinzuzufügen. Die verwalteten Benutzerrichtlinien, die Elastic Beanstalk bietet, decken nur Berechtigungen für von Elastic Beanstalk verwaltete Ressourcen ab.

Beispielsweise fügt die folgende Konfigurationsdatei einen Auto Scaling-Lebenszyklus-Hook zur Standard-Auto Scaling-Gruppe hinzu, die von Elastic Beanstalk erstellt wurde:

`~/my-app/.ebextensions/as-hook.config`

```
Resources:
  hookrole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument: {
               "Version": "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "autoscaling.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            }
      Policies: [ {
               "PolicyName": "SNS",
               "PolicyDocument": {
                      "Version": "2012-10-17",
                      "Statement": [{
                          "Effect": "Allow",
                          "Resource": "*",
                          "Action": [
                              "sqs:SendMessage",
                              "sqs:GetQueueUrl",
                              "sns:Publish"
                          ]
                        }
                      ]
```

```
                }
              } ]
  hooktopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint: "my-email@example.com"
          Protocol: email
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
      LifecycleTransition: autoscaling:EC2_INSTANCE_TERMINATING
      NotificationTargetARN: { "Ref" : "hooktopic" }
      RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

In diesem Beispiel werden drei Ressourcen definiert, `hookrole`, `hooktopic` und `lifecyclehook`. Die ersten beiden Ressourcen sind eine IAM-Rolle, die Amazon EC2 Auto Scaling die Erlaubnis erteilt, Nachrichten auf Amazon SNS zu veröffentlichen, und ein SNS-Thema, das Nachrichten von der Auto Scaling Scaling-Gruppe an eine E-Mail-Adresse weiterleitet. Elastic Beanstalk erstellt diese Ressourcen mit den angegebenen Eigenschaften und Typen.

Die letzte Ressource, `lifecyclehook`, ist der Lebenszyklus-Hook selbst:

```
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
      LifecycleTransition: autoscaling:EC2_INSTANCE_TERMINATING
      NotificationTargetARN: { "Ref" : "hooktopic" }
      RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

Die Lebenszyklus-Hook-Definition verwendet zwei Funktionen, um die Werte für die Hook-Eigenschaften zu füllen. `{ "Ref" : "AWSEBAutoScalingGroup" }` ruft den Namen der Auto Scaling-Gruppe ab, die von Elastic Beanstalk für die Umgebung erstellt wurde. `AWSEBAutoScalingGroup` ist einer der Standard-Ressourcennamen, die von Elastic Beanstalk bereitgestellt wurden.

Für `AWS::IAM::Role` gibt `Ref` nur den Namen der Rolle, nicht die ARN, zurück. Wenn Sie den ARN für den `RoleARN`-Parameter abrufen möchten, verwenden Sie stattdessen eine andere intrinsische Funktion, `Fn::GetAtt`, mit der jedes Attribut aus einer Ressource abgerufen werden

kann. `RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }` ruft das `Arn`-Attribut aus der `hookrole`-Ressource ab.

`{ "Ref" : "hooktopic" }` ruft den ARN des Amazon SNS-Themas ab, der zuvor in der Konfigurationsdatei erstellt wurde. Der von zurückgegebene Wert `Ref` variiert je nach Ressourcentyp und ist im Abschnitt zum Ressourcentyp im CloudFormation Benutzerhandbuch zu finden. AWS::SNS::Topic

Ändern der Ressourcen, die Elastic Beanstalk für Ihre Umgebung erstellt

Die Ressourcen, die Elastic Beanstalk für Ihre Umgebung erstellt, haben Namen. Mithilfe dieser Namen können Sie über eine Funktion Informationen zu den Ressourcen abrufen oder die Eigenschaften der Ressourcen ändern, um deren Verhalten anzupassen. In diesem Thema werden die AWS Ressourcen beschrieben, die Elastic Beanstalk in den verschiedenen Umgebungstypen verwendet.

> **Note**
>
> Das vorherige Thema Angepasste Ressourcen bietet einige Anwendungsfälle und Beispiele für das Anpassen von Umgebungsressourcen. Weitere Beispiele für Konfigurationsdateien finden Sie auch im späteren Thema Beispiele für benutzerdefinierte Ressourcen.

Webserver-Umgebungen verfügen über die folgenden Ressourcen.

Webserver-Umgebungen

- `AWSEBAutoScalingGroup`(AWS::AutoScaling::AutoScalingGruppe) — Die Auto Scaling Scaling-Gruppe, die mit Ihrer Umgebung verknüpft ist.
- Eine der folgenden beiden Ressourcen.
  - `AWSEBAutoScalingLaunchConfiguration`(AWS::AutoScaling::LaunchConfiguration) — Die Startkonfiguration, die der Auto Scaling Scaling-Gruppe Ihrer Umgebung zugeordnet ist.
  - `AWSEBEC2LaunchTemplate`(AWS::EC2::LaunchTemplate) — Die Amazon EC2 EC2-Startvorlage, die von der Auto Scaling Scaling-Gruppe Ihrer Umgebung verwendet wird.

> **Note**
>
> Wenn in Ihrer Umgebung Funktionen verwendet werden, die Amazon EC2-Startvorlagen erfordern und Ihrer Benutzerrichtlinie die erforderlichen Berechtigungen fehlen, schlägt

das Erstellen oder Aktualisieren der Umgebung möglicherweise fehl. Verwenden Sie die AdministratorAccessvon AWSElastic Beanstalkverwaltete Benutzerrichtlinie oder fügen Sie Ihrer benutzerdefinierten Richtlinie die erforderlichen Berechtigungen hinzu.

- `AWSEBEnvironmentName`(AWS::ElasticBeanstalk::Environment) — Ihre Umgebung.
- `AWSEBSecurityGroup`(AWS::EC2::SecurityGroup) — Die Sicherheitsgruppe, die Ihrer Auto Scaling Scaling-Gruppe zugeordnet ist.
- `AWSEBRDSDatabase`(AWS: :RDS:: DBInstance) — Die mit Ihrer Umgebung verbundene Amazon RDS-DB-Instance (falls zutreffend).

In einer Umgebung mit Load Balancer können Sie auf zusätzliche Ressourcen für den Load Balancer zugreifen. Classic Load Balancer verfügen über eine Ressource für den Load Balancer und über eine weitere Ressource für die ihm zugeordnete Sicherheitsgruppe. Anwendungs- und Netzwerk-Load Balancer verfügen über zusätzliche Ressourcen für den Standard-Listener, die Listener-Regel und die Zielgruppe des Load Balancers.

Umgebungen mit Load Balancing

- `AWSEBLoadBalancer`(AWS::ElasticLoadBalancing::LoadBalancer) — Der klassische Load Balancer Ihrer Umgebung.
- `AWSEBV2LoadBalancer`(AWS::ElasticLoadBalancingV2::LoadBalancer) — Die Anwendung oder der Netzwerk-Loadbalancer Ihrer Umgebung.
- `AWSEBLoadBalancerSecurityGroup`(AWS::EC2::SecurityGroup) — Nur in einer benutzerdefinierten Amazon Virtual Private Cloud (Amazon VPC) der Name der Sicherheitsgruppe, die Elastic Beanstalk für den Load Balancer erstellt. In einer Standard-VPC oder EC2 Classic weist Elastic Load Balancing dem Load Balancer eine Standard-Sicherheitsgruppe zu.
- `AWSEBV2LoadBalancerListener`(AWS::ElasticLoadBalancingV2::Listener) — Ein Listener, der es dem Load Balancer ermöglicht, nach Verbindungsanfragen zu suchen und diese an eine oder mehrere Zielgruppen weiterzuleiten.
- `AWSEBV2LoadBalancerListenerRule`(AWS::ElasticLoadBalancingV2::ListenerRule) — Definiert, auf welche Anfragen ein Elastic Load Balancing Listener reagiert und welche Aktion er ausführt.
- `AWSEBV2LoadBalancerTargetGroup`(AWS::ElasticLoadBalancingV2::TargetGroup) — Eine Elastic Load Balancing Balancing-Zielgruppe, die Anfragen an ein oder mehrere registrierte Ziele weiterleitet, z. B. Amazon EC2 EC2-Instances.

Worker-Umgebungen verfügen über Ressourcen für die SQS-Warteschlange, die eingehende Anfragen puffert, sowie eine Amazon DynamoDB-Tabelle, die von den Instances zur Leader-Wahl verwendet wird.

Worker-Umgebungen

- `AWSEBWorkerQueue`(AWS::SQS::Queue) — Die Amazon SQS SQS-Warteschlange, aus der der Daemon Anfragen abruft, die verarbeitet werden müssen.
- `AWSEBWorkerDeadLetterQueue`(AWS::SQS::Queue) — Die Amazon SQS SQS-Warteschlange, in der Nachrichten gespeichert werden, die nicht zugestellt werden können oder die vom Daemon nicht erfolgreich verarbeitet wurden.
- `AWSEBWorkerCronLeaderRegistry`(AWS::DynamoDB::Table) — Die Amazon DynamoDB-Tabelle, bei der es sich um die interne Registrierung handelt, die vom Daemon für periodische Aufgaben verwendet wird.

Andere CloudFormation Vorlagentasten

Wir haben bereits Schlüssel für Konfigurationsdateien CloudFormation wie `Resources`, `files`, und eingeführt`packages`. Elastic Beanstalk fügt der CloudFormation Vorlage, die Ihre Umgebung unterstützt, den Inhalt von Konfigurationsdateien hinzu, sodass Sie andere CloudFormation Abschnitte verwenden können, um erweiterte Aufgaben in Ihren Konfigurationsdateien auszuführen.

Schlüssel

- Parameters
- Outputs
- Mappings

Parameters

Parameter sind eine Alternative zu den eigenen benutzerdefinierten Optionen von Elastic Beanstalk, die Sie zum Definieren von Werten verwenden können, die Sie an anderer Stelle in Ihren Konfigurationsdateien verwenden. Ebenso wie benutzerdefinierte Optionen können Sie Parameter verwenden, um vom Benutzer konfigurierbare Werte zentral zu erfassen. Im Gegensatz zu benutzerdefinierten Optionen können Sie die API von Elastic Beanstalk nicht verwenden, um Parameterwerte festzulegen, und die Anzahl der Parameter, die Sie in einer Vorlage definieren können, ist begrenzt durch. CloudFormation

Ein Grund, warum Sie Parameter verwenden möchten, besteht darin, dass Ihre Konfigurationsdateien gleichzeitig als CloudFormation Vorlagen dienen. Wenn Sie Parameter anstelle von benutzerdefinierten Optionen verwenden, können Sie die Konfigurationsdatei verwenden, um dieselbe Ressource in einem CloudFormation eigenen Stapel zu erstellen. Beispiel: Sie haben eine Konfigurationsdatei, die ein Amazon EFS-Dateisystem zu Testzwecken zur Umgebung hinzufügt. Dann verwenden Sie dieselbe Datei, um ein unabhängiges Dateisystem zu erstellen, das nicht in den Produktionslebenszyklus Ihrer Umgebung eingebunden ist.

Das folgende Beispiel zeigt die Verwendung von Parametern, um vom Benutzer konfigurierbare Werte zu Beginn einer Konfigurationsdatei zu erfassen.

Example L oadbalancer-accesslogs-existingbucket .config — Parameter

```
Parameters:
  bucket:
    Type: String
    Description: "Name of the Amazon S3 bucket in which to store load balancer logs"
    Default: "amzn-s3-demo-bucket"
  bucketprefix:
    Type: String
    Description: "Optional prefix. Can't start or end with a /, or contain the word
 AWSLogs"
    Default: ""
```

Outputs

Verwenden Sie einen `Outputs`-Block, um Informationen zu den erstellen Ressourcen nach CloudFormation zu exportieren. Anschließend können Sie die `Fn::ImportValue` Funktion verwenden, um den Wert in eine CloudFormation Vorlage außerhalb von Elastic Beanstalk zu übertragen.

Im folgenden Beispiel wird ein Amazon SNS SNS-Thema erstellt und dessen ARN CloudFormation mit dem Namen `NotificationTopicArn` exportiert.

Example sns-topic.config

```
Resources:
  NotificationTopic:
    Type: AWS::SNS::Topic

Outputs:
```

```
  NotificationTopicArn:
    Description: Notification topic ARN
    Value: { "Ref" : "NotificationTopic" }
    Export:
      Name: NotificationTopicArn
```

In einer Konfigurationsdatei für eine andere Umgebung oder einer CloudFormation Vorlage außerhalb von Elastic Beanstalk können Sie die `Fn::ImportValue` Funktion verwenden, um den exportierten ARN abzurufen. In diesem Beispiel wird der exportierte Wert einer Umgebungseigenschaft namens `TOPIC_ARN` zugewiesen.

Example env.config

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    TOPIC_ARN: '`{ "Fn::ImportValue" : "NotificationTopicArn" }`'
```

Mappings

Sie können eine Zuweisung verwenden, um Schlüssel-Wert-Paare nach Namespace sortiert zu speichern. Mit Zuweisungen können Sie Werte, die Sie in Ihren Konfigurationsdateien verwenden, strukturieren oder einen Parameterwert abhängig von einem anderen Wert ändern. mit der folgenden Konfiguration wird beispielsweise der Wert eines Konto-ID-Parameters basierend auf der aktuellen Region festgelegt.

Example L oadbalancer-accesslogs-newbucket .config — Zuordnungen

```
Mappings:
  Region2ELBAccountId:
    us-east-1:
      AccountId: "111122223333"
    us-west-2:
      AccountId: "444455556666"
    us-west-1:
      AccountId: "123456789012"
    eu-west-1:
      AccountId: "777788889999"
...
            Principal:
              AWS:
                ? "Fn::FindInMap"
```

```
                :
                  - Region2ELBAccountId
                  -
                    Ref: "AWS::Region"
                  - AccountId
```

Funktionen

Sie können Funktionen in den Konfigurationsdateien verwenden, um Werte der Ressourceneigenschaften mit Informationen aus anderen Ressourcen oder aus den Einstellungen der Konfigurationsoptionen von Elastic Beanstalk zu füllen. Elastic Beanstalk unterstützt CloudFormation Funktionen (`Ref`,`Fn::GetAtt`,`Fn::Join`) und eine Elastic Beanstalk-spezifische Funktion,. `Fn::GetOptionSetting`

Funktionen

- Punkt
- Fn:: GetAtt
- Fn::Join
- Fn:: GetOptionSetting

Punkt

Wird verwendet`Ref`, um die standardmäßige Zeichenkettendarstellung einer Ressource abzurufen. AWS Der von `Ref` zurückgegebene Wert hängt vom Ressourcentyp – und gelegentlich auch von weiteren Faktoren – ab. Beispielsweise gibt eine Sicherheitsgruppe (AWS::EC2::SecurityGroup) entweder den Namen oder die ID der Sicherheitsgruppe zurück, je nachdem, ob sich die Sicherheitsgruppe in einer standardmäßigen Amazon Virtual Private Cloud (Amazon VPC), einer EC2 klassischen oder einer benutzerdefinierten VPC befindet.

```
{ "Ref" : "resource name" }
```

> **Note**
>
> Weitere Informationen zu den einzelnen Ressourcentypen einschließlich der Rückgabewerte von `Ref` finden Sie unter AWS -Ressourcentypen – Referenz im CloudFormation - Benutzerhandbuch.

Aus dem Beispiel Auto Scaling-Lebenszyklus-Hook:

```
Resources:
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
```

Sie können es auch verwenden`Ref`, um den Wert eines CloudFormation Parameters abzurufen, der an anderer Stelle in derselben Datei oder in einer anderen Konfigurationsdatei definiert wurde.

Fn:: GetAtt

Wird verwendet`Fn::GetAtt`, um den Wert eines Attributs auf einer AWS Ressource abzurufen.

```
{ "Fn::GetAtt" : [ "resource name", "attribute name"] }
```

Aus dem Beispiel Auto Scaling-Lebenszyklus-Hook:

```
Resources:
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

Weitere Informationen finden Sie unter Fn:: GetAtt.

Fn::Join

Mit `Fn::Join` werden Zeichenfolgen mit einem Trennzeichen zusammengefasst. Die Zeichenfolgen können hartcodiert sein oder die Ausgabe von `Fn::GetAtt` oder `Ref` verwenden.

```
{ "Fn::Join" : [ "delimiter", [ "string1", "string2" ] ] }
```

Weitere Informationen finden Sie unter Fn::Join.

Fn:: GetOptionSetting

Mit `Fn::GetOptionSetting` wird der Einstellungswert einer Konfigurationsoption abgerufen, die für die Umgebung gilt.

```
"Fn::GetOptionSetting":
  Namespace: "namespace"
  OptionName: "option name"
  DefaultValue: "default value"
```

Aus dem Beispiel storing private keys:

```
Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          buckets: ["elasticbeanstalk-us-west-2-123456789012"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
              DefaultValue: "aws-elasticbeanstalk-ec2-role"
```

Beispiele für benutzerdefinierte Ressourcen

Nachstehend finden Sie eine Liste mit Beispielen für Konfigurationsdateien, die Sie verwenden können, um Ihre Elastic Beanstalk-Umgebungen anzupassen:

- DynamoDB und CloudWatch SNS
- Elastic Load Balancing und CloudWatch
- ElastiCache
- RDS und CloudWatch
- SQS, SNS und CloudWatch

Unterthemen dieser Seite enthalten einige erweiterte Beispiele für das Hinzufügen und Konfigurieren benutzerdefinierter Ressourcen in einer Elastic Beanstalk-Umgebung.

Beispiele

- Beispiel: ElastiCache
- Beispiel: SQS CloudWatch, und SNS
- Beispiel: DynamoDB CloudWatch, und SNS

Beispiel: ElastiCache

Die folgenden Beispiele fügen den Plattformen EC2-Classic und EC2-VPC (sowohl Standard- als auch benutzerdefinierte Amazon Virtual Private Cloud (Amazon VPC)) einen ElastiCache Amazon-Cluster hinzu. Weitere Informationen zu diesen Plattformen und dazu, wie Sie feststellen können, welche Plattformen EC2 für Ihre Region und Ihr Konto unterstützt, finden Sie unter. AWS https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-platforms.html Sehen Sie dann im Abschnitt in diesem Thema nach, der für Ihre Plattform gilt.

- EC2-Classic-Plattformen
- EC2-VPC (Standard)
- EC2-VPC (Benutzerdefiniert)

EC2-Classic-Plattformen

Dieses Beispiel fügt einer Umgebung mit Instances, die auf der EC2-Classic-Plattform gestartet wurden, einen ElastiCache Amazon-Cluster hinzu. Alle Eigenschaften, die in diesem Beispiel aufgeführt sind, sind die mindestens erforderlichen Eigenschaften, die für jeden Ressourcentyp eingerichtet werden müssen. Sie können das Beispiel unter ElastiCache Beispiel herunterladen.

> **Note**
>
> In diesem Beispiel werden AWS Ressourcen erstellt, für die Ihnen möglicherweise Gebühren berechnet werden. Weitere Informationen zur AWS Preisgestaltung finden Sie unterhttps://aws.amazon.com/pricing/. Einige Dienste sind Teil des AWS kostenlosen Nutzungskontingents. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter https://aws.amazon.com/free/.

In diesem Beispiel gehen Sie wie folgt vor:

1. Erstellen Sie ein `.ebextensions`-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles.
2. Erstellen Sie zwei Konfigurationsdateien mit der Erweiterung `.config` und speichern Sie sie in Ihrem Verzeichnis `.ebextensions`. Eine Konfigurationsdatei definiert die Ressourcen und die andere die Optionen.
3. Stellen Sie Ihre Anwendung in Elastic Beanstalk bereit.

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Erstellen Sie eine Konfigurationsdatei (z. B. `elasticache.config`), in der die Ressourcen definiert werden. In diesem Beispiel erstellen wir den ElastiCache Cluster, indem wir den Namen der ElastiCache Clusterressource (`MyElastiCache`) angeben, ihren Typ deklarieren und dann die Eigenschaften für den Cluster konfigurieren. Das Beispiel verweist auf den Namen der ElastiCache Sicherheitsgruppenressource, die in dieser Konfigurationsdatei erstellt und definiert wird. Als Nächstes erstellen wir eine ElastiCache Sicherheitsgruppe. Wir definieren den Namen für diese Ressource, deklarieren ihren Typ und fügen eine Beschreibung für die Sicherheitsgruppe hinzu. Schließlich haben wir die Eingangsregeln für die ElastiCache Sicherheitsgruppe so festgelegt, dass der Zugriff nur von Instances innerhalb der ElastiCache Sicherheitsgruppe (`MyCacheSecurityGroup`) und der Elastic Beanstalk-Sicherheitsgruppe () aus zulässig ist. `AWSEBSecurityGroup` Der Parametername `AWSEBSecurityGroup` ist ein fester Ressourcennamen von Elastic Beanstalk. Sie `AWSEBSecurityGroup` müssen Ihrer ElastiCache Sicherheitsgruppe Eingangsregeln hinzufügen, damit Ihre Elastic Beanstalk Beanstalk-Anwendung eine Verbindung zu den Instances in Ihrem Cluster herstellen kann. ElastiCache

```
#This sample requires you to create a separate configuration file that defines the
 custom option settings for CacheCluster properties.

Resources:
  MyElastiCache:
    Type: AWS::ElastiCache::CacheCluster
    Properties:
      CacheNodeType:
         Fn::GetOptionSetting:
             OptionName : CacheNodeType
             DefaultValue: cache.m1.small
      NumCacheNodes:
           Fn::GetOptionSetting:
             OptionName : NumCacheNodes
             DefaultValue: 1
      Engine:
           Fn::GetOptionSetting:
             OptionName : Engine
             DefaultValue: memcached
      CacheSecurityGroupNames:
```

```
        - Ref: MyCacheSecurityGroup
  MyCacheSecurityGroup:
    Type: AWS::ElastiCache::SecurityGroup
    Properties:
      Description: "Lock cache down to webserver access only"
  MyCacheSecurityGroupIngress:
    Type: AWS::ElastiCache::SecurityGroupIngress
    Properties:
      CacheSecurityGroupName:
        Ref: MyCacheSecurityGroup
      EC2SecurityGroupName:
        Ref: AWSEBSecurityGroup
```

Weitere Informationen zu den Ressourcen, die in dieser Beispielkonfigurationsdatei verwendet wurden, finden Sie unter folgenden Referenzen:

- AWS::ElastiCache::CacheCluster
- AWS::ElastiCache::SecurityGroup
- AWS::ElastiCache: SecurityGroupIngress

Erstellen Sie eine separate Konfigurationsdatei mit dem Namen `options.config` und definieren Sie benutzerdefinierte Optionseinstellungen.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     CacheNodeType : cache.m1.small
     NumCacheNodes : 1
     Engine : memcached
```

Diese Zeilen weisen Elastic Beanstalk an, die Werte für die Eigenschaften, und Engine aus den CacheNodeType Werten CacheNodeType NumCacheNodes,, und Engine in einer Konfigurationsdatei (options.config in unserem Beispiel) abzurufen NumCacheNodes, die einen option_settings-Abschnitt mit einem aws:elasticbeanstalk:customoption-Abschnitt enthält, der ein Name-Wert-Paar enthält, das den tatsächlich zu verwendenden Wert enthält. Im obigen Beispiel werden folglich die Werte "cache.m1.small", "1" and "memcached" verwendet. Mehr über `Fn::GetOptionSetting` erfahren Sie unter Funktionen.

EC2-VPC (Standard)

Dieses Beispiel fügt einer Umgebung mit Instances, die auf der EC2-VPC-Plattform gestartet wurden, einen ElastiCache Amazon-Cluster hinzu. Dabei gelten vor allem die Informationen in diesem Abschnitt gilt für ein Szenario, in dem EC2 Instances in der Standard-VPC startet. Alle Eigenschaften in diesem Beispiel sind die mindestens erforderlichen Eigenschaften, die für jeden Ressourcentyp eingerichtet werden müssen. Weitere Informationen zur Standardeinstellung VPCs finden Sie unter Ihre Standard-VPC und Subnetze.

> **Note**
>
> In diesem Beispiel werden AWS Ressourcen erstellt, für die Ihnen möglicherweise Gebühren berechnet werden. Weitere Informationen zur AWS Preisgestaltung finden Sie unterhttps://aws.amazon.com/pricing/. Einige Dienste sind Teil des AWS kostenlosen Nutzungskontingents. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter https://aws.amazon.com/free/.

In diesem Beispiel gehen Sie wie folgt vor:

1. Erstellen Sie ein `.ebextensions`-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles.
2. Erstellen Sie zwei Konfigurationsdateien mit der Erweiterung `.config` und speichern Sie sie in Ihrem Verzeichnis `.ebextensions`. Eine Konfigurationsdatei definiert die Ressourcen und die andere die Optionen.
3. Stellen Sie Ihre Anwendung in Elastic Beanstalk bereit.

   Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Benennen Sie nun die Ressourcenkonfigurationsdatei `elasticache.config`. Um den ElastiCache Cluster zu erstellen, gibt dieses Beispiel den Namen der ElastiCache Clusterressource (`MyElastiCache`) an, deklariert ihren Typ und konfiguriert dann die Eigenschaften für den Cluster. Im Beispiel wird auf die ID der Sicherheitsgruppenressource verwiesen, die wir in dieser Konfigurationsdatei erstellt und definiert haben.

Als Nächstes erstellen Sie eine EC2-Sicherheitsgruppe. Wir definieren den Namen für diese Ressource, deklarieren ihren Typ, fügen eine Beschreibung hinzu und legen Ingress-Regeln für die Sicherheitsgruppe fest, um Zugriff nur von Instances innerhalb der Elastic Beanstalk-Sicherheitsgruppe (`AWSEBSecurityGroup`) zu erlauben. (Der Parametername,`AWSEBSecurityGroup`, ist ein fester Ressourcenname, der von Elastic Beanstalk bereitgestellt wird. Sie `AWSEBSecurityGroup` müssen Ihrer ElastiCache Sicherheitsgruppe Eingangsregeln hinzufügen, damit Ihre Elastic Beanstalk Beanstalk-Anwendung eine Verbindung zu den Instances in Ihrem ElastiCache Cluster herstellen kann.)

Die Ingress-Regeln für die EC2-Sicherheitsgruppe definiert auch das IP-Protokoll und die Port-Nummern, auf denen Cache-Knoten Verbindung akzeptieren können. Für Redis ist die Standard-Portnummer 6379.

```
#This sample requires you to create a separate configuration file that defines the
 custom option settings for CacheCluster properties.

Resources:
  MyCacheSecurityGroup:
    Type: "AWS::EC2::SecurityGroup"
    Properties:
      GroupDescription: "Lock cache down to webserver access only"
      SecurityGroupIngress :
        - IpProtocol : "tcp"
          FromPort :
            Fn::GetOptionSetting:
              OptionName : "CachePort"
              DefaultValue: "6379"
          ToPort :
            Fn::GetOptionSetting:
              OptionName : "CachePort"
              DefaultValue: "6379"
          SourceSecurityGroupName:
            Ref: "AWSEBSecurityGroup"
  MyElastiCache:
    Type: "AWS::ElastiCache::CacheCluster"
    Properties:
      CacheNodeType:
        Fn::GetOptionSetting:
          OptionName : "CacheNodeType"
          DefaultValue : "cache.t2.micro"
      NumCacheNodes:
        Fn::GetOptionSetting:
```

```
          OptionName : "NumCacheNodes"
          DefaultValue : "1"
      Engine:
        Fn::GetOptionSetting:
          OptionName : "Engine"
          DefaultValue : "redis"
      VpcSecurityGroupIds:
        -
          Fn::GetAtt:
            - MyCacheSecurityGroup
            - GroupId

Outputs:
  ElastiCache:
    Description : "ID of ElastiCache Cache Cluster with Redis Engine"
    Value :
      Ref : "MyElastiCache"
```

Weitere Informationen zu den Ressourcen, die in dieser Beispielkonfigurationsdatei verwendet wurden, finden Sie unter folgenden Referenzen:

- AWS::ElastiCache::CacheCluster
- AWS::EC2::SecurityGroup

Geben Sie als Nächstes der Optionen-Konfigurationsdatei den Namen `options.config` und definieren Sie benutzerdefinierte Optionseinstellungen.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType : cache.t2.micro
    NumCacheNodes : 1
    Engine : redis
    CachePort : 6379
```

Diese Zeilen informieren Elastic Beanstalk, die Werte für die Eigenschaften `CacheNodeType`, `NumCacheNodes`, `Engine` und `CachePort` aus den Werten `CacheNodeType`, `NumCacheNodes`, `Engine` und `CachePort` in einer Konfigurationsdatei (in diesem Beispiel `options.config`) abzurufen. Diese Datei enthält einen `aws:elasticbeanstalk:customoption`-Abschnitt (unter `option_settings`) mit Namen-Wert-Paaren mit den tatsächlichen zu verwendenden Werten. Im

vorherigen Beispiel würden `cache.t2.micro`, `1`, `redis` und `6379` für diese Werte verwendet. Mehr über `Fn::GetOptionSetting` erfahren Sie unter Funktionen.

EC2-VPC (Benutzerdefiniert)

Wenn Sie eine benutzerdefinierte VPC auf der EC2-VPC-Plattform erstellen und diese als die VPC angeben, in der EC2 Instances startet, unterscheidet sich der Prozess des Hinzufügens eines ElastiCache Amazon-Clusters zu Ihrer Umgebung von dem einer Standard-VPC. Der Hauptunterschied besteht darin, dass Sie eine Subnetzgruppe für den Cluster erstellen müssen. ElastiCache Alle Eigenschaften in diesem Beispiel sind die mindestens erforderlichen Eigenschaften, die für jeden Ressourcentyp eingerichtet werden müssen.

> **Note**
>
> In diesem Beispiel werden AWS Ressourcen erstellt, für die Ihnen möglicherweise Gebühren berechnet werden. Weitere Informationen zur AWS Preisgestaltung finden Sie unterhttps://aws.amazon.com/pricing/. Einige Dienste sind Teil des AWS kostenlosen Nutzungskontingents. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter https://aws.amazon.com/free/.

In diesem Beispiel gehen Sie wie folgt vor:

1. Erstellen Sie ein `.ebextensions`-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles.
2. Erstellen Sie zwei Konfigurationsdateien mit der Erweiterung `.config` und speichern Sie sie in Ihrem Verzeichnis `.ebextensions`. Eine Konfigurationsdatei definiert die Ressourcen und die andere die Optionen.
3. Stellen Sie Ihre Anwendung in Elastic Beanstalk bereit.

   Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Benennen Sie nun die Ressourcenkonfigurationsdatei `elasticache.config`. Um den ElastiCache Cluster zu erstellen, gibt dieses Beispiel den Namen der ElastiCache Clusterressource (`MyElastiCache`) an, deklariert ihren Typ und konfiguriert dann die Eigenschaften für den Cluster. Die Eigenschaften im Beispiel verweisen auf den Namen der Subnetzgruppe für den ElastiCache

Cluster sowie auf die ID der Sicherheitsgruppenressource, die wir in dieser Konfigurationsdatei erstellen und definieren.

Als Nächstes erstellen Sie eine EC2-Sicherheitsgruppe. Wir definieren den Namen für diese Ressource, deklarieren ihren Typ, fügen eine Beschreibung und die VPC-ID hinzu und legen Ingress-Regeln für die Sicherheitsgruppe fest, um Zugriff nur von Instances innerhalb der Elastic Beanstalk-Sicherheitsgruppe (`AWSEBSecurityGroup`) zu erlauben. (Der Parametername,`AWSEBSecurityGroup`, ist ein fester Ressourcenname, der von Elastic Beanstalk bereitgestellt wird. Sie `AWSEBSecurityGroup` müssen Ihrer ElastiCache Sicherheitsgruppe Eingangsregeln hinzufügen, damit Ihre Elastic Beanstalk Beanstalk-Anwendung eine Verbindung zu den Instances in Ihrem ElastiCache Cluster herstellen kann.)

Die Ingress-Regeln für die EC2-Sicherheitsgruppe definiert auch das IP-Protokoll und die Port-Nummern, auf denen Cache-Knoten Verbindung akzeptieren können. Für Redis ist die Standard-Portnummer 6379. Schließlich erstellt dieses Beispiel eine Subnetzgruppe für den Cluster. ElastiCache Wir definieren den Namen für diese Ressource, deklarieren ihren Typ und fügen eine Beschreibung und ID des Subnetzes in der Subnetzgruppe hinzu.

> **Note**
>
> Wir empfehlen, private Subnetze für den ElastiCache Cluster zu verwenden. Weitere Informationen zu einer VPC mit einem privaten Subnetz finden Sie unter https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html.

```
#This sample requires you to create a separate configuration file that defines the
 custom option settings for CacheCluster properties.

Resources:
  MyElastiCache:
    Type: "AWS::ElastiCache::CacheCluster"
    Properties:
      CacheNodeType:
        Fn::GetOptionSetting:
          OptionName : "CacheNodeType"
          DefaultValue : "cache.t2.micro"
      NumCacheNodes:
        Fn::GetOptionSetting:
          OptionName : "NumCacheNodes"
          DefaultValue : "1"
```

```
      Engine:
        Fn::GetOptionSetting:
          OptionName : "Engine"
          DefaultValue : "redis"
      CacheSubnetGroupName:
        Ref: "MyCacheSubnets"
      VpcSecurityGroupIds:
        - Ref: "MyCacheSecurityGroup"
  MyCacheSecurityGroup:
    Type: "AWS::EC2::SecurityGroup"
    Properties:
      GroupDescription: "Lock cache down to webserver access only"
      VpcId:
        Fn::GetOptionSetting:
          OptionName : "VpcId"
      SecurityGroupIngress :
        - IpProtocol : "tcp"
          FromPort :
            Fn::GetOptionSetting:
              OptionName : "CachePort"
              DefaultValue: "6379"
          ToPort :
            Fn::GetOptionSetting:
              OptionName : "CachePort"
              DefaultValue: "6379"
          SourceSecurityGroupId:
            Ref: "AWSEBSecurityGroup"
  MyCacheSubnets:
    Type: "AWS::ElastiCache::SubnetGroup"
    Properties:
      Description: "Subnets for ElastiCache"
      SubnetIds:
        Fn::GetOptionSetting:
          OptionName : "CacheSubnets"
Outputs:
  ElastiCache:
    Description : "ID of ElastiCache Cache Cluster with Redis Engine"
    Value :
      Ref : "MyElastiCache"
```

Weitere Informationen zu den Ressourcen, die in dieser Beispielkonfigurationsdatei verwendet wurden, finden Sie unter folgenden Referenzen:

- AWS::ElastiCache::CacheCluster
- AWS::EC2::SecurityGroup
- AWS::ElastiCache::SubnetGroup

Geben Sie als Nächstes der Optionen-Konfigurationsdatei den Namen `options.config` und definieren Sie benutzerdefinierte Optionseinstellungen.

> **Note**
>
> Im folgenden Beispiel ersetzen Sie die Beispiel-`CacheSubnets`- und -`VpcId`-Werte durch Ihre eigenen Subnetze und die VPC.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType : cache.t2.micro
    NumCacheNodes : 1
    Engine : redis
    CachePort : 6379
    CacheSubnets:
      - subnet-1a1a1a1a
      - subnet-2b2b2b2b
      - subnet-3c3c3c3c
    VpcId: vpc-4d4d4d4d
```

Diese Zeilen informieren Elastic Beanstalk, die Werte für die Eigenschaften `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` und `VpcId` aus den Werten `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` und `VpcId` in einer Konfigurationsdatei (in diesem Beispiel `options.config`) abzurufen. Diese Datei enthält einen `aws:elasticbeanstalk:customoption`-Abschnitt (unter `option_settings`) mit Namen-Wert-Paaren mit den Beispielwerten. Im vorherigen Beispiel würden `cache.t2.micro`, `1`, `redis`, `6379`, `subnet-1a1a1a1a`, `subnet-2b2b2b2b`, `subnet-3c3c3c3c` und `vpc-4d4d4d4d` für diese Werte verwendet. Mehr über `Fn::GetOptionSetting` erfahren Sie unter Funktionen.

Beispiel: SQS CloudWatch, und SNS

In diesem Beispiel wird der Umgebung eine Amazon SQS-Warteschlange sowie ein Alarm für die Warteschlange hinzugefügt. In diesem Beispiel werden die mindestens erforderlichen Eigenschaften

verwendet, die für jede Ressource eingerichtet werden müssen. Sie können das Beispiel unter SQS, SNS und herunterladen. CloudWatch

> **Note**
>
> In diesem Beispiel werden AWS Ressourcen erstellt, für die Ihnen möglicherweise Gebühren berechnet werden. Weitere Informationen zur AWS Preisgestaltung finden Sie unterhttps://aws.amazon.com/pricing/. Einige Dienste sind Teil des AWS kostenlosen Nutzungskontingents. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter https://aws.amazon.com/free/.

In diesem Beispiel gehen Sie wie folgt vor:

1. Erstellen Sie ein `.ebextensions`-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles.
2. Erstellen Sie zwei Konfigurationsdateien mit der Erweiterung `.config` und speichern Sie sie in Ihrem Verzeichnis `.ebextensions`. Eine Konfigurationsdatei definiert die Ressourcen und die andere die Optionen.
3. Stellen Sie Ihre Anwendung in Elastic Beanstalk bereit.

   Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Erstellen Sie eine Konfigurationsdatei (z. B. sqs.config), in der die Ressourcen definiert werden. In diesem Beispiel wird eine SQS-Warteschlange erstellt und die `VisbilityTimeout`-Eigenschaft wird in der `MySQSQueue`-Ressource definiert. Anschließend wird ein SNS-`Topic` erstellt und angegeben, dass die E-Mail im Falle eines ausgelösten Alarms an `someone@example.com` gesendet wird. Schließlich geben wir einen CloudWatch Alarm aus, wenn die Warteschlange mehr als 10 Nachrichten enthält. In der Eigenschaft `Dimensions` werden der Name der Dimension und der Wert, der die Maßeinheit der Dimension darstellt, angegeben. Hier wird `Fn::GetAtt` für die Rückgabe des Werts `QueueName` von `MySQSQueue` verwendet.

```
#This sample requires you to create a separate configuration file to define the custom
 options for the SNS topic and SQS queue.
Resources:
  MySQSQueue:
```

```
    Type: AWS::SQS::Queue
    Properties:
      VisibilityTimeout:
        Fn::GetOptionSetting:
          OptionName: VisibilityTimeout
          DefaultValue: 30
  AlarmTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint:
            Fn::GetOptionSetting:
              OptionName: AlarmEmail
              DefaultValue: "nobody@amazon.com"
          Protocol: email
  QueueDepthAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: "Alarm if queue depth grows beyond 10 messages"
      Namespace: "AWS/SQS"
      MetricName: ApproximateNumberOfMessagesVisible
      Dimensions:
        - Name: QueueName
          Value : { "Fn::GetAtt" : [ "MySQSQueue", "QueueName"] }
      Statistic: Sum
      Period: 300
      EvaluationPeriods: 1
      Threshold: 10
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - Ref: AlarmTopic
      InsufficientDataActions:
        - Ref: AlarmTopic

Outputs :
  QueueURL:
    Description : "URL of newly created SQS Queue"
    Value : { Ref : "MySQSQueue" }
  QueueARN :
    Description : "ARN of newly created SQS Queue"
    Value : { "Fn::GetAtt" : [ "MySQSQueue", "Arn"]}
  QueueName :
    Description : "Name newly created SQS Queue"
```

```
    Value : { "Fn::GetAtt" : [ "MySQSQueue", "QueueName"]}
```

Weitere Informationen zu den Ressourcen, die in dieser Beispielkonfigurationsdatei verwendet wurden, finden Sie unter folgenden Referenzen:

- AWS::SQS::Queue
- AWS::SNS::Topic
- AWS::CloudWatch::Alarm

Erstellen Sie eine separate Konfigurationsdatei mit dem Namen `options.config` und definieren Sie benutzerdefinierte Optionseinstellungen.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     VisibilityTimeout : 30
     AlarmEmail : "nobody@example.com"
```

Diese Zeilen weisen Elastic Beanstalk an, die Werte für die Eigenschaften und Subscription Endpoint aus den VisibilityTimeout Werten VisibilityTimeout und Subscription Endpoint in einer Konfigurationsdatei (options.config in unserem Beispiel) abzurufen, die einen Abschnitt option_settings mit einem aws:elasticbeanstalk:customoption-Abschnitt enthält, der ein Name-Wert-Paar enthält, das den tatsächlich zu verwendenden Wert enthält. Im obigen Beispiel werden folglich die Werte "30" und "nobody@amazon.com" verwendet. Mehr über `Fn::GetOptionSetting` erfahren Sie unter the section called “Funktionen”.

Beispiel: DynamoDB CloudWatch, und SNS

Diese Konfigurationsdatei richtet die DynamoDB-Tabelle als Sitzungshandler für eine PHP-basierte Anwendung ein, die das AWS SDK for PHP 2 verwendet. Wenn Sie dieses Beispiel verwenden möchten, müssen Sie ein IAM-Instance-Profil haben, das zu den Instances in Ihrer Umgebung hinzugefügt und für den Zugriff auf die DynamoDB-Tabelle verwendet wird.

Sie können das in diesem Schritt verwendete Beispiel unter DynamoDB Session Support-Beispiel herunterladen. Das Beispiel enthält die folgenden Dateien:

- Die Beispielanwendung, `index.php`
- Eine Konfigurationsdatei,`dynamodb.config`, um eine DynamoDB-Tabelle und andere AWS Ressourcen zu erstellen und zu konfigurieren und Software auf den EC2-Instances zu installieren, die die Anwendung in einer Elastic Beanstalk Beanstalk-Umgebung hosten

- Die Konfigurationsdatei `options.config`, die die Standardwerte in `dynamodb.config` mit spezifischen Einstellungen für diese bestimmte Installation überschreibt

`index.php`

```
<?php

// Include the SDK using the Composer autoloader
require '../vendor/autoload.php';

use Aws\DynamoDb\DynamoDbClient;

// Grab the session table name and region from the configuration file
list($tableName, $region) = file(__DIR__ . '/../sessiontable');
$tableName = rtrim($tableName);
$region = rtrim($region);

// Create a DynamoDB client and register the table as the session handler
$dynamodb = DynamoDbClient::factory(array('region' => $region));
$handler = $dynamodb->registerSessionHandler(array('table_name' => $tableName,
 'hash_key' => 'username'));

// Grab the instance ID so we can display the EC2 instance that services the request
$instanceId = file_get_contents("http://169.254.169.254/latest/meta-data/instance-id");
?>
<h1>Elastic Beanstalk PHP Sessions Sample</h1>
<p>This sample application shows the integration of the Elastic Beanstalk PHP
container and the session support for DynamoDB from the AWS SDK for PHP 2.
Using DynamoDB session support, the application can be scaled out across
multiple web servers. For more details, see the
<a href="https://aws.amazon.com/php/">PHP Developer Center</a>.</p>

<form id="SimpleForm" name="SimpleForm" method="post" action="index.php">
<?php
echo 'Request serviced from instance ' . $instanceId . '<br/>';
echo '<br/>';

if (isset($_POST['continue'])) {
  session_start();
  $_SESSION['visits'] = $_SESSION['visits'] + 1;
  echo 'Welcome back ' . $_SESSION['username'] . '<br/>';
  echo 'This is visit number ' . $_SESSION['visits'] . '<br/>';
  session_write_close();
```

```
  echo '<br/>';
  echo '<input type="Submit" value="Refresh" name="continue" id="continue"/>';
  echo '<input type="Submit" value="Delete Session" name="killsession"
 id="killsession"/>';
} elseif (isset($_POST['killsession'])) {
  session_start();
  echo 'Goodbye ' . $_SESSION['username'] . '<br/>';
  session_destroy();
  echo 'Username: <input type="text" name="username" id="username" size="30"/><br/>';
  echo '<br/>';
  echo '<input type="Submit" value="New Session" name="newsession" id="newsession"/>';
} elseif (isset($_POST['newsession'])) {
  session_start();
  $_SESSION['username'] = $_POST['username'];
  $_SESSION['visits'] = 1;
  echo 'Welcome to a new session ' . $_SESSION['username'] . '<br/>';
  session_write_close();
  echo '<br/>';
  echo '<input type="Submit" value="Refresh" name="continue" id="continue"/>';
  echo '<input type="Submit" value="Delete Session" name="killsession"
 id="killsession"/>';
} else {
  echo 'To get started, enter a username.<br/>';
  echo '<br/>';
  echo 'Username: <input type="text" name="username" id="username" size="30"/><br/>';
  echo '<input type="Submit" value="New Session" name="newsession" id="newsession"/>';
}
?>
</form>
```

.ebextensions/dynamodb.config

```
Resources:
  SessionTable:
    Type: AWS::DynamoDB::Table
    Properties:
      KeySchema:
        HashKeyElement:
          AttributeName:
            Fn::GetOptionSetting:
              OptionName : SessionHashKeyName
              DefaultValue: "username"
          AttributeType:
```

```
            Fn::GetOptionSetting:
              OptionName : SessionHashKeyType
              DefaultValue: "S"
      ProvisionedThroughput:
        ReadCapacityUnits:
          Fn::GetOptionSetting:
            OptionName : SessionReadCapacityUnits
            DefaultValue: 1
        WriteCapacityUnits:
          Fn::GetOptionSetting:
            OptionName : SessionWriteCapacityUnits
            DefaultValue: 1

  SessionWriteCapacityUnitsLimit:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }, "
write capacity limit on the session table." ]]}
      Namespace: "AWS/DynamoDB"
      MetricName: ConsumedWriteCapacityUnits
      Dimensions:
        - Name: TableName
          Value: { "Ref" : "SessionTable" }
      Statistic: Sum
      Period: 300
      EvaluationPeriods: 12
      Threshold:
          Fn::GetOptionSetting:
            OptionName : SessionWriteCapacityUnitsAlarmThreshold
            DefaultValue: 240
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - Ref: SessionAlarmTopic
      InsufficientDataActions:
        - Ref: SessionAlarmTopic

  SessionReadCapacityUnitsLimit:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }, " read
capacity limit on the session table." ]]}
      Namespace: "AWS/DynamoDB"
      MetricName: ConsumedReadCapacityUnits
      Dimensions:
```

```
        - Name: TableName
          Value: { "Ref" : "SessionTable" }
      Statistic: Sum
      Period: 300
      EvaluationPeriods: 12
      Threshold:
          Fn::GetOptionSetting:
            OptionName : SessionReadCapacityUnitsAlarmThreshold
            DefaultValue: 240
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - Ref: SessionAlarmTopic
      InsufficientDataActions:
        - Ref: SessionAlarmTopic

  SessionThrottledRequestsAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }, ":
requests are being throttled." ]]}
      Namespace: AWS/DynamoDB
      MetricName: ThrottledRequests
      Dimensions:
        - Name: TableName
          Value: { "Ref" : "SessionTable" }
      Statistic: Sum
      Period: 300
      EvaluationPeriods: 1
      Threshold:
        Fn::GetOptionSetting:
          OptionName: SessionThrottledRequestsThreshold
          DefaultValue: 1
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - Ref: SessionAlarmTopic
      InsufficientDataActions:
        - Ref: SessionAlarmTopic

  SessionAlarmTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint:
            Fn::GetOptionSetting:
```

```
              OptionName: SessionAlarmEmail
              DefaultValue: "nobody@amazon.com"
          Protocol: email

files:
  "/var/app/sessiontable":
    mode: "000444"
    content: |
      `{"Ref" : "SessionTable"}`
      `{"Ref" : "AWS::Region"}`

  "/var/app/composer.json":
    mode: "000744"
    content:
      {
        "require": {
           "aws/aws-sdk-php": "*"
        }
      }

container_commands:
 "1-install-composer":
   command: "cd /var/app; curl -s http://getcomposer.org/installer | php"
 "2-install-dependencies":
   command: "cd /var/app; php composer.phar install"
 "3-cleanup-composer":
   command: "rm -Rf /var/app/composer.*"
```

In der Beispielkonfigurationsdatei erstellen wir zunächst die DynamoDB-Tabelle und konfigurieren die Primärschlüssel-Struktur für die Tabelle und die Kapazitätseinheiten, um ausreichend Ressourcen zuzuweisen, um den angeforderten Durchsatz zu bieten. Als Nächstes erstellen wir Alarme für und. CloudWatch `WriteCapacity` `ReadCapacity` Wir erstellen ein SNS-Thema, das eine E-Mail an "nobody@amazon.com" sendet, wenn die Alarmschwellenwerte überschritten werden.

Nachdem wir unsere AWS Ressourcen für unsere Umgebung erstellt und konfiguriert haben, müssen wir die EC2-Instances anpassen. Wir verwenden den `files` Schlüssel, um die Details der DynamoDB-Tabelle an die EC2-Instances in unserer Umgebung zu übergeben und der `composer.json` Datei für das AWS SDK for PHP 2 ein „require“ hinzuzufügen. Schließlich führen wir Container-Befehle aus, um den Composer und die erforderlichen Abhängigkeiten zu installieren und das Installationsprogramm zu entfernen.

`.ebextensions/options.config`

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     SessionHashKeyName                      : username
     SessionHashKeyType                      : S
     SessionReadCapacityUnits                : 1
     SessionReadCapacityUnitsAlarmThreshold  : 240
     SessionWriteCapacityUnits               : 1
     SessionWriteCapacityUnitsAlarmThreshold : 240
     SessionThrottledRequestsThreshold       : 1
     SessionAlarmEmail                       : me@example.com
```

Ersetzen Sie den SessionAlarmEmail Wert durch die E-Mail, an die Alarmbenachrichtigungen gesendet werden sollen. Die `options.config`-Datei enthält die Werte, die für einige der in `dynamodb.config` definierten Variablen verwendet werden. Beispielsweise enthält `dynamodb.config` die folgenden Zeilen:

```
Subscription:
  - Endpoint:
      Fn::GetOptionSetting:
        OptionName: SessionAlarmEmail
        DefaultValue: "nobody@amazon.com"
```

Diese Zeilen, die Elastic Beanstalk anweisen, den Wert für die Endpoint-Eigenschaft aus dem SessionAlarmEmailWert in einer Konfigurationsdatei (`options.config`in unserer Beispielanwendung) abzurufen, die einen option_settings-Abschnitt mit einem aws:elasticbeanstalk:customoption-Abschnitt enthält, der ein Name-Wert-Paar enthält, das den tatsächlich zu verwendenden Wert enthält. Im obigen Beispiel würde das `nobody@amazon.com` bedeuten, dass der Wert zugewiesen würde. SessionAlarmEmail

Weitere Informationen zu den in diesem Beispiel verwendeten CloudFormation Ressourcen finden Sie in den folgenden Referenzen:

- AWS::DynamoDB::Table
- AWS::CloudWatch::Alarm
- AWS::SNS::Topic

Verwenden gespeicherter Elastic Beanstalk-Konfigurationen

Sie können die Umgebungskonfiguration in Amazon Simple Storage Service (Amazon S3) als Objekt speichern und dieses im Rahmen der Umgebungserstellung für andere Umgebungen verwenden oder in einer ausgeführten Umgebung einsetzen. Gespeicherte Konfigurationen sind Vorlagen im YAML-Format, mit denen die Plattformversion, Ebene, Einstellungen für Konfigurationsoptionen und Tags für eine Umgebung definiert werden.

Sie können Tags auf eine gespeicherte Konfiguration anwenden, wenn Sie sie erstellen, und die Tags der vorhandenen gespeicherten Konfigurationen bearbeiten. Die Tags einer gespeicherten Konfiguration sind nicht mit den Tags verbunden, die mit dem `Tags:`-Schlüssel in einer gespeicherten Konfiguration angegeben sind. Letztere werden auf eine Umgebung angewendet, wenn Sie die gespeicherte Konfiguration auf die Umgebung anwenden. Details hierzu finden Sie unter Markieren von gespeicherten Konfigurationen.

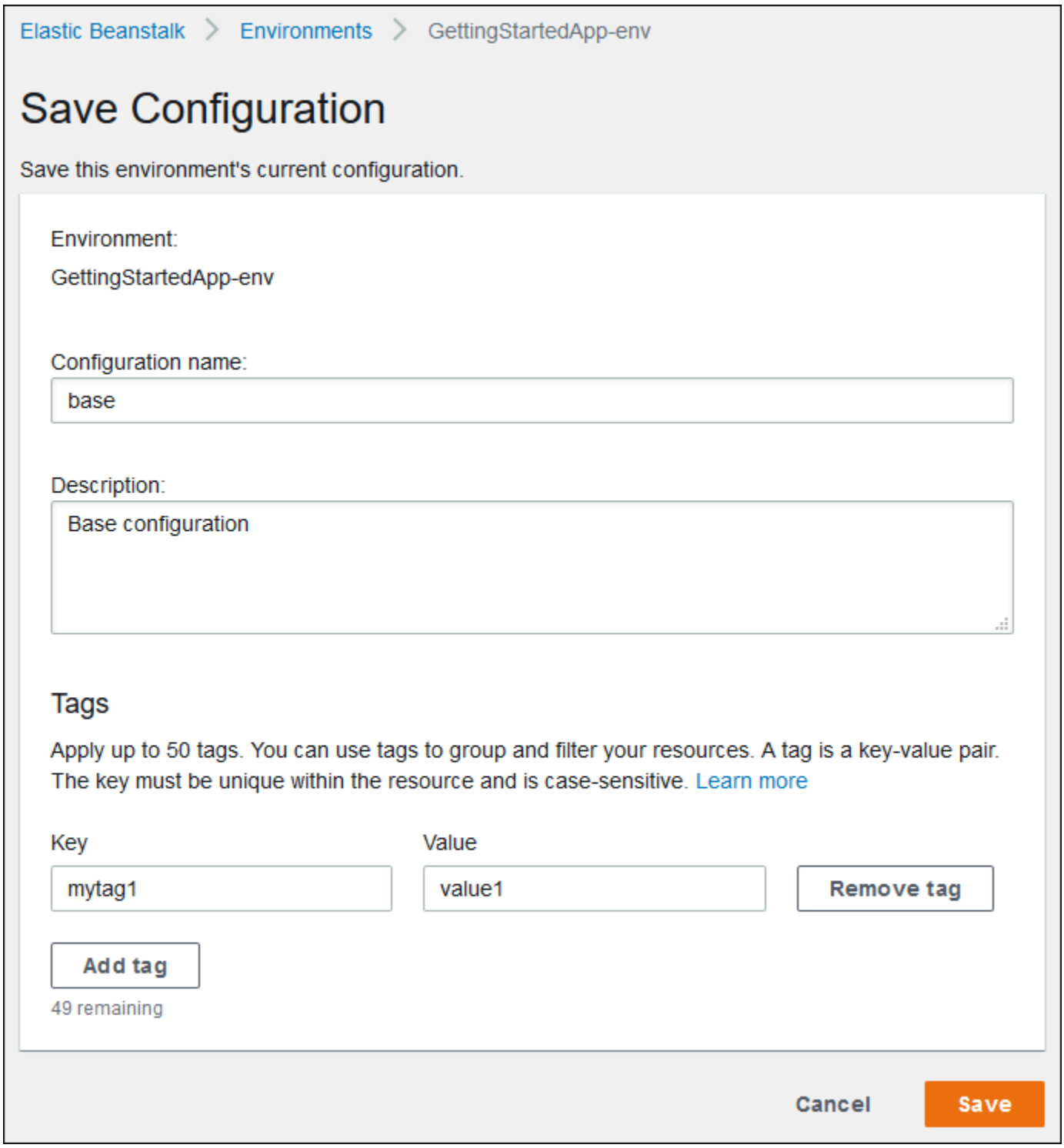
> **Note**
>
> Sie können gespeicherte Konfigurationen mit verschiedenen Methoden erstellen und auf Ihre Elastic Beanstalk-Umgebungen anwenden. Dazu gehören die Elastic Beanstalk-Konsole, die EB-CLI und AWS CLI.
> Weitere Informationen zu Beispiele zu alternativen Methoden zum Erstellen und Verwenden von gespeicherten Konfigurationen finden Sie in den folgenden Themen:
>
> - Festlegen von Konfigurationsoptionen vor der Umgebungserstellung
> - Festlegen von Konfigurationsoptionen während der Umgebungserstellung
> - Festlegen von Konfigurationsoptionen nach der Umgebungserstellung

Erstellen Sie eine gespeicherte Konfiguration des aktuellen Umgebungsstatus mit der Elastic Beanstalk-Managementkonsole.

So speichern Sie die Umgebungskonfiguration

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

3. Wählen Sie Actions (Aktionen) und anschließend Save Configuration (Konfiguration speichern) aus.
4. Verwenden Sie das Formular auf dem Bildschirm, um die gespeicherte Konfiguration zu benennen. Geben Sie optional eine kurze Beschreibung ein und fügen Sie Tag-Schlüssel und -Werte hinzu.
5. Wählen Sie Speichern.



Die gespeicherte Konfiguration enthält alle Einstellungen, die Sie über die Konsole oder einen anderen Client, der die Elastic Beanstalk-API nutzt, für die Umgebung vorgenommen haben. Sie können die gespeicherte Konfiguration zu einem späteren Zeitpunkt nutzen, um den ursprünglichen

Zustand der Umgebung wiederherzustellen, oder Sie können sie während der Umgebungserstellung in eine neue Umgebung übernehmen.

Sie können eine Konfiguration mit dem EB the section called “eb config” CLI-Befehl herunterladen, wie im folgenden Beispiel gezeigt. *NAME*ist der Name Ihrer gespeicherten Konfiguration.

```
eb config get NAME
```

So wenden Sie eine gespeicherte Konfiguration während der Umgebungserstellung an (Elastic-Beanstalk-Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Saved configurations (Gespeicherte Konfigurationen) aus.
4. Wählen Sie die gespeicherte Konfiguration aus, die Sie anwenden möchten, und wählen Sie dann Launch environment (Umgebung starten).
5. Führen Sie die Schritte im Assistenten aus, um die Umgebung zu erstellen.

Gespeicherte Konfigurationen enthalten keine Einstellungen, die mithilfe von Konfigurationsdateien im Anwendungsquellcode übernommen wurden. Falls dieselbe Einstellung sowohl in einer Konfigurationsdatei als auch in einer gespeicherten Konfiguration definiert ist, hat die Einstellung aus der gespeicherten Konfiguration Vorrang. Ebenso überschreiben in der Elastic Beanstalk-Konsole angegebene Optionen die Optionen in gespeicherten Konfigurationen. Weitere Informationen finden Sie unter Precedence.

Gespeicherte Konfigurationen werden im S3-Bucket von Elastic Beanstalk in einem Ordner gespeichert, der nach der Anwendung benannt ist. Beispielsweise befinden sich Konfigurationen für die Anwendung `my-app` in der Region us-west-2 für die Kontonummer 123456789012 unter `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/`.

Zeigen Sie die Inhalte einer gespeicherten Konfiguration in einem Texteditor an. In der folgenden Beispielkonfiguration wird die Konfiguration einer Webserverumgebung, die mit der Elastic Beanstalk-Managementkonsole gestartet wird, dargestellt.

```
EnvironmentConfigurationMetadata:
  Description: Saved configuration from a multicontainer Docker environment created
 with the Elastic Beanstalk Management Console
  DateCreated: '1520633151000'
  DateModified: '1520633151000'
Platform:
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
 Amazon Linux/2.5.0
OptionSettings:
  aws:elasticbeanstalk:command:
    BatchSize: '30'
    BatchSizeType: Percentage
  aws:elasticbeanstalk:sns:topics:
    Notification Endpoint: me@example.com
  aws:elb:policies:
    ConnectionDrainingEnabled: true
    ConnectionDrainingTimeout: '20'
  aws:elb:loadbalancer:
    CrossZone: true
  aws:elasticbeanstalk:environment:
    ServiceRole: aws-elasticbeanstalk-service-role
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: /
  aws:elasticbeanstalk:healthreporting:system:
    SystemType: enhanced
  aws:autoscaling:launchconfiguration:
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    InstanceType: t2.micro
    EC2KeyName: workstation-uswest2
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Health
    RollingUpdateEnabled: true
EnvironmentTier:
  Type: Standard
  Name: WebServer
AWSConfigurationTemplateVersion: 1.1.0.0
Tags:
  Cost Center: WebApp Dev
```

Sie können die Inhalte einer gespeicherten Konfiguration ändern und am gleichen Speicherort in Amazon S3 speichern. Alle ordnungsgemäß formatierten gespeicherten Konfigurationen können –

sofern sie am korrekten Speicherort gespeichert sind – mit der Elastic Beanstalk-Managementkonsole in eine Umgebung übernommen werden.

Folgende Schlüssel werden unterstützt.

- AWSConfigurationTemplateVersion(erforderlich) — Die Version der Konfigurationsvorlage (1.1.0.0).

```
AWSConfigurationTemplateVersion: 1.1.0.0
```

- Plattform – Der Amazon-Ressourcenname (ARN) der Plattformversion für die Umgebung. Sie können die Plattform nach ARN oder Lösungs-Stack-Namen angeben.

```
Platform:
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
 Amazon Linux/2.5.0
```

- SolutionStack— Der vollständige Name des Lösungsstapels, der zur Erstellung der Umgebung verwendet wurde.

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- OptionSettings— Einstellungen der Konfigurationsoptionen, die auf die Umgebung angewendet werden sollen. Beispielsweise wird mit folgendem Eintrag der Instance-Typ auf "t2.micro" festgelegt.

```
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: t2.micro
```

- Tags – Bis zu 47 Tags können für in der Umgebung erstellte Ressourcen verwendet werden.

```
Tags:
  Cost Center: WebApp Dev
```

- EnvironmentTier— Der Typ der zu erstellenden Umgebung. Bei einer Webserverumgebung kann dieser Bereich ausgelassen werden (Webserver ist der Standardwert). Bei einer Worker-Umgebung verwenden Sie Folgendes.

```
EnvironmentTier:
  Name: Worker
```

```
  Type: SQS/HTTP
```

> **Note**
>
> Sie können gespeicherte Konfigurationen mit verschiedenen Methoden erstellen und auf Ihre Elastic Beanstalk-Umgebungen anwenden. Dazu gehören die Elastic Beanstalk-Konsole, die EB-CLI und AWS CLI.
> Weitere Informationen zu Beispiele zu alternativen Methoden zum Erstellen und Verwenden von gespeicherten Konfigurationen finden Sie in den folgenden Themen:
>
> - Festlegen von Konfigurationsoptionen vor der Umgebungserstellung
> - Festlegen von Konfigurationsoptionen während der Umgebungserstellung
> - Festlegen von Konfigurationsoptionen nach der Umgebungserstellung

Markieren von gespeicherten Konfigurationen

Sie können Tags auf Ihre AWS Elastic Beanstalk gespeicherten Konfigurationen anwenden. Tags sind Schlüssel-Wert-Paare, die Ressourcen zugeordnet AWS sind. Weitere Informationen zum Elastic Beanstalk-Ressourcen-Tagging, zu Anwendungsfälle, Einschränkungen für Tag-Schlüssel und -Werte sowie zu unterstützten Ressourcentypen finden Sie unter Markieren von Elastic Beanstalk-Anwendungsressourcen.

Sie können Tags angeben, wenn Sie eine gespeicherte Konfiguration erstellen. In einer vorhandenen gespeicherten Konfiguration können Sie Tags hinzufügen oder entfernen sowie die Werte von vorhandenen Tags aktualisieren. Sie können jeder gespeicherten Konfiguration bis zu 50 Tags.

Hinzufügen von Tags beim Erstellen von gespeicherten Konfigurationen

Wenn Sie die Elastic Beanstalk-Konsole zum Speichern einer Konfiguration verwenden, können Sie Tag-Schlüssel und -Werte auf der Seite Save Configuration (Gespeicherte Konfiguration angeben.

Wenn Sie eine Konfiguration über die EB CLI speichern, verwenden Sie die `--tags`-Option mit dem Befehl eb config, um Tags hinzuzufügen.

```
~/workspace/my-app$ eb config --tags mytag1=value1,mytag2=value2
```

Fügen Sie mit dem AWS CLI oder anderen API-basierten Clients Tags hinzu, indem Sie den `--tags` Parameter im Befehl verwenden. create-configuration-template

```
$ aws elasticbeanstalk create-configuration-template \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
      --application-name my-app --template-name my-template --solution-stack-
name solution-stack
```

Verwalten von Tags einer vorhandenen gespeicherten Konfiguration

Sie können Tags in einer vorhandenen gespeicherten Elastic Beanstalk-Konfiguration hinzufügen, aktualisieren und löschen.

So verwalten Sie die Tags einer gespeicherten Konfiguration mit der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.
3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Saved configurations (Gespeicherte Konfigurationen) aus.
4. Wählen Sie die gespeicherte Konfiguration, die Sie verwalten möchten.
5. Klicken Sie auf Actions (Aktionen) und dann auf Manage Tags (Tags verwalten).
6. Verwenden Sie das Formular auf dem Bildschirm, um Tags hinzuzufügen, zu aktualisieren oder zu löschen.
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Wenn Sie die gespeicherte Konfiguration über die EB CLI aktualisieren, verwenden Sie den Befehl eb tags zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags.

Beispiel: Der folgende Befehl listet die Tags einer gespeicherten Konfiguration auf.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-
account-id:configurationtemplate/my-app/my-template"
```

Mit dem folgenden Befehl wird das Tag `mytag1` aktualisiert und das Tag `mytag2` gelöscht.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:configurationtemplate/my-app/my-template"
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter `eb tags`.

Verwenden Sie bei den AWS CLI oder anderen API-basierten Clients den list-tags-for-resource Befehl, um die Tags einer gespeicherten Konfiguration aufzulisten.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-
template"
```

Verwenden Sie den Befehl update-tags-for-resource zum Hinzufügen, Aktualisieren und Löschen von Tags in einer gespeicherten Konfiguration.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:configurationtemplate/my-app/my-template"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter `--tags-to-add` des Befehls update-tags-for-resource an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

> **Note**
>
> Um einige der EB-CLI und AWS CLI Befehle mit einer gespeicherten Elastic Beanstalk Beanstalk-Konfiguration zu verwenden, benötigen Sie den ARN der gespeicherten Konfiguration. Um den ARN zu erstellen, verwenden Sie zunächst den folgenden Befehl, um den Namen der gespeicherten Konfiguration abzurufen.
>
> ```
> $ aws elasticbeanstalk describe-applications --application-names my-app
> ```
>
> Suchen Sie den `ConfigurationTemplates`-Schlüssel in der Ausgabe des Befehls. Dieses Element zeigt den Namen der gespeicherten Konfiguration. Verwenden Sie diesen Namen, wobei *my-template* in den auf dieser Seite erwähnten Befehlen angegeben ist.

Umgebungsmanifest (`env.yaml`)

Sie können ein Umgebungsmanifest im YAML-Format in den Stamm des Quell-Bundles der Anwendung einbinden und so den Umgebungsnamen, das Lösungs-Stack und Umgebungslinks konfigurieren, die beim Erstellen der Umgebung verwendet werden sollen.

Dieses Dateiformat unterstützt Umgebungsgruppen. Für die Verwendung von Gruppen geben Sie den Umgebungsnamen mit einem Pluszeichen (+) am Ende im Manifest an. Beim Erstellen oder Aktualisieren der Umgebung geben Sie den Gruppennamen mit `--group-name` (AWS CLI) oder mit `--env-group-suffix` (EB CLI) an. Weitere Informationen zu Gruppen finden Sie unter Gruppen von Elastic Beanstalk Umgebungen erstellen oder aktualisieren.

Im folgenden Beispielmanifest wird eine Webserverumgebung mit einem Link zu einer Basiskomponente in einer Worker-Umgebung definiert. Im Manifest werden Gruppen verwendet, damit mehrere Umgebungen mit dem gleichen Quell-Bundle erstellt werden können:

`~/myapp/frontend/env.yaml`

```
AWSConfigurationTemplateVersion: 1.1.0.0
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.6 running Multi-container Docker 1.7.1
 (Generic)
OptionSettings:
  aws:elasticbeanstalk:command:
    BatchSize: '30'
    BatchSizeType: Percentage
  aws:elasticbeanstalk:sns:topics:
    Notification Endpoint: me@example.com
  aws:elb:policies:
    ConnectionDrainingEnabled: true
    ConnectionDrainingTimeout: '20'
  aws:elb:loadbalancer:
    CrossZone: true
  aws:elasticbeanstalk:environment:
    ServiceRole: aws-elasticbeanstalk-service-role
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: /
  aws:elasticbeanstalk:healthreporting:system:
    SystemType: enhanced
  aws:autoscaling:launchconfiguration:
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    InstanceType: t2.micro
    EC2KeyName: workstation-uswest2
```

```
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Health
    RollingUpdateEnabled: true
Tags:
  Cost Center: WebApp Dev
CName: front-A08G28LG+
EnvironmentName: front+
EnvironmentLinks:
  "WORKERQUEUE" : "worker+"
```

Folgende Schlüssel werden unterstützt.

- AWSConfigurationTemplateVersion(erforderlich) — Die Version der Konfigurationsvorlage (1.1.0.0).

  ```
  AWSConfigurationTemplateVersion: 1.1.0.0
  ```

- Plattform – Der Amazon-Ressourcenname (ARN) der Plattformversion für die Umgebung. Sie können die Plattform nach ARN oder Lösungs-Stack-Namen angeben.

  ```
  Platform:
    PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
   Amazon Linux/2.5.0
  ```

- SolutionStack— Der vollständige Name des Lösungsstapels, der zur Erstellung der Umgebung verwendet wurde.

  ```
  SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
  ```

- OptionSettings— Einstellungen der Konfigurationsoptionen, die auf die Umgebung angewendet werden sollen. Beispielsweise wird mit folgendem Eintrag der Instance-Typ auf "t2.micro" festgelegt.

  ```
  OptionSettings:
    aws:autoscaling:launchconfiguration:
      InstanceType: t2.micro
  ```

- Tags – Bis zu 47 Tags können für in der Umgebung erstellte Ressourcen verwendet werden.

  ```
  Tags:
    Cost Center: WebApp Dev
  ```

- EnvironmentTier— Der Typ der zu erstellenden Umgebung. Bei einer Webserverumgebung kann dieser Bereich ausgelassen werden (Webserver ist der Standardwert). Bei einer Worker-Umgebung verwenden Sie Folgendes.

  ```
  EnvironmentTier:
    Name: Worker
    Type: SQS/HTTP
  ```

- CName— Der CNAME für die Umgebung. Binden Sie das Plus-Zeichen (+) am Ende des Namens ein, um Gruppen zu aktivieren.

  ```
  CName: front-A08G28LG+
  ```

- EnvironmentName— Der Name der zu erstellenden Umgebung. Binden Sie das Plus-Zeichen (+) am Ende des Namens ein, um Gruppen zu aktivieren.

  ```
  EnvironmentName: front+
  ```

  Wenn Gruppen aktiviert sind, müssen Sie bei der Umgebungserstellung einen Gruppennamen angeben. Elastic Beanstalk hängt den Gruppennamen mit einem Bindestrich an die Umgebung an. Beispielsweise erstellt Elastic Beanstalk aus dem Umgebungsnamen `front+` und dem Gruppennamen `dev` eine Umgebung mit dem Namen `front-dev`.

- EnvironmentLinks— Eine Karte mit Variablennamen und Umgebungsnamen von Abhängigkeiten. Das folgende Beispiel macht aus der `worker+`-Umgebung eine Abhängigkeit und weist Elastic Beanstalk an, die Link-Informationen in einer Variable namens `WORKERQUEUE` zu speichern.

  ```
  EnvironmentLinks:
    "WORKERQUEUE" : "worker+"
  ```

  Der Wert der Link-Variable variiert abhängig vom Typ der verknüpften Umgebung. In einer Webserverumgebung besteht der Link aus dem CNAME der Umgebung. Bei einer Worker-Umgebung ist der Link der Amazon Simple Queue Service (Amazon SQS)-Warteschlangenname der Umgebung.

Die EnvironmentLinksSchlüssel CName, EnvironmentNameund können verwendet werden, um Umgebungsgruppen und Links zu anderen Umgebungen zu erstellen. Diese Funktionen werden derzeit unterstützt, wenn die EB-CLI AWS CLI oder ein SDK verwendet wird.

Verwenden eines benutzerdefinierten Amazon Machine Images (AMI) in Ihrer Elastic Beanstalk Beanstalk-Umgebung

In diesem Abschnitt wird erläutert, wann die Verwendung eines benutzerdefinierten AMI in Betracht gezogen werden sollte, und es werden die Verfahren zur Konfiguration und Verwaltung des benutzerdefinierten AMI in Ihrer Umgebung beschrieben. Wenn Sie eine AWS Elastic Beanstalk Umgebung erstellen, können Sie ein Amazon Machine Image (AMI) angeben, das anstelle des standardmäßigen Elastic Beanstalk Beanstalk-AMI verwendet werden soll, das in Ihrer Plattformversion enthalten ist. Ein benutzerdefiniertes AMI kann die Bereitstellungszeiten verbessern, wenn Instances in Ihrer Umgebung gestartet werden, wenn Sie viel Software installieren müssen, die nicht im Standard AMIs enthalten ist.

Die Verwendung von Konfigurationsdateien ist effektiv, um Ihre Umgebung schnell und konsistent anzupassen. Das Anwenden von Konfigurationen kann jedoch bei der Erstellung und Aktualisierung der Umgebung viel Zeit in Anspruch nehmen. Wenn Sie viele Server-Konfigurationen in Konfigurationsdateien vornehmen, können Sie diese Zeit verringern, indem Sie eine benutzerdefinierte AMI erstellen, die bereits über die benötigte Software und Konfiguration verfügt.

Mit einem benutzerdefinierten AMI können Sie auch Änderungen an niedrigen Komponenten vornehmen, wie z. B. dem Linux-Kernel, die schwierig zu implementieren sind oder lange für die Anwendung auf Konfigurationsdateien brauchen. Zum Erstellen eines benutzerdefinierten AMI starten Sie ein Elastic Beanstalk-AMI in Amazon EC2, passen die Software und Konfiguration an Ihre Anforderungen an, halten die Instance an und speichern ein AMI daraus.

Erstellen eines benutzerdefinierten AMI

AMIs Als Alternative zu diesen Verfahren können Sie EC2 Image Builder verwenden, um benutzerdefinierte zu erstellen und zu verwalten. Weitere Informationen finden Sie im Image Builder Builder-Benutzerhandbuch.

So identifizieren Sie das Basis-AMI von Elastic Beanstalk

1. Führen Sie in einem Befehlsfenster einen ähnlichen Befehl wie den folgenden aus. Weitere Informationen finden Sie unter describe-platform-version in der Referenz zum AWS CLI -Befehl.

Geben Sie die AWS Region an, in der Sie Ihr benutzerdefiniertes AMI verwenden möchten, und ersetzen Sie den Plattform-ARN und die Versionsnummer durch die Elastic Beanstalk-Plattform, auf der Ihre Anwendung basiert.

Example- Mac OS/Linux OS

```
$ aws elasticbeanstalk describe-platform-version --region us-east-2 \
      --platform-arn "arn:aws:elasticbeanstalk:us-east-2::platform/Node.js 20
 running on 64bit Amazon Linux 2023/6.1.7" \
      --query PlatformDescription.CustomAmiList
[
    {
        "VirtualizationType": "pv",
        "ImageId": ""
    },
    {
        "VirtualizationType": "hvm",
        "ImageId": "ami-020ae06fdda6a0f66"
    }
]
```

Example- Windows OS

```
C:\> aws elasticbeanstalk describe-platform-version --region us-east-2 --platform-
arn"arn:aws:elasticbeanstalk:us-east-2::platform/
IIS 10.0 running on 64bit Windows Server 2022/2.15.3" --query
 PlatformDescription.CustomAmiList
[
    {
        "VirtualizationType": "pv",
        "ImageId": ""
    },
    {
        "VirtualizationType": "hvm",
        "ImageId": "ami-020ae06fdda6a0f66"
    }
]
```

2. Notieren Sie sich den `ImageId`-Wert, der im Ergebnis wie `ami-020ae06fdda6a0f66` aussieht.

Der Wert ist das standardmäßige Elastic Beanstalk Beanstalk-AMI für die Plattformversion, die EC2-Instance-Architektur und die AWS Region, die für Ihre Anwendung relevant sind. Wenn Sie AMIs für mehrere Plattformen, Architekturen oder AWS Regionen erstellen müssen, wiederholen Sie diesen Vorgang, um das richtige Basis-AMI für jede Kombination zu ermitteln.

> **Note**
>
> Erstellen Sie kein AMI aus einer Instance, die in einer Elastic Beanstalk-Umgebung gestartet wurde. Elastic Beanstalk nimmt Änderungen an Instances während der Bereitstellung vor, die zu Problemen im gespeicherten AMI führen können. Durch das Speichern eines Image von einer Instance in einer Elastic Beanstalk-Umgebung wird auch die Version Ihrer Anwendung, die für die Instance bereitgestellt wurde, zu einem festen Teil des Image.

Für Linux ist es auch möglich, ein benutzerdefiniertes AMI von einem Community-AMI zu erstellen, dass nicht von Elastic Beanstalk veröffentlicht wurde. Sie können das neueste Amazon Linux-AMI als Ausgangspunkt verwenden. Wenn Sie eine Umgebung mit einem Linux AMI starten, das nicht von Elastic Beanstalk verwaltet wird, versucht Elastic Beanstalk die Plattformsoftware (Sprache, Framework, Proxy-Server usw.) und zusätzliche Komponenten zur Unterstützung von Funktionen, wie Erweiterte Zustandsberichte zu installieren.

> **Note**
>
> Benutzerdefiniert, die auf Windows Server AMIs basieren, erfordern das standardmäßige Elastic Beanstalk AMI`describe-platform-version`, von dem zurückgegeben wurde, wie zuvor in Schritt 1 gezeigt.

Obwohl Elastic Beanstalk ein AMI verwenden kann, das nicht von Elastic Beanstalk verwaltet wird, kann die erhöhte Bereitstellungszeit, die sich daraus ergibt, dass Elastic Beanstalk fehlende Komponenten installiert, die Vorteile der Erstellung eines benutzerdefinierten AMI im Vorhinein verringern oder aufheben. Andere Linux-Distributionen könnten mit einiger Fehlerbehebung funktionieren, werden aber nicht offiziell unterstützt. Wenn Ihre Anwendung eine spezifische Linux-Distribution erfordert, ist es alternativ auch möglich, ein Docker-Image zu erstellen und es auf der Docker-Plattform oder Multicontainer Docker-Plattform von Elastic Beanstalk auszuführen.

So erstellen Sie ein benutzerdefiniertes AMI

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie Launch Instance aus.
3. Wenn Sie ein Basis-AMI von Elastic Beanstalk (unter Verwendung von `describe-platform-version`) oder ein Amazon Linux-AMI identifiziert haben, geben Sie die AMI-ID in das Suchfeld ein. Drücken Sie anschließend die Eingabetaste.

   Sie können auch die Liste für eine andere Community-AMI durchsuchen, die Ihren Anforderungen entspricht.

   > **Note**
   >
   > Wir empfehlen, dass Sie eine AMI mit HVM-Virtualisierung verwenden. In ihrer AMIs Beschreibung wird der Virtualisierungstyp: hvm angegeben.
   > Weitere Informationen finden Sie unter Virtualisierungstypen im Amazon EC2 EC2-Benutzerhandbuch.

4. Wählen Sie Select (Auswählen) zum Auswählen des AMI aus.
5. Wählen Sie einen Instance-Typen und danach Weiter: Instance-Details konfigurieren aus.
6. (Für veraltete Amazon Linux AMI (AL1) -Plattformen) Überspringen Sie diesen Schritt, wenn Ihre Umgebung auf einer unterstützten Linux-basierten Plattform oder auf einer Windows-Plattform läuft.

   Erweitern Sie den Abschnitt Erweiterte Details und fügen Sie den folgenden Text in das Feld Benutzerdaten ein.

   ```
   #cloud-config
     repo_releasever: repository version number
     repo_upgrade: none
   ```

   Die Repository-Versionsnummer ist das Jahr und der Monat der Version im AMI-Namen. AMIs Basierend auf der Version von Amazon Linux vom März 2015 haben Sie beispielsweise eine Repository-Versionsnummer`2015.03`. Bei einem Elastic Beanstalk-Image entspricht dies dem Datum, das im Namen des Lösungs-Stacks für Ihre Plattformversion auf der Basis von Amazon Linux AMI (Vorgängerversion von Amazon Linux 2) angegeben ist.

> **Note**
>
> Die `repo_releasever` Einstellung konfiguriert die lock-on-launch Funktion für ein Amazon Linux AMI. Dies bewirkt, dass das AMI beim Start eine feste, spezifische Repository-Version verwendet. Diese Funktion wird unter Amazon Linux 2 nicht unterstützt . Geben Sie sie nicht an, wenn Ihre Umgebung einen aktuellen Amazon Linux 2-Plattformzweig verwendet. Die Einstellung ist erforderlich, wenn Sie ein benutzerdefiniertes AMI mit Elastic Beanstalk ausschließlich auf Amazon Linux AMI-Plattformzweigen verwenden (vor Amazon Linux 2).
> Die Einstellung `repo_upgrade` deaktiviert die automatische Installation von Sicherheitsupdates. Dies ist für die Verwendung eines benutzerdefinierten AMI mit Elastic Beanstalk erforderlich.

7. Führen Sie die Schritte im Assistenten aus, um die EC2-Instance zu starten. Wenn Sie dazu aufgefordert werden, wählen Sie ein Schlüsselpaar, auf das Sie Zugriff haben, um die Verbindung zur Instance für die nächsten Schritte herzustellen.
8. Stellen Sie eine Verbindung mit der Instance über SSH oder RDP her.
9. Führen Sie alle gewünschten Anpassungen durch.
10. (Windows-Plattformen) Führen Sie den EC2 Config-Dienst Sysprep aus. Informationen zu EC2 Config finden Sie unter Konfiguration einer Windows-Instanz mithilfe des EC2 Config-Dienstes. Stellen Sie sicher, dass Sysprep konfiguriert ist, um ein zufälliges Passwort zu generieren, das von der AWS-Managementkonsole abgerufen werden kann.
11. Stoppen Sie in der Amazon EC2-Konsole die EC2-Instance. Anschließend wählen Sie im Menü Instance Actions (Instance-Aktionen) die Option Create Image (EBS AMI) (Image erstellen (EBS AMI)) aus.
12. Um zusätzliche AWS Gebühren zu vermeiden, beenden Sie die EC2-Instance.

So verwenden Sie Ihr benutzerdefiniertes AMI in einer Elastic Beanstalk-Umgebung

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Geben Sie bei AMI-ID Ihre benutzerdefinierte AMI-ID ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Wenn Sie eine neue Umgebung mit dem benutzerdefinierten AMI erstellen, sollten Sie dieselbe Plattformversion verwenden, die Sie als Grundlage für das Erstellen des AMI verwendet haben.

Verwaltung einer Umgebung mit einem benutzerdefinierten AMI

Plattformaktualisierungen

Bei Verwendung eines benutzerdefinierten AMI verwendet Elastic Beanstalk weiterhin dasselbe benutzerdefinierte AMI in einer Umgebung, wenn die Plattformversion aktualisiert wird, unabhängig davon, ob das Update manuell oder über verwaltete Plattformupdates angewendet wird. Die Umgebung wird nicht zurückgesetzt, um das Standard-AMI der neuen Plattformversion zu verwenden.

Wir empfehlen, dass Sie ein neues benutzerdefiniertes AMI erstellen, das auf dem Standard-AMI der neuen Plattformversion basiert. Dadurch werden die in der neuen Plattformversion verfügbaren Patches angewendet und außerdem werden Bereitstellungsfehler aufgrund inkompatibler Paket- oder Bibliotheksversionen minimiert.

Weitere Informationen zum Erstellen eines neuen benutzerdefinierten AMI finden Sie weiter Erstellen eines benutzerdefinierten AMI oben in diesem Thema.

Ein benutzerdefiniertes AMI entfernen

Wenn Sie ein benutzerdefiniertes AMI aus einer Umgebung entfernen und es so zurücksetzen möchten, dass es das Standard-AMI für die Plattformversion der Umgebung verwendet, verwenden Sie den folgenden CLI-Befehl.

```
aws elasticbeanstalk update-environment \
  --application-name my-application \
  --environment-name my-environment \
  --region us-east-1 \
  --options-to-remove Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId
```

> **Note**
>
> Um Betriebsunterbrechungen zu vermeiden, testen Sie Ihre Anwendung mit einem Standard-AMI, bevor Sie diese Änderung auf Ihre Produktionsumgebung anwenden.

Bereinigen eines benutzerdefinierten AMI

Wenn Sie mit einem benutzerdefinierten AMI fertig sind und dieses nicht mehr benötigen, um Elastic Beanstalk-Umgebungen zu starten, sollten Sie in Betracht ziehen, es zu bereinigen, um die Speicherkosten zu minimieren. Die Bereinigung eines benutzerdefinierten AMI beinhaltet dessen Abmeldung von Amazon EC2 und das Löschen anderer zugehöriger Ressourcen. Weitere Informationen finden Sie unter Abmelden Ihres Linux AMI oder Abmelden Ihres Windows AMI.

Behalten des Zugriffs auf ein Amazon Machine Image (AMI) für eine außer Betrieb genommene Plattform

Elastic Beanstalk setzt einen Plattformzweig-Status auf außer Betrieb genommen, wenn das Betriebssystem oder die Hauptkomponente, die vom Zweig verwendet wird, das Ende der Lebensdauer erreicht. Das Basis-AMI von Elastic Beanstalk für den Plattformzweig kann auch privat gemacht werden, um die Verwendung dieses out-of-date AMI zu verhindern. Umgebungen, in AMIs denen das als privat deklariert wurde, werden keine Instances mehr starten können.

Wenn Sie Ihre Anwendung nicht in eine unterstützte Umgebung migrieren können, bevor sie außer Betrieb genommen wurde, befindet sich Ihre Umgebung möglicherweise in dieser Situation. Es kann sein, dass eine Umgebung für einen Zweig der Beanstalk-Plattform aktualisiert werden muss, dessen Elastic-Beanstalk-Basis-AMI privat gemacht wurde. Ein alternativer Ansatz ist verfügbar. Sie können eine bestehende Umgebung aktualisieren, die auf einer Kopie des von Ihrer Umgebung verwendeten Elastic-Beanstalk-Basis-AMIs basiert.

In diesem Thema finden Sie einige Schritte und ein eigenständiges Skript zur Aktualisierung einer bestehenden Umgebung auf der Grundlage einer Kopie des von Ihrer Umgebung verwendeten Elastic-Beanstalk-AMI. Sobald Sie Ihre Anwendung zu einer unterstützten Plattform migrieren können, können Sie weiterhin die Standardverfahren für die Wartung Ihrer Anwendung und der unterstützten Umgebungen verwenden.

Manuelle Schritte

So aktualisieren Sie eine Umgebung, die auf einer AMI-Kopie des von Elastic-Beanstalk-Basis-AMI basiert

1. Ermitteln Sie, welches AMI Ihre Umgebung verwendet. Dieser Befehl gibt das AMI zurück, das von der Elastic-Beanstalk-Umgebung verwendet wird, die Sie in den Parametern angeben. Der zurückgegebene Wert wird als source-ami-idim nächsten Schritt verwendet.

   Führen Sie in einem Befehlsfenster einen ähnlichen Befehl wie den folgenden aus. Weitere Informationen finden Sie unter describe-configuration-settings in der Referenz zum AWS CLI -Befehl.

   Geben Sie die AWS Region an, in der das Quell-AMI gespeichert wird, das Sie kopieren möchten. Ersetzen Sie den Anwendungsnamen und den Umgebungsnamen durch diejenigen, die auf dem Quell-AMI basieren. Geben Sie den Text für den Abfrageparameter wie in der Abbildung gezeigt ein.

   Example

   ```
   >aws elasticbeanstalk describe-configuration-settings \
     --application-name my-application \
     --environment-name my-environment \
     --region us-east-2 \
     --query "ConfigurationSettings[0].OptionSettings[?OptionName=='ImageId'] |
    [0].Value"
   ```

2. Kopieren Sie das AMI in Ihr Konto. Dieser Befehl gibt das neue AMI zurück, das sich aus dem Kopieren des source-ami-idim vorherigen Schritt zurückgegebenen ergibt.

   > **Note**
   >
   > Notieren Sie sich unbedingt die neue AMI-ID, die mit diesem Befehl ausgegeben wird. Sie müssen ihn im nächsten Schritt eingeben und ihn copied-ami-idim Beispielbefehl ersetzen.

   Führen Sie in einem Befehlsfenster einen ähnlichen Befehl wie den folgenden aus. Weitere Informationen finden Sie unter copy-image in der AWS CLI -Befehlsreferenz.

Geben Sie die AWS Region des Quell-AMI an, das Sie kopieren möchten (--source-region), und die Region, in der Sie Ihr neues benutzerdefiniertes AMI verwenden möchten (--region). source-ami-id Ersetzen Sie es durch das AMI des Images, das Sie kopieren. Das source-ami-idwurde vom Befehl im vorherigen Schritt zurückgegeben. new-ami-nameErsetzen Sie es durch einen Namen, der das neue AMI in der Zielregion beschreibt. Das Skript, das diesem Verfahren folgt, generiert den neuen AMI-Namen, indem es die Zeichenfolge "Kopie von" an den Anfang des Namens von anhängt. source-ami-id

```
>aws ec2 copy-image \
    --region us-east-2 \
    --source-image-id source-ami-id \
    --source-region us-east-2 \
    --name new-ami-name
```

3. Aktualisieren Sie eine Umgebung, um das kopierte AMI zu verwenden. Nachdem der Befehl ausgeführt wurde, gibt er den Status der Umgebung zurück.

   Führen Sie in einem Befehlsfenster einen ähnlichen Befehl wie den folgenden aus. Weitere Informationen finden Sie unter update-environment in der AWS CLI -Befehlsreferenz.

   Geben Sie die AWS Region der Umgebung und Anwendung an, die Sie aktualisieren möchten. Ersetzen Sie den Anwendungs- und Umgebungsnamen durch die Namen, die Sie copied-ami-idaus dem vorherigen Schritt zuordnen müssen. Ersetzen *copied-ami-id* Sie den Parameter --option-setttings durch die AMI-ID, die Sie in der Ausgabe des vorherigen Befehls notiert haben.

   ```
   >aws elasticbeanstalk update-environment \
     --application-name my-application \
     --environment-name my-environment \
     --region us-east-2 \
     --option-settings
    "Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=copied-
   ami-id"
   ```

> **Note**
>
> Um die Speicherkosten zu minimieren, sollten Sie Ihr benutzerdefiniertes AMI bereinigen, wenn Sie es nicht mehr zum Starten von Elastic-Beanstalk-Umgebungen benötigen. Weitere Informationen finden Sie unter Bereinigen eines benutzerdefinierten AMI.

Eigenständiges Skript

Das folgende Skript liefert dieselben Ergebnisse wie die vorherigen manuellen Schritte. Laden Sie das Skript herunter, indem Sie auf diesen Link klicken: copy_ami_and_update_env.zip.

Skriptquelle: copy_ami_and_update_env.sh

```
#!/bin/bash

set -ue

USAGE="This script is used to copy an AMI used by your Elastic Beanstalk environment
 into your account to use in your environment.\n\n"
USAGE+="Usage:\n\n"
USAGE+="./$(basename $0) [OPTIONS]\n"
USAGE+="OPTIONS:\n"
USAGE+="\t--application-name <application-name>\tThe name of your Elastic Beanstalk
 application.\n"
USAGE+="\t--environment-name <environment-name>\tThe name of your Elastic Beanstalk
 environment.\n"
USAGE+="\t--region <region> \t\t\tThe AWS region your Elastic Beanstalk environment is
 deployed to.\n"
USAGE+="\n\n"
USAGE+="Script Usage Example(s):\n"
USAGE+="./$(basename $0) --application-name my-application --environment-name my-
environment --region us-east-1\n"

if [ $# -eq 0 ]; then
  echo -e $USAGE
  exit
fi

while [[ $# -gt 0 ]]; do
  case $1 in
    --application-name)       APPLICATION_NAME="$2"; shift ;;
```

```
    --environment-name)       ENVIRONMENT_NAME="$2"; shift ;;
    --region)                 REGION="$2"; shift ;;
    *)                        echo "Unknown option $1" ; echo -e $USAGE ; exit ;;
  esac
  shift
done

aws_cli_version="$(aws --version)"
if [ $? -ne 0 ]; then
  echo "aws CLI not found. Please install it: https://docs.aws.amazon.com/cli/latest/
userguide/getting-started-install.html. Exiting."
  exit 1
fi
echo "Using aws CLI version: ${aws_cli_version}"

account=$(aws sts get-caller-identity --query "Account" --output text)
echo "Using account ${account}"

environment_ami_id=$(aws elasticbeanstalk describe-configuration-settings \
  --application-name "$APPLICATION_NAME" \
  --environment-name "$ENVIRONMENT_NAME" \
  --region "$REGION" \
  --query "ConfigurationSettings[0].OptionSettings[?OptionName=='ImageId'] | [0].Value"
 \
  --output text)
echo "Image associated with environment ${ENVIRONMENT_NAME} is ${environment_ami_id}"

owned_image=$(aws ec2 describe-images \
  --owners self \
  --image-ids "$environment_ami_id" \
  --region "$REGION" \
  --query "Images[0]" \
  --output text)
if [ "$owned_image" != "None" ]; then
  echo "${environment_ami_id} is already owned by account ${account}. Exiting."
  exit
fi

source_image_name=$(aws ec2 describe-images \
  --image-ids "$environment_ami_id" \
  --region "$REGION" \
  --query "Images[0].Name" \
  --output text)
if [ "$source_image_name" = "None" ]; then
```

```
  echo "Cannot find ${environment_ami_id}. Please contact AWS support if you need
 additional help: https://aws.amazon.com/support."
  exit 1
fi

copied_image_name="Copy of ${source_image_name}"
copied_ami_id=$(aws ec2 describe-images \
  --owners self \
  --filters Name=name,Values="${copied_image_name}" \
  --region "$REGION" \
  --query "Images[0].ImageId" \
  --output text)
if [ "$copied_ami_id" != "None" ]; then
  echo "Detected that ${environment_ami_id} has already been copied by account
 ${account}. Skipping image copy."
else
  echo "Copying ${environment_ami_id} to account ${account} with name
 ${copied_image_name}"
  copied_ami_id=$(aws ec2 copy-image \
    --source-image-id "$environment_ami_id" \
    --source-region "$REGION" \
    --name "$copied_image_name" \
    --region "$REGION" \
    --query "ImageId" \
    --output text)
  echo "New AMI ID is ${copied_ami_id}"

  echo "Waiting for ${copied_ami_id} to become available"
  aws ec2 wait image-available \
    --image-ids "$copied_ami_id" \
    --region "$REGION"
  echo "${copied_ami_id} is now available"
fi

echo "Updating environment ${ENVIRONMENT_NAME} to use ${copied_ami_id}"
environment_status=$(aws elasticbeanstalk update-environment \
  --application-name "$APPLICATION_NAME" \
  --environment-name "$ENVIRONMENT_NAME" \
  --option-settings
 "Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=
${copied_ami_id}" \
  --region "$REGION" \
  --query "Status" \
  --output text)
```

```
echo "Environment ${ENVIRONMENT_NAME} is now ${environment_status}"

echo "Waiting for environment ${ENVIRONMENT_NAME} update to complete"
aws elasticbeanstalk wait environment-updated \
  --application-name "$APPLICATION_NAME" \
  --environment-names "$ENVIRONMENT_NAME" \
  --region "$REGION"
echo "Environment ${ENVIRONMENT_NAME} update complete"
```

> **Note**
>
> Sie müssen das AWS CLI installiert haben, um das Skript ausführen zu können. Installationsanweisungen finden Sie unter Installieren oder Aktualisieren der aktuellen Version der AWS CLI im Benutzerhandbuch zu AWS Command Line Interface .
> Nach der Installation müssen Sie es auch so konfigurieren AWS CLI, dass es das AWS Konto verwendet, dem die Umgebung gehört. Weitere Informationen finden Sie unter Konfigurieren der AWS CLI im AWS Command Line Interface -Benutzerhandbuch. Das Konto muss außerdem über Berechtigungen zum Erstellen eines AMI und zum Aktualisieren der Elastic-Beanstalk-Umgebung verfügen.

Diese Schritte beschreiben den Prozess, dem das Skript folgt.

1. Drucken Sie das verwendete Konto aus.
2. Ermitteln Sie, welches AMI von der Umgebung verwendet wird (Quell-AMI).
3. Prüfen Sie, ob das Quell-AMI bereits dem Konto gehört. Falls ja, beenden Sie das Programm.
4. Ermitteln Sie den Namen des Quell-AMI, damit er im neuen AMI-Namen verwendet werden kann. Dies dient auch dazu, den Zugriff auf das Quell-AMI zu bestätigen.
5. Prüfen Sie, ob das Quell-AMI bereits in das Konto kopiert wurde. Dazu wird nach AMIs dem Namen des kopierten AMI gesucht, das dem Konto gehört. Wenn der AMI-Name zwischen den Skriptausführungen geändert wurde, wird das Image erneut kopiert.
6. Wenn das Quell-AMI noch nicht kopiert wurde, kopieren Sie das Quell-AMI in das Konto und warten Sie, bis das neue AMI verfügbar ist.
7. Aktualisieren Sie die Umgebungskonfiguration, um das neue AMI zu verwenden.
8. Warten Sie, bis die Aktualisierung der Umgebung abgeschlossen ist.

Nachdem Sie das Skript aus der Datei copy_ami_and_update_env.zip extrahiert haben, führen Sie es aus, indem Sie das folgende Beispiel ausführen. Ersetzen Sie den Anwendungsnamen und den Umgebungsnamen im Beispiel durch Ihre eigenen Werte.

```
>sh copy_ami_and_update_env.sh \
  --application-name my-application \
  --environment-name my-environment \
  --region us-east-1
```

> **Note**
>
> Um die Speicherkosten zu minimieren, sollten Sie Ihr benutzerdefiniertes AMI bereinigen, wenn Sie es nicht mehr zum Starten von Elastic-Beanstalk-Umgebungen benötigen. Weitere Informationen finden Sie unter Bereinigen eines benutzerdefinierten AMI.

Bereitstellen von statischen Dateien

Um die Leistung zu verbessern, können Sie den Proxy-Server so konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, anstatt die Anfrage an Ihre Anwendung weiterzuleiten.

Elastic Beanstalk unterstützt die Konfiguration des Proxys zur Bereitstellung statischer Dateien für die meisten auf Amazon Linux 2 basierenden Branches. Die einzige Ausnahme ist Docker.

> **Note**
>
> Elastic Beanstalk konfiguriert auf den Python- und Ruby-Plattformen standardmäßig einige statische Dateiordner. Weitere Informationen finden Sie in den Abschnitten zur Konfiguration statischer Dateien für Python und Ruby. Sie können zusätzliche Ordner konfigurieren, wie auf dieser Seite erläutert.

Konfigurieren statischer Dateien über die Konsole

So konfigurieren Sie den Proxy-Server für die Bereitstellung statischer Dateien

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Scrollen Sie zum Abschnitt Platform software (Plattformsoftware) und suchen Sie die Gruppe Static files (Statische Dateien).

   a. Um eine statische Dateizuordnung hinzuzufügen, wählen Sie Add static files (Statische Dateien hinzufügen) aus. In der zusätzlichen Zeile, die angezeigt wird, geben Sie einen Pfad für die Bereitstellung statischer Dateien und das Verzeichnis ein, das die zu bereitstellenden statischen Dateien enthält.

      - Beginnen Sie im Feld Pfad den Pfadnamen mit einem Schrägstrich (/) (z. B. „/images“).
      - Geben Sie im Feld Directory (Verzeichnis) einen Verzeichnisnamen an, der sich im Stammverzeichnis des Quellcodes Ihrer Anwendung befindet. Beginnen Sie ihn nicht mit einem Schrägstrich (zum Beispiel „static/image-files“).

      > **Note**
      >
      > Wenn Ihnen der Abschnitt Static Files (Statische Dateien) nicht angezeigt wird, müssen Sie mittels einer Konfigurationsdatei mindestens eine Zuweisung hinzufügen. Weitere Informationen finden Sie unter the section called “Konfigurieren statischer Dateien mithilfe von Konfigurationsoptionen” auf dieser Seite.

   b. Um eine Zuordnung zu entfernen, klicken Sie auf Remove (Entfernen).
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Konfigurieren statischer Dateien mithilfe von Konfigurationsoptionen

Sie können eine Konfigurationsdatei verwenden, um statische Dateipfade und Verzeichnisspeicherorte mithilfe von Konfigurationsoptionen zu konfigurieren. Sie können dem Quellpaket Ihrer Anwendung eine Konfigurationsdatei hinzufügen und sie während der Erstellung der Umgebung oder einer späteren Bereitstellung bereitstellen.

Wenn Ihre Umgebung einen Plattformvariante basierend auf Amazon Linux 2 verwendet, verwenden Sie den Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles`.

In der folgenden Beispielkonfigurationsdatei wird der Proxyserver angewiesen, Dateien im Ordner `statichtml` auf dem Pfad `/html` und Dateien im Ordner `staticimages` auf dem Pfad `/images` bereitzustellen.

Example.ebextensions/static-files.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /html: statichtml
    /images: staticimages
```

Wenn Ihre Elastic Beanstalk-Umgebung eine Amazon Linux-AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die folgenden zusätzlichen Informationen:

Plattformspezifische Namespaces für Amazon Linux-AMI

Auf Amazon Linux AMI-Plattformvarianten variieren Namespaces für statische Dateien je nach Plattform. Details finden Sie auf einer der folgenden Seiten:

- Namespace der Go-Konfiguration
- Java SE-Konfigurations-Namespace
- Tomcat-Konfigurations-Namespaces
- Node.js-Konfigurations-Namespace
- Namespaces der Python-Konfiguration

Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung

In diesen Themen in diesem Abschnitt wird erklärt, wie Sie HTTPS für Ihre Elastic Beanstalk Beanstalk-Umgebung konfigurieren. HTTPS ist ein Muss für jede Anwendung, die Benutzerdaten oder Anmeldeinformationen überträgt.

Wenn Sie einen benutzerdefinierten Domänennamen für Ihre Elastic Beanstalk-Umgebung gekauft und konfiguriert haben, können Sie HTTPS verwenden, damit die Benutzer eine sichere Verbindung mit Ihrer Website herstellen können.

Wenn Sie nicht über einen Domänennamen verfügen, können Sie HTTPS trotzdem mit einem selbstsignierten Zertifikat für Entwicklungs- und Testzwecke verwenden. Weitere Informationen finden Sie unter Serverzertifikate.

Konfiguration der HTTPS-Terminierung am Load Balancer

Ein Load Balancer verteilt Anfragen an die EC2-Instances, auf denen Ihre Anwendung ausgeführt wird. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden. Die einfachste Methode, HTTPS mit einer Elastic Beanstalk-Multiinstanzumgebung zu verwenden, besteht darin, einen sicheren Listener für den Load Balancer zu konfigurieren. Die Verbindung zwischen dem Client und dem Load Balancer bleibt sicher, sodass Sie den Load Balancer so konfigurieren können, dass HTTPS beendet wird. Die Back-End-Verbindungen zwischen dem Load Balancer und den EC2-Instances verwenden HTTP, sodass keine zusätzliche Konfiguration der Instances erforderlich ist. Eine ausführliche Anleitung zur Konfiguration eines sicheren Listenters finden Sie unter. Konfiguration der HTTPS-Terminierung am Load Balancer

Konfiguration der HTTPS-Terminierung auf der EC2-Instance

Wenn Sie Ihre Anwendung in einer Umgebung mit einer einzelnen Instance ausführen oder die Verbindung bis zu den EC2-Instances hinter dem Load Balancer sichern müssen, können Sie den Proxyserver, der auf der Instance ausgeführt wird, so konfigurieren, dass HTTPS beendet wird. Das Konfigurieren Ihrer Instances für das Beenden von HTTPS-Verbindungen erfordert die Nutzung von Konfigurationsdateien zum Ändern der Software, die auf den Instances ausgeführt wird, und die Änderung von Sicherheitsgruppen, um sichere Verbindungen zuzulassen. Weitere Informationen finden Sie unter Konfiguration der HTTPS-Terminierung auf der Instanz.

Konfiguration von HTTPS end-to-end

Für end-to-end HTTPS in einer Umgebung mit Lastenausgleich können Sie Instance- und Load Balancer-Terminierung kombinieren, um beide Verbindungen zu verschlüsseln. Wenn Sie den Load

Balancer so konfigurieren, dass er Datenverkehr mithilfe von HTTPS weiterleitet, wird standardmäßig jedem Zertifikat vertraut, das ihm von den Backend-Instances vorgelegt wird. Zur Maximierung der Sicherheit können Sie Richtlinien an den Load Balancer anfügen, die verhindern, dass eine Verbindung mit Instances hergestellt wird, die kein öffentliches Zertifikat darstellen, dem er vertraut. Weitere Informationen finden Sie unter Konfiguration der end-to-end Verschlüsselung in einer Elastic Beanstalk Beanstalk-Umgebung mit Lastenausgleich.

Konfiguration von HTTPS mit TCP-Passthrough

Sie können den Load Balancer auch so konfigurieren, dass er HTTPS-Datenverkehr ohne Entschlüsselung weiterleitet. Weitere Informationen finden Sie unter Konfigurieren des Load Balancers der Umgebung für TCP-Pass-Through.

> **Note**
>
> Hat es Schlangen? Die Beispielanwendung on GitHub enthält Konfigurationsdateien und Anweisungen für jede Methode zur Konfiguration von HTTPS mit einer Tomcat-Webanwendung. Weitere Informationen dazu finden Sie in der Readme-Datei und den HTTPS-Anweisungen.

Themen

- Serverzertifikate
- Konfiguration der HTTPS-Terminierung am Load Balancer
- Konfiguration der HTTPS-Terminierung auf der Instanz
- Konfiguration der end-to-end Verschlüsselung in einer Elastic Beanstalk Beanstalk-Umgebung mit Lastenausgleich
- Konfigurieren des Load Balancers der Umgebung für TCP-Pass-Through
- Konfigurieren der Umleitung von HTTP zu HTTPS

Serverzertifikate

In diesem Thema werden die verschiedenen Arten von Zertifikaten beschrieben, die Sie zur Konfiguration von HTTPS verwenden können, und wann die einzelnen Zertifikate jeweils angewendet werden müssen. Die Unterthemen in diesem Abschnitt enthalten Anweisungen zum Erstellen Ihres eigenen Zertifikats und zum Hochladen.

AWS Certificate Manager (ACM)

Für die Bereitstellung und Verwaltung von Serverzertifikaten empfehlen wir ACM zu verwenden. Sie können dies programmgesteuert oder mit dem tun. AWS CLI Mit ACM können Sie kostenlos ein vertrauenswürdiges Zertifikat für Ihre Domainnamen erstellen.

ACM-Zertifikate können nur mit AWS Load Balancern und CloudFront Amazon-Distributionen verwendet werden, und ACM ist nur in bestimmten Regionen verfügbar. AWS Informationen zur Verwendung eines ACM-Zertifikats mit Elastic Beanstalk finden Sie unter Konfiguration der HTTPS-Terminierung am Load Balancer. Weitere Informationen zu ACM finden Sie im Benutzerhandbuch.AWS Certificate Manager

> **Note**
>
> Eine Liste der Regionen, in denen ACM verfügbar ist, finden Sie unter ACM-Endpunkte und Kontingente im. Allgemeine Amazon Web Services-Referenz

Wenn ACM in Ihrer AWS Region nicht verfügbar ist, können Sie ein Drittanbieter-Zertifikat oder ein selbstsigniertes Zertifikat und einen privaten Schlüssel auf (IAM) hochladen. AWS Identity and Access Management Sie können den verwenden, AWS CLI um das Zertifikat hochzuladen. In IAM gespeicherte Zertifikate können mit Load Balancern und CloudFront Distributionen verwendet werden. Weitere Informationen finden Sie unter Hochladen eines Zertifikats in IAM.

Zertifikat eines Drittanbieters

Wenn ACM in Ihrer Region nicht verfügbar ist, können Sie ein vertrauenswürdiges Zertifikat von einem Drittanbieter kaufen. Ein Drittanbieter-Zertifikat kann verwendet werden, um HTTPS-Datenverkehr auf Ihrem Load Balancer, auf den Backend-Instances oder beiden zu entschlüsseln.

Selbstsigniertes Zertifikat

Für die Entwicklung und das Testen können Sie selbst ein Zertifikat erstellen und signieren. Verwenden Sie dazu Open Source-Tools. Selbstsignierte Zertifikate können kostenlos und einfach erstellt werden, sie können aber nicht für die Front-End-Entschlüsselung auf öffentlichen Websites verwendet werden. Wenn Sie versuchen, ein selbstsigniertes Zertifikat für eine HTTPS-Verbindung mit einem Client zu verwenden, zeigt der Browser des Benutzers eine Fehlermeldung an, die darauf hinweist, dass Ihre Website nicht sicher ist. Sie können jedoch ein selbstsigniertes Zertifikat verwenden, um Backend-Verbindungen ohne Probleme zu sichern.

Erstellen und Signieren eines X.509-Zertifikats

Sie können ein X.509-Zertifikat für Ihre Anwendung mit `OpenSSL` erstellen. OpenSSL ist eine Standard-Open-Source-Bibliothek, die eine Vielzahl von kryptografischen Funktionen unterstützt, einschließlich der Erstellung und Signierung von x509-Zertifikaten. Weitere Informationen zu OpenSSL finden Sie unter www.openssl.org.

> **Note**
>
> Sie müssen nur ein Zertifikat lokal erstellen, wenn Sie HTTPS in einer Einzel-Instance-Umgebung verwenden oder auf dem Backend neu verschlüsseln möchten mit einem selbstsignierten Zertifikat. Wenn Sie einen Domainnamen besitzen, können Sie mithilfe von AWS Certificate Manager (ACM) kostenlos ein Zertifikat in einer Umgebung mit Lastenausgleich erstellen AWS und verwenden. Weitere Informationen finden Sie unter Request a Certificate (Anfordern eines Zertifikats) im AWS Certificate Manager Benutzerhandbuch.

Führen Sie `openssl version` in der Befehlszeile aus, um zu prüfen, ob Sie bereits OpenSSL installiert haben. Wenn Sie dies nicht tun, können Sie den Quellcode anhand der Anweisungen im öffentlichen GitHub Repository erstellen und installieren oder Ihren bevorzugten Paketmanager verwenden. OpenSSL ist auch auf den Linux-Images von Elastic Beanstalk installiert. Eine schnelle Alternative besteht also darin, mithilfe des Befehls der EB-CLI eine Verbindung zu einer EC2 Instance in einer laufenden Umgebung herzustellen: eb ssh

```
~/eb$ eb ssh
[ec2-user@ip-255-55-55-255 ~]$ openssl version
OpenSSL 1.0.1k-fips 8 Jan 2015
```

Sie müssen einen privaten RSA-Schlüssel erstellen, um Ihre Zertifikatsignieranforderung zu erstellen. Zum Erstellen Ihres privaten Schlüssels verwenden Sie den Befehl openssl genrsa:

```
[ec2-user@ip-255-55-55-255 ~]$ openssl genrsa 2048 > privatekey.pem
Generating RSA private key, 2048 bit long modulus
.....................................................................................
+++
..............+++
e is 65537 (0x10001)
```

privatekey.pem

Der Name der Datei, in der Sie den privaten Schlüssel speichern möchten. Normalerweise zeigt der Befehl openssl genrsa den Inhalt des privaten Schlüssels auf dem Bildschirm an; dieser Befehl übergibt die Ausgabe jedoch in eine Datei. Wählen Sie einen beliebigen Dateinamen und speichern Sie die Datei an einem sicheren Ort, sodass Sie sie später abrufen können. Wenn Sie den privaten Schlüssel verlieren, können Sie das Zertifikat nicht mehr verwenden.

Eine CSR ist eine Datei, die Sie an eine Zertifizierungsstelle senden, um ein digitales Serverzertifikat zu erhalten. Zum Erstellen einer CSR verwenden Sie den Befehl openssl req:

```
$ openssl req -new -key privatekey.pem -out csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Geben Sie die erforderlichen Informationen ein und drücken Sie Enter (Eingabetaste). Die folgende Tabelle beschreibt und enthält Beispiele für jedes Feld.

Name	Beschreibung	Beispiel
Ländername	Die zweistellige ISO-Abkürzung für Ihr Land	US = USA
Bundesstaat oder Provinz	Der Name des Bundesstaats oder der Provinz, in dem bzw. der sich Ihre Organisat ion befindet. Dieser Name darf nicht abgekürzt werden.	Washington
Locality Name	Der Name der Stadt, in der sich Ihre Organisation befindet.	Seattle
Name der Organisation	Der vollständige, offizielle Name Ihrer Organisation. Kürzen Sie den Namen Ihrer Organisation nicht ab.	Beispielunternehmen

Name	Beschreibung	Beispiel
Organisatorische Einheit	Optional, für weitere Informationen zu Ihrer Organisation.	Marketing
Common Name	Der vollqualifizierte Domänenname für Ihre Website. Dieser muss mit dem Domänenna men übereinstimmen, den Benutzer sehen, wenn sie Ihre Website besuchen. Andernfalls werden Zertifikatfehler angezeigt.	www.example.com
E-Mail-Adresse	Die E-Mail-Adresse des Websiteadministrat ors.	someone@example.com

Sie können die Signierungsanforderung zur Signierung an einen Dritten weitergeben oder sie für Entwicklung und Tests selbst signieren. Selbstsignierte Zertifikate können auch für Backend-HTTPS zwischen einem Load Balancer und Instanzen verwendet werden. EC2

Zum Signieren des Zertifikats verwenden Sie den Befehl openssl x509. Im folgenden Beispiel werden der private Schlüssel aus dem vorherigen Schritt (*privatekey.pem*) und die Signaturanforderung (*csr.pem*) verwendet, um ein öffentliches Zertifikat mit dem Namen zu erstellen*public.crt*, das tagelang gültig ist. *365*

```
$ openssl x509 -req -days 365 -in csr.pem -signkey privatekey.pem -out public.crt
Signature ok
subject=/C=us/ST=washington/L=seattle/O=example corporation/OU=marketing/
CN=www.example.com/emailAddress=someone@example.com
Getting Private key
```

Behalten Sie den privaten Schlüssel und das öffentliche Zertifikat für eine spätere Nutzung. Sie können die Signierungsanforderung verwerfen. Speichern Sie den privaten Schlüssel immer an einem sicheren Ort und vermeiden Sie es, ihn zu Ihrem Quellcode hinzuzufügen.

Zum Verwenden des Zertifikats mit der Windows Server-Plattform müssen Sie es in ein PFX-Format konvertieren. Verwenden Sie den folgenden Befehl zum Erstellen eines PFX-Zertifikats aus dem privaten Schlüssel und den öffentlichen Zertifikatsdateien:

```
$ openssl pkcs12 -export -out example.com.pfx -inkey privatekey.pem -in public.crt
```

```
Enter Export Password: password
Verifying - Enter Export Password: password
```

Wenn Sie ein Zertifikat haben, können Sie es für die Verwendung mit einem Load Balancer in IAM hochladen oder die Instances in Ihrer Umgebung konfigurieren, um HTTPS zu beenden.

Hochladen eines Zertifikats in IAM

Um Ihr Zertifikat mit dem Load Balancer Ihrer Elastic Beanstalk Beanstalk-Umgebung zu verwenden, laden Sie das Zertifikat und den privaten Schlüssel in AWS Identity and Access Management (IAM) hoch. Sie können ein in IAM gespeichertes Zertifikat mit Elastic Load Balancing Balancing-Load Balancern und CloudFront Amazon-Distributionen verwenden.

> **Note**
>
> AWS Certificate Manager (ACM) ist das bevorzugte Tool für die Bereitstellung, Verwaltung und Bereitstellung Ihrer Serverzertifikate. Weitere Informationen zum Anfordern eines ACM-Zertifikats finden Sie unter Zertifikat anfordern im AWS Certificate Manager - Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in ACM finden Sie unter Importieren von Zertifikaten im AWS Certificate Manager -Benutzerhandbuch. Verwenden Sie IAM nur dann, um ein Zertifikat hochzuladen, wenn ACM in Ihrer Region nicht verfügbar ist. AWS

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um Ihr Zertifikat hochzuladen. Mit dem folgenden Befehl wird ein selbstsigniertes Zertifikat *https-cert.crt* mit einem privaten Schlüssel namens hochgeladen: *private-key.pem*

```
$ aws iam upload-server-certificate --server-certificate-name elastic-beanstalk-x509 --
certificate-body file://https-cert.crt --private-key file://private-key.pem
{
    "ServerCertificateMetadata": {
        "ServerCertificateId": "AS5YBEIONO2Q7CAIHKNGC",
        "ServerCertificateName": "elastic-beanstalk-x509",
        "Expiration": "2017-01-31T23:06:22Z",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:server-certificate/elastic-beanstalk-x509",
        "UploadDate": "2016-02-01T23:10:34.167Z"
    }
}
```

Das `file://` Präfix weist AWS CLI an, den Inhalt einer Datei in das aktuelle Verzeichnis zu laden. *`elastic-beanstalk-x509`*gibt den Namen an, unter dem das Zertifikat in IAM aufgerufen werden soll.

Wenn Sie ein Zertifikat von einer Zertifizierungsstelle erworben und eine Zertifikatskettendatei erhalten haben, laden Sie diese auch mit der `--certificate-chain`-Option hoch:

```
$ aws iam upload-server-certificate --server-certificate-name elastic-beanstalk-x509 --
certificate-chain file://certificate-chain.pem --certificate-body file://https-cert.crt
 --private-key file://private-key.pem
```

Notieren Sie den Amazon-Ressourcennamen (ARN) für Ihr Zertifikat. Sie verwenden diesen, wenn Sie die Konfigurationseinstellungen Ihres Load Balancer für die Verwendung von HTTPS aktualisieren.

> **Note**
>
> Ein Zertifikat in IAM hochgeladenes Zertifikat bleibt auch dann gespeichert, wenn es in keinem Load Balancer der Umgebung mehr verwendet wird. Es enthält sensible Daten. Wenn Sie das Zertifikat für keine Umgebung mehr benötigen, stellen Sie sicher, dass Sie es löschen. Weitere Informationen zum Löschen eines Zertifikats von IAM finden Sie unter https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_server-certs.html#delete-server-certificate.

Weitere Informationen über Serverzertifikate in IAM finden Sie unter Arbeiten mit Serverzertifikaten im IAM-Benutzerhandbuch.

Sicheres Speichern von privaten Schlüsseln in Amazon S3

Der private Schlüssel, den Sie zum Signieren des öffentlichen Zertifikats verwenden, ist geheim und sollte nicht an den Quellcode übergeben werden. Sie können vermeiden, dass private Schlüssel in Konfigurationsdateien gespeichert werden, indem Sie sie in Amazon S3 hochladen und Elastic Beanstalk so konfigurieren, dass die Datei während der Anwendungsbereitstellung aus Amazon S3 heruntergeladen wird.

Das folgende Beispiel veranschaulicht, wie Sie in den Abschnitten Resources und files einer Konfigurationsdatei eine Datei mit einem privaten Schlüssel aus einem Amazon S3-Bucket herunterladen.

Example.ebextensions/privatekey.config

```
Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          buckets: ["elasticbeanstalk-us-west-2-123456789012"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
              DefaultValue: "aws-elasticbeanstalk-ec2-role"
files:
  # Private key
  "/etc/pki/tls/certs/server.key":
    mode: "000400"
    owner: root
    group: root
    authentication: "S3Auth"
    source: https://elasticbeanstalk-us-west-2-123456789012.s3.us-west-2.amazonaws.com/
server.key
```

Ersetzen Sie den Bucket-Namen und die URL im Beispiel durch Ihre eigenen Angaben. Mit dem ersten Eintrag der Datei wird eine Authentifizierungsmethode mit dem Namen `S3Auth` zu den Metadaten der Auto Scaling-Gruppe für diese Umgebung hinzugefügt. Falls Sie ein benutzerdefiniertes Instance-Profil für die Umgebung konfiguriert haben, wird dieses verwendet. Andernfalls wird der Standardwert von `aws-elasticbeanstalk-ec2-role` übernommen. Das Instance-Standardprofil besitzt die Berechtigung zum Lesen der Speicher-Buckets von Elastic Beanstalk. Bei Verwendung eines anderen Buckets fügen Sie Berechtigungen zum Instance-Profil hinzu.

Vom zweiten Eintrag wird die Authentifizierungsmethode `S3Auth` verwendet, um den privaten Schlüssel über die angegebene URL herunterzuladen und unter `/etc/pki/tls/certs/server.key` zu speichern. Der Proxy-Server kann dann den privaten Schlüssel an diesem Speicherort auslesen und HTTPS-Verbindungen zur Instance beenden.

Das Instance-Profil, das den EC2 Instances Ihrer Umgebung zugewiesen ist, muss berechtigt sein, das Schlüsselobjekt aus dem angegebenen Bucket zu lesen. Stellen Sie sicher, dass das Instance-

Profil über die Berechtigung zum Lesen des Objekts in IAM verfügt und dass die Berechtigungen für den Bucket und das Objekt das Instance-Profil nicht ausschließen.

So zeigen Sie die Berechtigungen eines Buckets an

1. Öffnen Sie die Amazon S3-Managementkonsole.
2. Wählen Sie einen Bucket aus.
3. Wählen Sie Properties (Eigenschaften) und dann Permissions (Berechtigungen) aus.
4. Vergewissern Sie sich, dass Ihr Konto für diesen Bucket berechtigt ist und Leseberechtigung hat.
5. Wenn eine Bucket-Richtlinie angefügt ist, wählen Sie Bucket policy (Bucket-Richtlinie) aus, um die dem Bucket zugewiesenen Berechtigungen anzuzeigen.

Konfiguration der HTTPS-Terminierung am Load Balancer

Um Ihre AWS Elastic Beanstalk Umgebung für die Verwendung von HTTPS zu aktualisieren, müssen Sie einen HTTPS-Listener für den Load Balancer in Ihrer Umgebung konfigurieren. Zwei Arten von Load Balancer unterstützen einen HTTPS-Listener: Classic Load Balancer und Application Load Balancer.

Sie können die Elastic Beanstalk-Konsole oder eine Konfigurationsdatei verwenden, um einen sicheren Listener zu konfigurieren und das Zertifikat zuzuweisen.

> **Note**
>
> Umgebungen mit einer Instance enthalten keinen Load Balancer und unterstützen die HTTPS-Terminierung am Load Balancer nicht.

Sicheren Listener mit der Elastic Beanstalk-Konsole konfigurieren

So weisen Sie ein Zertifikat zum Load Balancer Ihrer Umgebung hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

4. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.

> **Note**
>
> Ist in der Konfigurationskategorie Load balancer die Schaltfläche Edit (Bearbeiten) nicht vorhanden, verfügt Ihre Umgebung über keinen Load Balancer.

5. Auf der Seite Modify load balancer (Load Balancer ändern) variiert die Vorgehensweise je nach dem Typ des Load Balancer, der Ihrer Umgebung zugeordnet ist.

   - Classic Load Balancer

     a. Wählen Sie Add listener (Listener hinzufügen) aus.

     b. Konfigurieren Sie im Dialogfeld Classic Load Balancer listener die folgenden Einstellungen:

        - Geben Sie für Listener port (Listener-Port) den Port für eingehenden Datenverkehr ein, in der Regel `443`.
        - Wählen Sie für Listener protocol (Listener-Protokoll) die Option HTTPS aus.
        - Geben Sie für Instance port (Instance-Port) den Wert `80` ein.
        - Wählen Sie für Instance protocol (Instance-Protokoll) die Option HTTP aus.
        - Wählen Sie für SSL certificate (SSL-Zertifikat) Ihr Zertifikat aus.

     c. Wählen Sie Hinzufügen aus.

   - Application Load Balancer

     a. Wählen Sie Add listener (Listener hinzufügen) aus.

     b. Konfigurieren Sie im Dialogfeld Application Load Balancer listener die folgenden Einstellungen:

        - Geben Sie für Port den Port für eingehenden Datenverkehr ein, in der Regel `443`.
        - Wählen Sie für Protocol (Protokoll) die Option HTTPS aus.
        - Wählen Sie für SSL certificate (SSL-Zertifikat) Ihr Zertifikat aus.

     c. Wählen Sie Hinzufügen aus.

> **Note**
>
> Wenn das Dropdown-Menü für Classic Load Balancer und Application Load Balancer keine Zertifikate anzeigt, sollten Sie ein Zertifikat für Ihren benutzerdefinierten Domänennamen in AWS Certificate Manager (ACM) erstellen oder hochladen (bevorzugt). Alternativ können Sie mit der AWS CLI ein Zertifikat in IAM hochladen.

- Network Load Balancer

  a. Wählen Sie Add listener (Listener hinzufügen) aus.

  b. Geben Sie im Dialogfeld Network Load Balancer-Listener für Port den Port für eingehenden Datenverkehr ein, normalerweise `443`.

  c. Wählen Sie Hinzufügen aus.

6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Konfigurieren eines sicheren Listeners mit einer Konfigurationsdatei

Sie können einen sicheren Listener auf Ihrem Load Balancer mit einer der folgenden Konfigurationsdateien konfigurieren.

Example.ebextensions/securelistener-clb.config

Verwenden Sie dieses Beispiel, wenn Ihre Umgebung über einen Classic Load Balancer verfügt. Im Beispiel werden Optionen im `aws:elb:listener`-Namespace verwendet, um einen HTTPS-Listener auf Port 443 mit dem angegebenen Zertifikat zu konfigurieren und den verschlüsselten Datenverkehr an die Instances in Ihrer Umgebung auf Port 80 weiterzuleiten.

```
option_settings:
  aws:elb:listener:443:
    SSLCertificateId: arn:aws:acm:us-east-2:1234567890123:certificate/
###################################
    ListenerProtocol: HTTPS
    InstancePort: 80
```

Ersetzen Sie den markierten Text mit dem ARN Ihres Zertifikats. Das Zertifikat kann eines sein, das Sie erstellt oder in AWS Certificate Manager (ACM) hochgeladen haben (bevorzugt), oder eines, das Sie mit dem in IAM hochgeladen haben. AWS CLI

Weitere Informationen zu Classic Load Balancer-Konfigurationsoptionen finden Sie unter Classic Load Balancer-Konfiguration in Namespaces.

Example.ebextensions/securelistener-alb.config

Verwenden Sie dieses Beispiel, wenn Ihre Umgebung über einen Application Load Balancer verfügt. Im Beispiel werden Optionen im `aws:elbv2:listener`-Namespace zum Konfigurieren eines HTTPS-Listener auf Port 443 mit dem angegebenen Zertifikat verwendet. Der Listener leitet den Datenverkehr an den Standardprozess weiter.

```
option_settings:
  aws:elbv2:listener:443:
    ListenerEnabled: 'true'
    Protocol: HTTPS
    SSLCertificateArns: arn:aws:acm:us-east-2:1234567890123:certificate/
###################################
```

Example.ebextensions/securelistener-nlb.config

Verwenden Sie dieses Beispiel, wenn Ihre Umgebung über einen Network Load Balancer verfügt. Im Beispiel werden Optionen im `aws:elbv2:listener`-Namespace zum Konfigurieren eines Listener auf Port 443 verwendet. Der Listener leitet den Datenverkehr an den Standardprozess weiter.

```
option_settings:
  aws:elbv2:listener:443:
    ListenerEnabled: 'true'
```

Konfigurieren einer Sicherheitsgruppe

Wenn Sie Ihren Load Balancer so konfigurieren, dass er Datenverkehr an einen anderen Instance-Port als Port 80 weiterleitet, müssen Sie der Sicherheitsgruppe eine entsprechende Regel hinzufügen, die eingehenden Datenverkehr über den Instance-Port von Ihrem Load Balancer zulässt. Wenn Sie Ihre Umgebung in einer benutzerdefinierten VPC erstellen, fügt Elastic Beanstalk diese Regel für Sie hinzu.

Sie fügen diese Regel durch das Hinzufügen eines `Resources`-Schlüssels zu einer Konfigurationsdatei im `.ebextensions`-Verzeichnis für Ihre Anwendung hinzu.

Die folgende Beispiel-Konfigurationsdatei fügt eine Ingress-Regel zur `AWSEBSecurityGroup`-Sicherheitsgruppe hinzu. Dadurch ist Datenverkehr auf Port 1000 von der Sicherheitsgruppe des Load Balancer zulässig.

Example.ebextensions/sg-ingressfromlb.config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 1000
      FromPort: 1000
      SourceSecurityGroupId: {"Fn::GetAtt" : ["AWSEBLoadBalancerSecurityGroup",
 "GroupId"]}
```

Konfiguration der HTTPS-Terminierung auf der Instanz

Sie können Konfigurationsdateien zum Konfigurieren des Proxy-Servers verwenden, der Datenverkehr an Ihre Anwendung weiterleitet, um HTTPS-Verbindungen zu beenden. Dies ist nützlich, wenn Sie HTTPS mit einer Umgebung mit einer einzelnen Instance verwenden möchten, oder wenn Sie den Load Balancer so konfigurieren möchten, dass Datenverkehr durchgeht, ohne entschlüsselt werden.

Um HTTPS zu aktivieren, müssen Sie eingehenden Traffic auf Port 443 zu der EC2 Instance zulassen, auf der Ihre Elastic Beanstalk Beanstalk-Anwendung läuft. Verwenden Sie dazu den `Resources` Schlüssel in der Konfigurationsdatei, um den Eingangsregeln für die AWSEBSecurity Gruppensicherheitsgruppe eine Regel für Port 443 hinzuzufügen.

Der folgende Codeausschnitt fügt eine entsprechende Ingress-Regel zur `AWSEBSecurityGroup`-Sicherheitsgruppe hinzu, die Port 443 für allen Datenverkehr für eine Umgebung mit einer einzelnen Instance öffnet:

`.ebextensions/https-instance-securitygroup.config`

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
```

```
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Load-Balancing in einer Amazon Virtual Private Cloud-Standardumgebung (Amazon VPC) können Sie diese Richtlinie so ändern, dass nur Datenverkehr vom Load Balancer akzeptiert wird. Ein Beispiel finden Sie unter Konfiguration der end-to-end Verschlüsselung in einer Elastic Beanstalk Beanstalk-Umgebung mit Lastenausgleich.

Plattformen

- Beenden von HTTPS auf EC2-Instances, auf denen Docker ausgeführt wird
- Beenden von HTTPS auf EC2 Instances, auf denen Go ausgeführt wird
- Beenden von HTTPS auf EC2-Instances, auf denen Java SE ausgeführt wird
- Beenden von HTTPS auf EC2-Instances, auf denen Node.js ausgeführt wird
- Beenden von HTTPS auf EC2 Instanzen, auf denen PHP ausgeführt wird
- Beenden von HTTPS auf EC2-Instances, auf denen Python ausgeführt wird
- Beenden von HTTPS auf EC2-Instances, auf denen Ruby ausgeführt wird
- Beenden von HTTPS auf EC2-Instances, auf denen Tomcat ausgeführt wird
- HTTPS auf Amazon EC2-Instances mit .NET Core unter Linux terminieren
- Beenden von HTTPS auf EC2 Amazon-Instances, auf denen .NET ausgeführt wird

Beenden von HTTPS auf EC2-Instances, auf denen Docker ausgeführt wird

Verwenden Sie bei Docker-Containern eine Konfigurationsdatei für die HTTPS-Aktivierung.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `files` generiert folgende Dateien auf der Instance:

  `/etc/nginx/conf.d/https.conf`

  Konfiguriert den nginx-Server. Diese Datei wird mit dem Start des nginx-Service geladen.

`/etc/pki/tls/certs/server.crt`

Damit wird die Zertifikatdatei auf der Instance erstellt. *`certificate file contents`*Ersetzen Sie es durch den Inhalt Ihres Zertifikats.

> **Note**
>
> Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

Example.ebextensions/https-instance.config

```
files:
  /etc/nginx/conf.d/https.conf:
    mode: "000644"
    owner: root
    group: root
```

```
    content: |
      # HTTPS Server

      server {
        listen 443;
        server_name localhost;

        ssl on;
        ssl_certificate /etc/pki/tls/certs/server.crt;
        ssl_certificate_key /etc/pki/tls/certs/server.key;

        ssl_session_timeout 5m;

        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_prefer_server_ciphers on;

        location / {
          proxy_pass http://docker;
          proxy_http_version 1.1;

          proxy_set_header Connection "";
          proxy_set_header Host $host;
          proxy_set_header X-Real-IP $remote_addr;
          proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
          proxy_set_header X-Forwarded-Proto https;
        }
      }

  /etc/pki/tls/certs/server.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
```

```
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Beenden von HTTPS auf EC2 Instances, auf denen Go ausgeführt wird

Verwenden Sie bei Go-Containertypen eine Konfigurationsdatei für die HTTPS-Aktivierung sowie eine nginx-Konfigurationsdatei, um den nginx-Server für die Verwendung von HTTPS zu konfigurieren.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Platzhalter für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `Resources` aktiviert Port 443 auf der Sicherheitsgruppe, die von der Umgebungs-Instance verwendet wird.
- Der Schlüssel `files` generiert folgende Dateien auf der Instance:

  `/etc/pki/tls/certs/server.crt`

  Damit wird die Zertifikatdatei auf der Instance erstellt. *`certificate file contents`*Ersetzen Sie es durch den Inhalt Ihres Zertifikats.

  > **Note**
  >
  > Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

  Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

  ```
      -----BEGIN CERTIFICATE-----
  certificate file contents
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  first intermediate certificate
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  second intermediate certificate
  -----END CERTIFICATE-----
  ```

  `/etc/pki/tls/certs/server.key`

  Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

Example.ebextensions/https-instance.config

```
files:
  /etc/pki/tls/certs/server.crt:
```

```
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

Binden Sie folgendes Snippet in eine `.conf`-Datei ein und platzieren Sie diese im Verzeichnis `.ebextensions/nginx/conf.d/` des Quell-Bundles (z. B. `.ebextensions/nginx/conf.d/https.conf`). *`app_port`*Ersetzen Sie es durch die Portnummer, auf der Ihre Anwendung lauscht. In diesem Beispiel wird der nginx-Server zur Überwachung von Port 443 mit SSL konfiguriert. Weitere Informationen zu diesen Konfigurationsdateien für die Go-Plattform finden Sie unter Konfigurieren des Proxy-Server.

Example. ebextensions/nginx/conf.d/https.conf

```
# HTTPS server

server {
    listen       443;
    server_name  localhost;

    ssl                  on;
    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;

    ssl_session_timeout  5m;
```

```
    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers   on;

    location / {
        proxy_pass  http://localhost:app_port;
        proxy_set_header   Connection "";
        proxy_http_version 1.1;
        proxy_set_header        Host            $host;
        proxy_set_header        X-Real-IP       $remote_addr;
        proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header        X-Forwarded-Proto https;
    }
}
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Beenden von HTTPS auf EC2-Instances, auf denen Java SE ausgeführt wird

Verwenden Sie bei Java SE-Containertypen eine .ebextensions-Konfigurationsdatei für die HTTPS-Aktivierung sowie eine nginx-Konfigurationsdatei, um den nginx-Server für die Verwendung von HTTPS zu konfigurieren.

Alle AL2023 AL2 /-Plattformen unterstützen eine einheitliche Proxykonfigurationsfunktion. Weitere Informationen zur Konfiguration des Proxyservers auf Ihren Plattformversionen, auf denen AL2023/ ausgeführt wirdAL2, finden Sie unterReverse-Proxy-Konfiguration.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Platzhalter für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `files` generiert folgende Dateien auf der Instance:

  `/etc/pki/tls/certs/server.crt`

  Damit wird die Zertifikatdatei auf der Instance erstellt. *`certificate file contents`*Ersetzen Sie es durch den Inhalt Ihres Zertifikats.

  > **Note**
  >
  > Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

  Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

Example.ebextensions/https-instance.config

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

Binden Sie folgendes Snippet in eine `.conf`-Datei ein und platzieren Sie diese im Verzeichnis `.ebextensions/nginx/conf.d/` des Quell-Bundles (z. B. `.ebextensions/nginx/conf.d/https.conf`). *`app_port`*Ersetzen Sie es durch die Portnummer, auf der Ihre Anwendung lauscht. In diesem Beispiel wird der nginx-Server zur Überwachung von Port 443 mit SSL konfiguriert. Weitere Informationen zu diesen Konfigurationsdateien für die Java SE-Plattform finden Sie unter Konfigurieren des Proxy-Server.

Example. ebextensions/nginx/conf.d/https.conf

```
# HTTPS server

server {
    listen       443;
    server_name  localhost;

    ssl                  on;
    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;

    ssl_session_timeout  5m;

    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers   on;

    location / {
        proxy_pass  http://localhost:app_port;
        proxy_set_header   Connection "";
        proxy_http_version 1.1;
        proxy_set_header        Host            $host;
        proxy_set_header        X-Real-IP       $remote_addr;
        proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header        X-Forwarded-Proto https;
    }
}
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
```

```
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Beenden von HTTPS auf EC2-Instances, auf denen Node.js ausgeführt wird

Die folgende Beispielkonfigurationsdatei erweitert die Standard-Nginx-Konfiguration so, dass sie Port 443 abhört und SSL/TLS Verbindungen mit einem öffentlichen Zertifikat und einem privaten Schlüssel beendet.

Wenn Sie Ihre Umgebung für erweiterte Zustandsberichte konfiguriert haben, müssen Sie nginx für die Generierung von Zugriffsprotokollen konfigurieren. Hierzu muss die Auskommentierung des Zeilenblocks unter dem Kommentar mit dem Text `# For enhanced health...` durch Entfernen der führenden #-Zeichen erfolgen.

Example.ebextensions/https-instance.config

```
files:
  /etc/nginx/conf.d/https.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      # HTTPS server

      server {
          listen       443;
          server_name  localhost;

          ssl                  on;
          ssl_certificate      /etc/pki/tls/certs/server.crt;
          ssl_certificate_key  /etc/pki/tls/certs/server.key;

          ssl_session_timeout  5m;

          ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
          ssl_prefer_server_ciphers   on;

          # For enhanced health reporting support, uncomment this block:
```

```
        #if ($time_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})T(\d{2})") {
        #    set $year $1;
        #    set $month $2;
        #    set $day $3;
        #    set $hour $4;
        #}
        #access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
healthd;
        #access_log  /var/log/nginx/access.log  main;

        location / {
            proxy_pass  http://nodejs;
            proxy_set_header   Connection "";
            proxy_http_version 1.1;
            proxy_set_header        Host            $host;
            proxy_set_header        X-Real-IP       $remote_addr;
            proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header        X-Forwarded-Proto https;
        }
    }

 /etc/pki/tls/certs/server.crt:
   mode: "000400"
   owner: root
   group: root
   content: |
     -----BEGIN CERTIFICATE-----
     certificate file contents
     -----END CERTIFICATE-----

 /etc/pki/tls/certs/server.key:
   mode: "000400"
   owner: root
   group: root
   content: |
     -----BEGIN RSA PRIVATE KEY-----
     private key contents # See note below.
     -----END RSA PRIVATE KEY-----
```

Der Schlüssel `files` generiert folgende Dateien auf der Instance:

`/etc/nginx/conf.d/https.conf`

Konfiguriert den nginx-Server. Diese Datei wird mit dem Start des nginx-Service geladen.

`/etc/pki/tls/certs/server.crt`

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie es *`certificate file contents`* durch den Inhalt Ihres Zertifikats.

> **Note**
>
> Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Beenden von HTTPS auf EC2 Instanzen, auf denen PHP ausgeführt wird

Verwenden Sie bei PHP-Containertypen eine Konfigurationsdatei, um den Apache HTTP-Server für die Verwendung von HTTPS zu aktivieren.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis `.ebextensions`.

Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `packages` nutzt "yum" für die Installation von `mod24_ssl`.
- Der Schlüssel `files` generiert folgende Dateien auf der Instance:

  `/etc/httpd/conf.d/ssl.conf`

  Konfiguriert den Apache-Server. Diese Datei wird mit dem Start des Apache-Service geladen.

  `/etc/pki/tls/certs/server.crt`

  Damit wird die Zertifikatdatei auf der Instance erstellt. *`certificate file contents`*Ersetzen Sie es durch den Inhalt Ihres Zertifikats.

> **Note**
>
> Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

Example.ebextensions/https-instance.config

```
packages:
  yum:
    mod24_ssl : []

files:
  /etc/httpd/conf.d/ssl.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
```

```
      LoadModule ssl_module modules/mod_ssl.so
      Listen 443
      <VirtualHost *:443>
        <Proxy *>
          Order deny,allow
          Allow from all
        </Proxy>

        SSLEngine             on
        SSLCertificateFile    "/etc/pki/tls/certs/server.crt"
        SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
        SSLCipherSuite        EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
        SSLProtocol           All -SSLv2 -SSLv3
        SSLHonorCipherOrder   On
        SSLSessionTickets     Off

        Header always set Strict-Transport-Security "max-age=63072000;
includeSubdomains; preload"
        Header always set X-Frame-Options DENY
        Header always set X-Content-Type-Options nosniff

        ProxyPass / http://localhost:80/ retry=0
        ProxyPassReverse / http://localhost:80/
        ProxyPreserveHost on
        RequestHeader set X-Forwarded-Proto "https" early

      </VirtualHost>

  /etc/pki/tls/certs/server.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
```

```
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Beenden von HTTPS auf EC2-Instances, auf denen Python ausgeführt wird

Für Python-Containertypen mit Apache HTTP-Server mit dem Web Server Gateway Interface (WSGI) verwenden Sie eine Konfigurationsdatei, um den Apache HTTP-Server für die Verwendung von HTTPS zu aktivieren.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu. Folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im `.ebextensions`-Verzeichnis des Quell-Bundles. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `packages` nutzt "yum" für die Installation von `mod_ssl`.
- Der Schlüssel `files` generiert folgende Dateien auf der Instance:

  `/etc/httpd/conf.d/ssl.conf`

  Konfiguriert den Apache-Server. Wenn Ihre Anwendung nicht den Namen `application.py` hat, ersetzen Sie den markierten Text im Wert für `WSGIScriptAlias` durch den lokalen Pfad zu Ihrer Anwendung. Beispiel: Eine Django-Anwendung kann sich zum Beispiel unter `django/wsgi.py` befinden. Der Speicherort sollte mit dem Wert der `WSGIPath`-Option, die Sie für Ihre Umgebung festlegen, übereinstimmen.

  Abhängig von den Anforderungen Ihrer Anwendung müssen Sie auch andere Verzeichnisse zum python-path-Parameter hinzufügen.

  `/etc/pki/tls/certs/server.crt`

  Damit wird die Zertifikatdatei auf der Instance erstellt. *`certificate file contents`*Ersetzen Sie es durch den Inhalt Ihres Zertifikats.

  > **Note**
  >
  > Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

  Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

- Mit dem Schlüssel `container_commands` wird der httpd-Service nach der abgeschlossenen Konfiguration beendet, sodass der Service die neue `https.conf`-Datei und das entsprechende Zertifikat verwendet.

> **Note**
>
> Das Beispiel funktioniert nur in Umgebungen, die die Python-Plattform verwenden.

Example.ebextensions/https-instance.config

```
packages:
  yum:
    mod_ssl: []

files:
  /etc/httpd/conf.d/ssl.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      LoadModule wsgi_module modules/mod_wsgi.so
      WSGIPythonHome /var/app/venv/staging-LQM1lest
      WSGISocketPrefix run/wsgi
      WSGIRestrictEmbedded On
      Listen 443
      <VirtualHost *:443>
        SSLEngine on
        SSLCertificateFile "/etc/pki/tls/certs/server.crt"
        SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"

        Alias /static/ /var/app/current/static/
        <Directory /var/app/current/static>
        Order allow,deny
        Allow from all
```

```
        </Directory>

        WSGIScriptAlias / /var/app/current/application.py

        <Directory /var/app/current>
        Require all granted
        </Directory>

        WSGIDaemonProcess wsgi-ssl processes=1 threads=15 display-name=%{GROUP} \
          python-path=/var/app/current \
          python-home=/var/app/venv/staging-LQM1lest \
          home=/var/app/current \
          user=webapp \
          group=webapp
        WSGIProcessGroup wsgi-ssl

      </VirtualHost>

  /etc/pki/tls/certs/server.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----

container_commands:
  01killhttpd:
    command: "killall httpd"
  02waitforhttpddeath:
    command: "sleep 3"
```

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

> **Hinweis für Amazon Linux 2023-Umgebungen**
>
> Auf Amazon Linux 2023 `mod_wsgi` muss es separat installiert werden, da es nicht in den Paket-Repositorys verfügbar ist. Installationsanweisungen finden Sie unter mod_wsgi auf PyPI.

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Beenden von HTTPS auf EC2-Instances, auf denen Ruby ausgeführt wird

Bei Ruby-Containertypen hängt die Art und Weise der HTTPS-Aktivierung vom eingesetzten Anwendungsserver ab.

Themen

- Konfigurieren von HTTPS für Ruby mit Puma
- Konfigurieren von HTTPS für Ruby mit Passenger

Konfigurieren von HTTPS für Ruby mit Puma

Verwenden Sie bei Ruby-Containertypen, die einen Puma-Anwendungsserver nutzen, eine Konfigurationsdatei für die HTTPS-Aktivierung.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `files` generiert folgende Dateien auf der Instance:

  `/etc/nginx/conf.d/https.conf`

  Konfiguriert den nginx-Server. Diese Datei wird mit dem Start des nginx-Service geladen.

  `/etc/pki/tls/certs/server.crt`

  Damit wird die Zertifikatdatei auf der Instance erstellt. *`certificate file contents`*Ersetzen Sie es durch den Inhalt Ihres Zertifikats.

  > **Note**
  >
  > Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

  Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

Example.ebextensions/https-instance.config

```
files:
  /etc/nginx/conf.d/https.conf:
    content: |
      # HTTPS server

      server {
          listen       443;
          server_name  localhost;

          ssl                  on;
          ssl_certificate      /etc/pki/tls/certs/server.crt;
          ssl_certificate_key  /etc/pki/tls/certs/server.key;

          ssl_session_timeout  5m;

          ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
          ssl_prefer_server_ciphers   on;

          location / {
              proxy_pass  http://my_app;
              proxy_set_header        Host            $host;
              proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
              proxy_set_header        X-Forwarded-Proto https;
          }

          location /assets {
            alias /var/app/current/public/assets;
            gzip_static on;
            gzip on;
            expires max;
```

```
            add_header Cache-Control public;
          }

          location /public {
            alias /var/app/current/public;
            gzip_static on;
            gzip on;
            expires max;
            add_header Cache-Control public;
          }
      }

  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
```

```
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Konfigurieren von HTTPS für Ruby mit Passenger

Verwenden Sie bei Ruby-Containertypen, die einen Passenger-Anwendungsserver nutzen, sowohl eine Konfigurationsdatei als auch eine JSON-Datei für die HTTPS-Aktivierung.

So konfigurieren Sie HTTPS für Ruby mit Passenger

1. Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

   - Der Schlüssel `files` generiert folgende Dateien auf der Instance:
     `/etc/pki/tls/certs/server.crt`

     Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie es durch den Inhalt Ihres *`certificate file contents`* Zertifikats.

     > **Note**
     >
     > Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

     Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
    -----BEGIN CERTIFICATE-----
  certificate file contents
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  first intermediate certificate
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  second intermediate certificate
  -----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

Example.Ebextensions-Snippet zum Konfigurieren von HTTPS für Ruby mit Passenger

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte

Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

2. Erstellen Sie eine Textdatei und fügen Sie die folgenden JSON-Daten ein. Speichern Sie die Datei im Stammverzeichnis des Quell-Bundles mit dem Namen `passenger-standalone.json`. Mit dieser JSON-Datei wird Passenger für die Verwendung von HTTPS konfiguriert.

   Important

   Die JSON-Datei darf keine Markierung der Bytereihenfolge enthalten. Sollte das der Fall sein, kann die Passenger-JSON-Bibliothek die Datei nicht korrekt lesen und der Passenger-Service wird nicht gestartet.

   Example passenger-standalone.json

```
{
  "ssl" : true,
  "ssl_port" : 443,
  "ssl_certificate" : "/etc/pki/tls/certs/server.crt",
  "ssl_certificate_key" : "/etc/pki/tls/certs/server.key"
}
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Beenden von HTTPS auf EC2-Instances, auf denen Tomcat ausgeführt wird

Verwenden Sie bei Tomcat-Containertypen eine Konfigurationsdatei, um den Apache HTTP-Server für die Verwendung von HTTPS zu aktivieren, wenn dieser als Reverse-Proxy-Server für Tomcat eingesetzt wird.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `files` generiert folgende Dateien auf der Instance:

  `/etc/pki/tls/certs/server.crt`

  Damit wird die Zertifikatdatei auf der Instance erstellt. *`certificate file contents`*Ersetzen Sie es durch den Inhalt Ihres Zertifikats.

  > **Note**
  >
  > Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

  `/etc/pki/tls/certs/server.key`

  Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

  `/opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh`

  Erstellt nach der Bereitstellung ein Hook-Skript für den Neustart des httpd-Service.

Example.ebextensions/https-instance.config

```
files:
  /etc/pki/tls/certs/server.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----

  /opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh:
    mode: "000755"
    owner: root
    group: root
    content: |
      #!/usr/bin/env bash
      sudo service httpd restart
```

Sie müssen auch den Proxy-Server Ihrer Umgebung für die Überwachung von Port 443 konfigurieren. Mit der folgenden Apache 2.4-Konfiguration wird ein Listener auf Port 443 hinzugefügt: Weitere Informationen hierzu finden Sie unter Konfigurieren des Proxy-Server.

Example. ebextensions/httpd/conf.d/ssl.conf

```
Listen 443
<VirtualHost *:443>
  ServerName server-name
  SSLEngine on
  SSLCertificateFile "/etc/pki/tls/certs/server.crt"
  SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"

  <Proxy *>
```

```
    Require all granted
  </Proxy>
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-ssl-error_log

</VirtualHost>
```

Möglicherweise erhalten Sie von Ihrem Zertifikatanbieter auch Zwischenzertifikate, die Sie für eine bessere Kompatibilität mit mobilen Clients installieren können. Konfigurieren Sie Apache mit einem Zwischenzertifizierungsstellen-Bundle, indem Sie Folgendes zur SSL-Konfigurationsdatei hinzufügen (siehe Erweiterung und Außerkraftsetzung der Apache-Standardkonfiguration — Amazon Linux AMI () AL1 für die Platzierung):

- Geben Sie in den `ssl.conf`-Dateiinhalten die Chain-Datei an:

```
SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
SSLCertificateChainFile "/etc/pki/tls/certs/gd_bundle.crt"
SSLCipherSuite        EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

- Fügen Sie dem Schlüssel `files` einen neuen Eintrag mit den Inhalten der Zwischenzertifikate hinzu:

```
files:
  /etc/pki/tls/certs/gd_bundle.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      First intermediate certificate
      -----END CERTIFICATE-----
      -----BEGIN CERTIFICATE-----
      Second intermediate certificate
      -----END CERTIFICATE-----
```

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

HTTPS auf Amazon EC2-Instances mit .NET Core unter Linux terminieren

Sie verwenden für .NT Core on Linux-Containertypen eine `.ebextensions`-Konfigurationsdatei, um HTTPS zu aktivieren, und eine nginx-Konfigurationsdatei, um den nginx-Server für die Verwendung von HTTPS zu konfigurieren.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Platzhalter für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `files` generiert folgende Dateien auf der Instance:

`/etc/pki/tls/certs/server.crt`

Damit wird die Zertifikatdatei auf der Instance erstellt. *`certificate file contents`*Ersetzen Sie es durch den Inhalt Ihres Zertifikats.

> **Note**
>
> Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Damit wird die private Schlüsseldatei auf der Instance erstellt. *`private key contents`*Ersetzen Sie es durch den Inhalt des privaten Schlüssels, der zur Erstellung der Zertifikatsanforderung oder des selbstsignierten Zertifikats verwendet wurde.

Example.ebextensions/https-instance.config

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
```

```
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter Sicheres Speichern von privaten Schlüsseln in Amazon S3.

Fügen Sie Folgendes in eine Datei mit der Erweiterung `.conf` im Verzeichnis `.platform/nginx/conf.d/` Ihres Quellpakets ein (z. B. `.platform/nginx/conf.d/https.conf`). *`app_port`*Ersetzen Sie es durch die Portnummer, auf der Ihre Anwendung lauscht. In diesem Beispiel wird der nginx-Server zur Überwachung von Port 443 mit SSL konfiguriert. Weitere Informationen zu diesen Konfigurationsdateien auf der .NET Core on Linux-Plattform finden Sie unter the section called “Proxy-Server”.

Example. platform/nginx/conf.d/https.conf

```
# HTTPS server

server {
    listen       443 ssl;
    server_name  localhost;

    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;

    ssl_session_timeout  5m;

    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers   on;

    location / {
```

```
        proxy_pass  http://localhost:app_port;
        proxy_set_header   Connection "";
        proxy_http_version 1.1;
        proxy_set_header        Host            $host;
        proxy_set_header        X-Real-IP       $remote_addr;
        proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header        X-Forwarded-Proto https;
    }
}
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Beenden von HTTPS auf EC2 Amazon-Instances, auf denen .NET ausgeführt wird

Die folgende Konfigurationsdatei erstellt und führt ein PowerShell Windows-Skript aus, das die folgenden Aufgaben ausführt:

- Überprüft, ob ein vorhandenes HTTPS-Zertifikat an Port 443 gebunden ist.
- Ruft das PFX-Zertifikat aus einem Amazon S3 S3-Bucket ab.

> **Note**
>
> Fügen Sie eine `AmazonS3ReadOnlyAccess` Richtlinie für den `aws-elasticbeanstalk-ec2-role` Zugriff auf das SSL-Zertifikat im Amazon S3 S3-Bucket hinzu.

- Ruft das Passwort von ab AWS Secrets Manager.

> **Note**
>
> Fügen Sie eine Anweisung hinzu`aws-elasticbeanstalk-ec2-role`, die die `secretsmanager:GetSecretValue` Aktion für das Geheimnis ermöglicht, das das Zertifikatspasswort enthält

- Installiert das Zertifikat.
- Bindet das Zertifikat an Port 443.

> **Note**
>
> Um den HTTP-Endpunkt (Port 80) zu entfernen, fügen Sie den `Remove-WebBinding`-Befehl unter dem Abschnitt Remove the HTTP binding des Beispiels hinzu.

Example.ebeextensions/ .config https-instance-dotnet

```
files:
  "C:\\certs\\install-cert.ps1":
    content: |
      import-module webadministration
      ## Settings - replace the following values with your own
      $bucket = "amzn-s3-demo-bucket"  ## S3 bucket name
      $certkey = "example.com.pfx"    ## S3 object key for your PFX certificate
      $secretname = "example_secret"  ## AWS Secrets Manager name for a secret that
 contains the certificate's password
      ##

      # Set variables
      $certfile = "C:\cert.pfx"
      $pwd = Get-SECSecretValue -SecretId $secretname | select -expand SecretString
```

```
      # Clean up existing binding
      if ( Get-WebBinding "Default Web Site" -Port 443 ) {
        Echo "Removing WebBinding"
        Remove-WebBinding -Name "Default Web Site" -BindingInformation *:443:
      }
      if ( Get-Item -path IIS:\SslBindings\0.0.0.0!443 ) {
        Echo "Deregistering WebBinding from IIS"
        Remove-Item -path IIS:\SslBindings\0.0.0.0!443
      }

      # Download certificate from S3
      Read-S3Object -BucketName $bucket -Key $certkey -File $certfile

      # Install certificate
      Echo "Installing cert..."
      $securepwd = ConvertTo-SecureString -String $pwd -Force -AsPlainText
      $cert = Import-PfxCertificate -FilePath $certfile cert:\localMachine\my -Password
 $securepwd

      # Create site binding
      Echo "Creating and registering WebBinding"
      New-WebBinding -Name "Default Web Site" -IP "*" -Port 443 -Protocol https
      New-Item -path IIS:\SslBindings\0.0.0.0!443 -value $cert -Force

      ## Remove the HTTP binding
      ## (optional) Uncomment the following line to unbind port 80
      # Remove-WebBinding -Name "Default Web Site" -BindingInformation *:80:
      ##

      # Update firewall
      netsh advfirewall firewall add rule name="Open port 443" protocol=TCP
 localport=443 action=allow dir=OUT

commands:
  00_install_ssl:
    command: powershell -NoProfile -ExecutionPolicy Bypass -file C:\\certs\\install-
cert.ps1
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Die folgende Konfigurationsdatei ruft mithilfe einer CloudFormation Funktion die ID der Sicherheitsgruppe ab und fügt ihr eine Regel hinzu.

Example.ebeextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In einer Umgebung mit Lastenausgleich konfigurieren Sie den Load Balancer so, dass er entweder sicheren Datenverkehr unangetastet weiterleitet oder zur Verschlüsselung entschlüsselt und erneut verschlüsselt. end-to-end

Konfiguration der end-to-end Verschlüsselung in einer Elastic Beanstalk Beanstalk-Umgebung mit Lastenausgleich

Das Beenden sicherer Verbindungen am Load Balancer und das Verwenden von HTTP am Backend kann ausreichend für Ihre Anwendung sein. Der Netzwerkverkehr zwischen AWS -Ressourcen kann nicht von Instances überwacht werden, die nicht Teil der Verbindung sind, selbst wenn sie im selben Konto ausgeführt werden.

Wenn Sie jedoch eine Anwendung entwickeln, die strenge externe Vorschriften einhalten muss, müssen Sie ggf. alle Netzwerkverbindungen sichern. Sie können die Elastic Beanstalk-Konsole oder Konfigurationsdateien verwenden, damit der Load Balancer der Elastic Beanstalk-Umgebung eine sichere Verbindung zu Backend-Instances herstellt, um diese Anforderungen zu erfüllen. Die folgenden Schritte konzentrieren sich auf Konfigurationsdateien.

Zuerst fügen Sie einen sicheren Listener zu Ihrem Load Balancer hinzu, falls Sie dies noch nicht getan haben.

Sie müssen auch die Instances in Ihrer Umgebung so konfigurieren, dass sie auf dem sicheren Port überwachen und HTTPS-Verbindungen beenden. Die Konfiguration variiert je nach Plattform. Detaillierte Anweisungen finden Sie unter Konfiguration der HTTPS-Terminierung auf der Instanz. Sie können problemlos ein selbstsigniertes Zertifikat für die Instances verwenden. EC2

Anschließend konfigurieren Sie den Listener für die Weiterleitung von Datenverkehr mithilfe von HTTPS auf den sicheren Port, der von Ihrer Anwendung verwendet wird. Verwenden Sie eine der folgenden Konfigurationsdateien, je nach Typ des Load Balancer, den Ihre Umgebung verwendet.

`.ebextensions/https-reencrypt-clb.config`

Verwenden Sie diese Konfigurationsdatei mit einem Classic Load Balancer. Neben der Konfiguration des Load Balancer ändert die Konfigurationsdatei auch die standardmäßige Zustandsprüfung, sodass Port 443 und HTTPS verwendet werden, um sicherzustellen, dass der Load Balancer in der Lage ist, eine sichere Verbindung herzustellen.

```
option_settings:
  aws:elb:listener:443:
    InstancePort: 443
    InstanceProtocol: HTTPS
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: HTTPS:443/
```

`.ebextensions/https-reencrypt-alb.config`

Verwenden Sie diese Konfigurationsdatei mit einem Application Load Balancer.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
    Protocol: HTTPS
```

`.ebextensions/https-reencrypt-nlb.config`

Verwenden Sie diese Konfigurationsdatei mit einem Network Load Balancer.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
```

Die Option `DefaultProcess` hat diesen Namen wegen Application Load Balancers, die Nicht-Standard-Listener auf demselben Port für Datenverkehr auf bestimmten Pfaden haben können (siehe Application Load Balancer für weitere Informationen). Bei einem Network Load Balancer gibt die Option den einzigen Zielprozess für diesen Listener an.

In diesem Beispiel haben wir den Prozess `https` genannt, da er sicheren (HTTPS) Datenverkehr überwacht. Der Listener sendet den Datenverkehr über das TCP-Protokoll an den Prozess auf dem festgelegten Port, da ein Network Load Balancer nur mit TCP funktioniert. Das ist in Ordnung, da der Netzwerkdatenverkehr für HTTP und HTTPS zusätzlich zu TCP implementiert ist.

> **Note**
>
> Die EB-CLI und Elastic-Beanstalk-Konsole wenden die für die vorhergehenden Optionen empfohlenen Werte an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter Empfohlene Werte.

In der nächsten Aufgabe müssen Sie die Sicherheitsgruppe des Load Balancer so ändern, dass Datenverkehr zulässig ist. Abhängig von der Amazon Virtual Private Cloud (Amazon VPC) (die Standard-VPC oder eine benutzerdefinierte VPC), in der Sie Ihre Umgebung starten, variiert die Sicherheitsgruppe des Load Balancers. In einer Standard-VPC bietet Elastic Load Balancing eine Standard-Sicherheitsgruppe, die alle Load Balancer verwenden können. In einer von Ihnen erstellten Amazon VPC erstellt Elastic Beanstalk eine Sicherheitsgruppe, die der Load Balancer verwenden kann.

Zur Unterstützung beider Szenarien können Sie eine Sicherheitsgruppe erstellen und Elastic Beanstalk anweisen, diese zu verwenden. Die folgende Konfigurationsdatei erstellt eine Sicherheitsgruppe und fügt Sie an den Load Balancer an.

`.ebextensions/https-lbsecuritygroup.config`

```
option_settings:
  # Use the custom security group for the load balancer
  aws:elb:loadbalancer:
    SecurityGroups: ``{ "Ref" : "loadbalancersg" }``
    ManagedSecurityGroup: ``{ "Ref" : "loadbalancersg" }``

Resources:
  loadbalancersg:
```

```
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: load balancer security group
      VpcId: vpc-########
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 443
          ToPort: 443
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
      SecurityGroupEgress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
```

Ersetzen Sie den markierten Text mit der Standard- oder benutzerdefinierten VPC-ID. Das vorherige Beispiel enthält ein- und ausgehenden Datenverkehr über Port 80, um HTTP-Verbindungen zuzulassen. Sie können diese Eigenschaften entfernen, wenn Sie nur sichere Verbindungen zulassen möchten.

Fügen Sie nun Eingangs- und Ausgangsregeln hinzu, mit denen die Kommunikation über Port 443 zwischen der Sicherheitsgruppe des Load Balancer und der Sicherheitsgruppe der Instances möglich ist.

`.ebextensions/https-backendsecurity.config`

```
Resources:
  # Add 443-inbound to instance security group (AWSEBSecurityGroup)
  httpsFromLoadBalancerSG:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      SourceSecurityGroupId: {"Fn::GetAtt" : ["loadbalancersg", "GroupId"]}
  # Add 443-outbound to load balancer security group (loadbalancersg)
  httpsToBackendInstances:
    Type: AWS::EC2::SecurityGroupEgress
```

```
    Properties:
      GroupId: {"Fn::GetAtt" : ["loadbalancersg", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      DestinationSecurityGroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
```

Wenn Sie dies separat von der Erstellung der Sicherheitsgruppe durchführen, können Sie die Quell- und Zielsicherheitsgruppen beschränken, ohne einen Zirkelbezug zu erstellen.

Wenn alle vorherigen Aufgaben durchgeführt wurden, stellt der Load Balancer eine sichere Verbindung zu Ihren Backend-Instances über HTTPS her. Der Load Balancer berücksichtigt nicht, ob das Zertifikat der Instance selbstsigniert oder von einer vertrauenswürdigen Zertifizierungsstelle ausgestellt wurde, und akzeptiert alle Zertifikate, die im vorgelegt werden.

Sie können dieses Verhalten ändern, indem Sie Richtlinien zum Load Balancer hinzufügen, die besagen, dass nur einem bestimmten Zertifikat vertraut werden darf. Mit der folgenden Konfigurationsdatei werden zwei Richtlinien erstellt. Eine Richtlinie gibt ein öffentliches Zertifikat an und die andere weist den Load Balancer an, nur diesem Zertifikat für Verbindungen zum Instance-Port 443 zu vertrauen

`.ebextensions/https-backendauth.config`

```
option_settings:
  # Backend Encryption Policy
  aws:elb:policies:backendencryption:
    PublicKeyPolicyNames: backendkey
    InstancePorts:  443
  # Public Key Policy
  aws:elb:policies:backendkey:
    PublicKey: |
      -----BEGIN CERTIFICATE-----
      ################################################################
      ################################################################
      ################################################################
      ################################################################
      ################################################
      -----END CERTIFICATE-----
```

Ersetzen Sie den hervorgehobenen Text durch den Inhalt des öffentlichen Zertifikats Ihrer EC2 Instanz.

Konfigurieren des Load Balancers der Umgebung für TCP-Pass-Through

Wenn Sie nicht möchten, dass der Load Balancer in Ihrer AWS Elastic Beanstalk Umgebung den HTTPS-Verkehr entschlüsselt, können Sie den sicheren Listener so konfigurieren, dass er Anfragen unverändert an Back-End-Instances weiterleitet.

> **Important**
>
> Die Konfiguration des Load Balancers so, dass er HTTPS-Verkehr weiterleitet, ohne ihn zu entschlüsseln, stellt einen Nachteil dar. Der Load Balancer kann die verschlüsselten Anfragen nicht sehen und kann daher das Routing nicht optimieren oder Antwortmetriken melden.

Konfigurieren Sie zunächst die EC2 Instanzen Ihrer Umgebung so, dass sie HTTPS beenden. Testen Sie die Konfiguration in einer Umgebung mit einer einzelnen Instance und stellen Sie die ordnungsgemäße Funktionsweise sicher, bevor Sie den Load Balancer hinzufügen.

Fügen Sie eine Konfigurationsdatei zum Projekt hinzu, um einen Listener auf Port 443 zu konfigurieren, der TCP-Pakete unverändert an Port 443 auf den Backend-Instances weitergibt:

`.ebextensions/https-lb-passthrough.config`

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: TCP
    InstancePort: 443
    InstanceProtocol: TCP
```

In einer Standard-Amazon Virtual Private Cloud (Amazon VPC) muss der Sicherheitsgruppe der Instances zudem eine Regel hinzugefügt werden, damit eingehender Verkehr vom Load Balancer über Port 443 zulässig ist:

`.ebextensions/https-instance-securitygroup.config`

```
Resources:
  443inboundfromloadbalancer:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
```

```
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      SourceSecurityGroupName: { "Fn::GetAtt": ["AWSEBLoadBalancer",
"SourceSecurityGroup.GroupName"] }
```

In einer benutzerdefinierten VPC aktualisiert Elastic Beanstalk die Konfiguration der Sicherheitsgruppe für Sie.

Konfigurieren der Umleitung von HTTP zu HTTPS

In diesem Thema wird beschrieben, wie Sie den HTTP-Verkehr zu Ihrer Anwendung handhaben, wenn Endbenutzer ihn dennoch initiieren. Konfigurieren Sie hierzu die Umleitung von HTTP zu HTTPS (manchmal auch bezeichnet als Erzwingen von HTTPS).

Um die Umleitung zu konfigurieren, müssen Sie zunächst Ihre Umgebung für die Verarbeitung von HTTPS-Datenverkehr konfigurieren. Dann leiten Sie HTTP-Datenverkehr zu HTTPS um. Diese beiden Schritte werden in den folgenden Unterabschnitten erläutert.

Umgebung für die Verarbeitung von HTTPS-Datenverkehr konfigurieren

Führen Sie je nach Load-Balancing-Konfiguration in Ihrer Umgebung einen der folgenden Schritte aus:

- Umgebung mit Lastausgleich – Konfiguriert den Load Balancer zum Beenden von HTTPS.
- Umgebung mit einer einzelnen Instance – Konfiguriert Ihre Anwendung zum Beenden von HTTPS-Verbindungen auf der Instance. Diese Konfiguration ist abhängig von Ihrer Umgebungsplattform.

HTTP-Datenverkehr an HTTPS umleiten

Um den HTTP-Verkehr für Ihre Anwendung auf HTTPS umzuleiten, können Sie entweder die Webserver auf den Instances Ihrer Umgebung oder den Application Load Balancer der Umgebung konfigurieren.

Konfigurieren Sie die Instanz-Webserver

Diese Methode funktioniert in jeder Webserver-Umgebung. Konfigurieren Sie Webserver auf Ihren EC2 Amazon-Instances so, dass sie auf HTTP-Verkehr mit einem Antwortstatus für die HTTP-Umleitung reagieren.

Diese Konfiguration ist abhängig von Ihrer Umgebungsplattform. Suchen Sie den Ordner für Ihre Plattform in der `https-redirect`Sammlung auf GitHub und verwenden Sie die Beispielkonfigurationsdatei in diesem Ordner.

Wenn Ihre Umgebung Elastic Load Balancing-Zustandsprüfungen verwendet, erwartet der Load Balancer, dass die HTTP-Zustandsprüfungs-Nachrichten von einer funktionierenden Instance mit einer 200 OK-HTTP-Antwort beantwortet werden. Aus diesem Grund sollte Ihr Webserver diese Nachrichten nicht an HTTPS umleiten. In den Beispiel-Konfigurationsdateien in `https-redirect` wird diese Anforderung korrekt erfüllt.

Konfigurieren Sie den Load Balancer

Diese Methode funktioniert, wenn Sie über eine Umgebung mit Lastenausgleich verfügen, die einen Application Load Balancer verwendet. Ein Application Load Balancer kann Umleitungsantworten senden, wenn HTTP-Verkehr eingeht. In diesem Fall müssen Sie die Umleitung für die Instances Ihrer Umgebung nicht konfigurieren.

Wir haben zwei Beispielkonfigurationsdateien GitHub , die zeigen, wie ein Application Load Balancer für die Umleitung konfiguriert wird.

- Die Konfigurationsdatei `alb-http-to-https-redirection-full.config` erstellt einen HTTPS-Listener an Port 443 und ändert den Standard-Port 80-Listener, um eingehenden HTTP-Datenverkehr an HTTPS umzuleiten.
- In der `alb-http-to-https-redirection.config`Konfigurationsdatei wird erwartet, dass der 443-Listener definiert ist. Um es zu definieren, können Sie standardmäßige Elastic Beanstalk Beanstalk-Konfigurationsnamespaces oder die Elastic Beanstalk Beanstalk-Konsole verwenden. Anschließend sorgt sie dafür, dass der Port 80-Listener für die Umleitung geändert wird.

Elastic Beanstalk-Plattformen

AWS Elastic Beanstalk bietet eine Vielzahl von Plattformen, auf denen Sie Ihre Anwendungen erstellen können. Sie gestalten Ihre Webanwendung auf einer dieser Plattformen und Elastic Beanstalk stellt Ihren Code auf der von Ihnen gewählte Plattformversion bereit, um eine aktive Anwendungsumgebung zu erstellen.

Elastic Beanstalk bietet Plattformen für verschiedene Programmiersprachen , Anwendungsserver sowie Docker-Container. Einige Plattformen verfügen über mehrere gleichzeitig unterstützte Versionen.

Themen

- Elastic Beanstalk-Plattformen – Glossar
- Modell der übergreifenden Verantwortlichkeit für die Wartung der Elastic Beanstalk-Plattform
- Support-Richtlinie für die Elastic Beanstalk-Plattform
- Zeitplan für die Veröffentlichung der Elastic Beanstalk-Plattform
- Von Elastic Beanstalk unterstützte Plattformen
- Elastic Beanstalk-Linux-Plattformen
- Erweitern von Elastic Beanstalk-Linux-Plattformen

Elastic Beanstalk-Plattformen – Glossar

Im Folgenden finden Sie wichtige Begriffe zu AWS Elastic Beanstalk Plattformen und ihrem Lebenszyklus.

Laufzeit

> Das Programmieren von sprachspezifischer Laufzeitsoftware (Framework, Bibliotheken, Interpreter, vm usw.), die zur Ausführung Ihres Anwendungscodes erforderlich sind.

Elastic Beanstalk-Komponenten

> Softwarekomponenten, die Elastic Beanstalk zu einer Plattform hinzufügt, um die Elastic Beanstalk-Funktionalität zu ermöglichen. Beispiel: Der Agent erweiterter Zustandsberichte wird für die Erfassung und Berichterstellung von Zustandsinformationen benötigt.

Plattform

Eine Plattform ist eine Kombination aus Betriebssystem- (OS), Laufzeit-, Webserver-, Anwendungsserver- und Elastic Beanstalk-Komponenten. Plattformen stellen Komponenten bereit, die für die Ausführung Ihrer Anwendung verfügbar sind.

Plattformversion

Eine Kombination spezifischer Versionen von Betriebssystem- (OS), Laufzeit-, Webserver-, Anwendungsserver- und Elastic Beanstalk-Komponenten. Sie erstellen eine Elastic Beanstalk-Umgebung basierend auf einer Plattformversion und stellen Ihre Anwendung für diese bereit.

Eine Plattformversion verfügt über eine semantische Versionsnummer im Format X.Y.Z. Dabei ist X die Hauptversion, Y die Nebenversion und Z die Patch-Version.

Eine Plattformversion kann sich in einem der folgenden Zustände befinden:

- Empfohlen — Die neueste Plattformversion in einer unterstützten Plattformbranche. Diese Version enthält die meisten up-to-date Komponenten und wird für den Einsatz in Produktionsumgebungen empfohlen. Wenn Elastic Beanstalk eine neue Plattformversion veröffentlicht, ersetzt die neue Version die vorherige Version und wird zur empfohlenen Plattformversion für den entsprechenden Plattformzweig.
- Nicht empfohlen — Jede Plattformversion, die nicht die neueste Version in ihrem Plattformzweig ist. Diese Versionen können zwar weiterhin funktionsfähig sein, wir empfehlen jedoch dringend, auf die neueste Plattformversion zu aktualisieren. Sie können verwaltete Plattform-Updates verwenden, um up-to-date automatisch zu bleiben.

Sie können überprüfen, ob eine Plattformversion empfohlen wird, indem Sie den AWS-CLI-Befehl verwenden describe-platform-version und das `PlatformLifecycleState` Feld überprüfen.

Plattformzweig

Eine Reihe von Plattformversionen, denen bestimmte Versionen (in der Regel Hauptversionen) einiger Komponenten gemeinsam sind, z. B. das Betriebssystem, die Laufzeit oder Elastic Beanstalk-Komponenten. Zum Beispiel: Python 3.13 läuft auf 64-Bit-Amazon Linux 2023; IIS 10.0 läuft auf 64-Bit-Windows Server 2025. Plattformzweige erhalten Updates in Form neuer Plattformversionen. Jede nachfolgende Plattformversion in einem Zweig ist ein Update der vorherigen.

Die empfohlene Version in jeder unterstützten Plattformbranche steht Ihnen für die Erstellung einer Umgebung bedingungslos zur Verfügung. Frühere Plattformversionen sind weiterhin für Konten mit aktiven oder beendeten Umgebungen zugänglich, die sie zu dem Zeitpunkt nutzten,

als sie durch eine neue Version ersetzt wurden. Früheren Plattformversionen fehlen die meisten up-to-date Komponenten und ihre Verwendung wird nicht empfohlen.

> **Note**
>
> Wenn Sie Zugriff auf frühere Plattformversionen benötigen, die über die oben beschriebene Standardverfügbarkeit hinausgehen, können Sie sich an das AWS Support Center wenden, um Unterstützung zu erhalten.

Ein Plattformzweig kann sich in einem der folgenden Zustände befinden:

- Unterstützt – Ein Plattformzweig auf dem aktuellen Stand. Er besteht vollständig aus unterstützten Komponenten. Die unterstützten Komponenten haben das von ihren Lieferanten angegebene Ende der Lebensdauer (EOL) noch nicht erreicht. Er erhält kontinuierlich Plattformaktualisierungen und wird für die Verwendung in Produktionsumgebungen empfohlen. Die Liste der unterstützten Plattformzweige finden Sie unter Von Elastic Beanstalk unterstützte Plattformen im Handbuch AWS Elastic Beanstalk -Platfformen.
- Beta – Ein Plattformzweig in der Vorversion vor der Veröffentlichung. Ein solcher Plattformzweig ist experimenteller Natur. Er erhält möglicherweise für eine Weile kontinuierlich Plattformaktualisierungen, wird jedoch nicht langfristig unterstützt. Ein Plattformzweig in der Betaphase wird nicht für die Verwendung in Produktionsumgebungen empfohlen. Sie sollten einen solchen Plattformzweig ausschließlich zu Evaluierungszwecken verwenden. Die Liste der Plattformzweige in der Betaphase finden Sie unter Elastic Beanstalk-Plattformversionen in der öffentlichen Beta im Handbuch AWS Elastic Beanstalk -Plattformen.
- Veraltet — Ein Plattformzweig, in dem eine oder mehrere Komponenten (z. B. die Laufzeit oder das Betriebssystem) sich dem Ende des Lebenszyklus (EOL) nähern oder das EOL erreicht haben, wie von ihren Lieferanten angegeben. Während ein veralteter Plattformzweig bis zu seinem Auslaufdatum weiterhin neue Plattformversionen erhält, erhalten Komponenten, die EOL erreicht haben, keine Updates. Wenn eine Runtime-Version beispielsweise EOL erreicht, wird der Plattformzweig als veraltet markiert, erhält aber weiterhin Betriebssystemupdates, bis der Plattformzweig eingestellt wird. Der Plattformzweig erhält keine weiteren Updates für die EOL-Laufzeitversion. Die Verwendung eines veralteten Plattformzweigs wird nicht empfohlen.
- Im Ruhestand — Ein Plattformzweig, der keine Updates mehr erhält. Stillgelegte Plattformbranches sind nicht verfügbar, um neue Elastic Beanstalk Beanstalk-Umgebungen mit der Elastic Beanstalk Beanstalk-Konsole zu erstellen. Wenn Ihre Umgebung einen veralteten Plattform-Branch verwendet, müssen Sie auf einen unterstützten Plattform-

Branch aktualisieren, um weiterhin Updates zu erhalten. Die Verwendung eines stillgelegten Plattformzweigs wird nicht empfohlen. Weitere Informationen zu stillgelegten Plattformzweigen finden Sie unterthe section called “Plattform-Unterstützungsrichtlinie”. Eine Liste der Plattformfilialen, deren Stilllegung geplant ist, finden Sie unter Liste stillgelegter Plattformfilialen. Weitere Informationen zu stillgelegten Plattformfilialen finden Sie unter Historie stillgelegter Plattformzweige.

Wenn Ihre Umgebung einen veralteten oder außer Betrieb genommenen Plattformzweig verwendet, sollten Sie diesen auf eine Plattformversion in einem unterstützten Plattformzweig aktualisieren. Details hierzu finden Sie unter the section called “Plattformaktualisierungen”.

Sie können den Status eines Plattformzweigs überprüfen, indem Sie den AWS-CLI-Befehl verwenden describe-platform-version und das `PlatformBranchLifecycleState` Feld überprüfen.

Plattformaktualisierung

Eine Veröffentlichung einer neuen Plattformversion, die Updates für einige Komponenten der Plattform enthält — Betriebssystem-, Runtime-, Webserver-, Anwendungsserver- und Elastic Beanstalk Beanstalk-Komponenten. Wenn Elastic Beanstalk eine neue Plattformversion veröffentlicht, ersetzt die neue Version die vorherige Version und wird zur empfohlenen Plattformversion für den entsprechenden Plattformzweig. Plattform-Updates folgen der semantischen Versionstaxonomie und können drei Stufen haben:

- Hauptaktualisierung – Eine Aktualisierung, die Änderungen enthält, die mit vorhandenen Plattformversionen nicht kompatibel sind. Möglicherweise müssen Sie Ihre Anwendung ändern, damit sie auf einer neuen Hauptversion ordnungsgemäß ausgeführt wird. Eine Hauptaktualisierung besitzt eine neue Hauptversionsnummer für die Plattform.
- Kleineres Update — Ein Update, das Änderungen enthält, die in den meisten Fällen abwärtskompatibel mit bestehenden Plattformversionen sind. Abhängig von Ihrer Anwendung müssen Sie Ihre Anwendung möglicherweise ändern, damit sie auf einer neuen Nebenversion ordnungsgemäß ausgeführt wird. Eine Nebenaktualisierung besitzt eine neue Nebenversionsnummer für die Plattform.
- Patch-Aktualisierung – Eine Aktualisierung, die aus Wartungsveröffentlichungen (Fehlerbehebungen, Sicherheitsupdates und Leistungsverbesserungen) besteht, die mit einer vorhandenen Plattformversion abwärtskompatibel sind. Eine Patch-Aktualisierung besitzt eine neue Patch-Plattform-Versionsnummer.

Verwaltete Aktualisierungen

Eine Elastic Beanstalk-Funktion, die automatisch Patch- und Nebenaktualisierungen auf Betriebssystem- (OS), Laufzeit-, Webserver-, Anwendungsserver- und Elastic Beanstalk-Komponenten für eine von Elastic Beanstalk unterstützte Plattformversion anwendet. Eine verwaltete Aktualisierung wendet eine neuere Plattformversion im selben Plattformzweig auf Ihre Umgebung an. Sie können verwaltete Aktualisierungen so konfigurieren, dass nur Patch-Aktualisierungen oder Neben- und Patch-Aktualisierungen angewendet werden. Sie können verwaltete Aktualisierungen auch vollständig deaktivieren.

Weitere Informationen finden Sie unter Verwaltete Plattformaktualisierungen.

Modell der übergreifenden Verantwortlichkeit für die Wartung der Elastic Beanstalk-Plattform

AWS und unsere Kunden sind gemeinsam dafür verantwortlich, ein hohes Maß an Sicherheit und Konformität der Softwarekomponenten zu erreichen. Dieses Modell der übergreifenden Verantwortung bedeutet für Sie eine deutliche Verringerung des Betriebsaufwands.

Einzelheiten finden Sie im Modell der AWS geteilten Verantwortung.

AWS Elastic Beanstalk hilft Ihnen dabei, Ihren Teil des Modells der gemeinsamen Verantwortung umzusetzen, indem es eine Funktion für verwaltete Updates bereitstellt. Diese Funktion wendet automatisch Patches und kleinere Aktualisierungen für eine von Elastic Beanstalk unterstützte Plattformversion an. Wenn eine verwaltete Aktualisierung fehlschlägt, benachrichtigt Elastic Beanstalk Sie über den Ausfall, um sicherzustellen, dass Sie informiert sind und unverzüglich reagieren können.

Weitere Informationen finden Sie unter Verwaltete Plattformaktualisierungen.

Darüber hinaus leistet Elastic Beanstalk Folgendes:

- Veröffentlicht seine Richtlinie zur Plattformunterstützung und seinen Aussonderungsplan für die nächsten 12 Monate.
- Veröffentlichung von Patch-, Minor- und Major-Updates für Betriebssystem (OS), Laufzeitumgebung, Anwendungsserver und Webserverkomponenten in der Regel innerhalb von 30 Tagen nach ihrer Verfügbarkeit. Elastic Beanstalk ist für die Erstellung von Updates für Elastic Beanstalk-Komponenten verantwortlich, die auf den unterstützten Plattformversionen vorhanden

sind. Alle anderen Aktualisierungen stammen direkt von ihren Lieferanten (Eigentümern oder Community).

Wir kündigen alle Updates unserer unterstützten Plattformen in unseren Versionshinweisen im Leitfaden zu AWS Elastic Beanstalk Versionshinweisen an. Im Plattformleitfaden AWS Elastic Beanstalk finden Sie auch eine Liste aller unterstützten Plattformen und ihrer Komponenten sowie einen Plattformverlauf. Weitere Informationen finden Sie unter Unterstützte Plattformen und Komponentenverlauf.

Sie sind für Folgendes verantwortlich:

- Aktualisieren Sie alle Komponenten, die Sie kontrollieren (im Modell mit AWS geteilter Verantwortung als Kunde gekennzeichnet). Dies umfasst die Gewährleistung der Sicherheit Ihrer Anwendung, Ihrer Daten und aller Komponenten, die Ihre Anwendung benötigt und die Sie heruntergeladen haben.
- Sicherstellen, dass Ihre Elastic Beanstalk-Umgebungen auf einer unterstützte Plattformversion ausgeführt werden, und Migrieren einer jeden Umgebung, die auf einer nicht mehr unterstützten Plattformversion ausgeführt wird, auf eine unterstützte Version.
- Wenn Sie ein benutzerdefiniertes Amazon Machine Image (AMI) für Ihre Elastic Beanstalk-Umgebung verwenden, patchen, warten und testen Sie Ihr benutzerdefiniertes AMI, damit es aktuell und kompatibel mit einer unterstützten Elastic Beanstalk-Plattformversion bleibt. Weitere Informationen zur Verwaltung von Umgebungen mit einem benutzerdefinierten AMI finden Sie unterVerwenden eines benutzerdefinierten Amazon Machine Images (AMI) in Ihrer Elastic Beanstalk Beanstalk-Umgebung.
- Lösen aller Probleme bei fehlgeschlagenen verwalteten Aktualisierungsversuchen und Neuversuchen der Aktualisierung.
- Selbstständiges Patchen von Betriebssystem-, Laufzeit-, Anwendungsserver- und Webserver-Komponenten, wenn Sie sich gegen Elastic Beanstalk-verwaltete Aktualisierungen entschieden haben. Sie können hierzu Plattformaktualisierungen manuell anwenden oder die Komponenten auf allen relevanten Umgebungsressourcen direkt patchen.
- Verwalte die Sicherheit und Compliance aller AWS Services, die du außerhalb von Elastic Beanstalk nutzt, gemäß dem AWS Shared Responsibility Model.

Support-Richtlinie für die Elastic Beanstalk-Plattform

Elastic Beanstalk unterstützt Plattform-Branches, die laufende Minor- und Patch-Updates von ihren Lieferanten (Eigentümer oder Community) erhalten. Eine vollständige Definition zugehöriger Begriffe finden Sie unter Elastic Beanstalk-Plattformen – Glossar.

Ausgemusterte Plattformvarianten

Wenn eine Komponente eines unterstützten Plattformzweigs von seinem Lieferanten als End of Life (EOL) markiert wird, markiert Elastic Beanstalk den Plattformzweig als ausgemustert. Zu den Komponenten eines Plattformzweigs gehören: Betriebssystem (OS), Laufzeit-Sprachversion, Anwendungsserver oder Webserver.

Sobald ein Plattformzweig als veraltet markiert ist, gelten die folgenden Richtlinien:

- Elastic Beanstalk stellt die Bereitstellung von Wartungsupdates, einschließlich Sicherheitsupdates, ein.
- Elastic Beanstalk bietet keinen technischen Support mehr für stillgelegte Plattformfilialen.
- Elastic Beanstalk stellt den Plattformzweig neuen Elastic Beanstalk-Kunden nicht mehr für Implementierungen in neuen Umgebungen zur Verfügung. Für Bestandskunden mit aktiven Umgebungen, die auf eingestellten Plattformzweigen laufen, gibt es eine 90-tägige Nachfrist ab dem angekündigten Einstellungsdatum.

> **Note**
>
> Ein stillgelegter Plattformzweig wird in der Elastic Beanstalk Beanstalk-Konsole nicht verfügbar sein. Es wird jedoch über die EB-CLI und die AWS CLI EB-API für Kunden verfügbar sein, die über bestehende Umgebungen verfügen, die auf der stillgelegten Plattformbranche basieren. Bestandskunden können auch die Konsolen Clone Environment und Rebuild Environment verwenden.

Eine Liste der Plattformbranchen, deren Stilllegung geplant ist, finden Sie Zeitplan für die Ausmusterung von Plattformvarianten im folgenden Thema im Elastic Beanstalk-Plattformplan.

Weitere Informationen darüber, was zu erwarten ist, wenn der Plattform-Branch Ihrer Umgebung eingestellt wird, finden Sie unter. Häufig gestellte Fragen zur Einstellung von Plattformen

Über die 90-Tage-Nachfrist hinaus

Unsere Richtlinie für stillgelegte Plattform-Zweigstellen verhindert weder den Zugriff auf Umgebungen noch das Löschen von Ressourcen. Bestandskunden, die eine Elastic Beanstalk Beanstalk-Umgebung auf einer stillgelegten Plattformfiliale betreiben, sollten sich jedoch der damit verbundenen Risiken bewusst sein. Solche Umgebungen können in eine unvorhersehbare Situation geraten, da Elastic Beanstalk nicht in der Lage ist, Sicherheitsupdates, technischen Support oder Hotfixes für stillgelegte Plattformfilialen bereitzustellen, weil der Lieferant die Komponente als EOL markiert hat.

Beispielsweise kann eine nachteilige und kritische Sicherheitslücke in einer Umgebung auftreten, die in einem eingestellten Plattformzweig ausgeführt wird. Es könnte auch passieren, dass eine EB-API-Aktion nicht mehr für die Umgebung funktioniert, wenn sie irgendwann nicht mehr mit dem Elastic-Beanstalk-Service kompatibel ist. Die Gefahr dafür steigt, je länger ein Umfeld auf einem eingestellten Plattformzweig aktiv bleibt. Damit Sie weiterhin von wichtigen Sicherheits-, Leistungs- und Funktionserweiterungen profitieren können, die von Komponentenlieferanten in neueren Versionen angeboten werden, empfehlen wir Ihnen dringend, alle Ihre Elastic Beanstalk-Umgebungen auf eine unterstützte Plattformversion zu aktualisieren.

Falls bei der Ausführung Ihrer Anwendung auf einer ausrangierten Plattform Probleme auftreten und Sie sie nicht auf eine unterstützte Plattform migrieren können, müssen Sie andere Alternativen in Betracht ziehen. Eine mögliche Umgehung ist das Einkapseln der Anwendung in ein Docker-Image zum Ausführen als Docker-Container. Dies würde es einem Kunden ermöglichen, jede unserer Docker-Lösungen wie unsere Elastic Beanstalk AL2 023/ AL2 Docker-Plattformen oder andere Docker-basierte Dienste wie Amazon ECS oder Amazon EKS zu nutzen. Zu den Alternativen, die nicht zu Docker gehören, gehört unser AWS CodeDeploy Service, der eine vollständige Anpassung der von Ihnen gewünschten Laufzeiten ermöglicht.

Zeitplan für die Veröffentlichung der Elastic Beanstalk-Plattform

Neben der monatlichen Veröffentlichung neuer Plattformversionen umfasst unsere Release-Wartung auch die folgenden Prozesse:

- Veröffentlichung neuer Plattformzweige — Diese führen in der Regel eine neue Hauptversion einer Laufzeitsprache, eines Betriebssystems oder eines Anwendungsservers ein.
- Außerbetriebnahme von Plattformzweigen — Wir müssen einen Plattformzweig stilllegen, wenn eine seiner Komponenten das Ende des Lebenszyklus (EOL) erreicht. Weitere Informationen zu unseren Richtlinien für stillgelegte Filialen finden Sie unter Support-Richtlinie für die Elastic Beanstalk-Plattform

Themen

- Ressourcen planen
- Kommende Veröffentlichungen in den Plattformbranchen
- Zeitplan für die Ausmusterung von Plattformvarianten
- Bisherige eingestellte Plattformzweige
- Verlauf des Servers und des Betriebssystems im Ruhestand

Ressourcen planen

Die folgenden Ressourcen können Ihnen bei der Planung von Wartung und Support für Ihre Anwendung helfen, die auf einer Elastic Beanstalk-Plattform ausgeführt wird.

- AWS Elastic Beanstalk Leitfaden für Plattformen — Dieser Leitfaden enthält eine detaillierte Komponentenliste für jeden unserer Plattformzweige. Es enthält auch einen Plattformverlauf nach Veröffentlichungsdatum mit denselben Details. Dieser Leitfaden kann Sie darüber informieren, wenn sich bestimmte Komponenten Ihres Plattformzweigs geändert haben. Wenn sich Ihre Anwendung anders verhält, können Sie auch im Plattformleitfaden auf das Datum des Vorfalls verweisen, um festzustellen, ob Plattformänderungen vorgenommen wurden, die sich auf Ihre Anwendung ausgewirkt haben könnten.
- AWS Elastic Beanstalk Versionshinweise — In unseren Versionshinweisen werden alle kleineren und größeren Plattformversionen angekündigt. Dazu gehören unsere monatlichen Plattform-Updates, Sicherheitsversionen, Hotfixes und Ankündigungen zur Einstellung unserer Website. Sie können unsere RSS-Feeds in der Dokumentation zu den Versionshinweisen abonnieren.

Kommende Veröffentlichungen in den Plattformbranchen

In der folgenden Tabelle sind die kommenden Elastic Beanstalk-Plattformzweige und ihr angestrebtes Veröffentlichungsdatum aufgeführt. Diese Daten sind vorläufig und können sich ändern.

Laufzeitversion//Plattformzweig	Betriebssystem	Angestrebtes Veröffentlichungsdatum
Python 3.15	Amazon Linux 2023	November 2026
Node.js 26	Amazon Linux 2023	November 2026

Laufzeitversion//Plattformzweig	Betriebssystem	Angestrebtes Veröffentlichungsdatum
.NET 11	Amazon Linux 2023	Dezember 2026
PHP 8.6	Amazon Linux 2023	Januar 2027
Ruby 4.1	Amazon Linux 2023	Februar 2027

Zeitplan für die Ausmusterung von Plattformvarianten

In der folgenden Tabelle sind die Filialen der Elastic Beanstalk-Plattform aufgeführt, deren Stilllegung geplant ist, da einige ihrer Komponenten ihr Lebensende (EOL) erreichen. Alle Zweigstellen AL2 auf Plattformen mit Sitz in Betrieb genommen werden, die nicht später als 30. Juni 2026 ablaufen, wenn Amazon Linux 2 das Ende der Laufzeit erreicht. Weitere Informationen zu Amazon Linux 2 finden Sie unter Amazon Linux FAQs 2.

Eine detailliertere Liste der auslaufenden Plattformzweige, einschließlich ihrer spezifischen Komponenten, finden Sie im Plattformhandbuch unter Plattformversionen zurückziehen.AWS Elastic Beanstalk

Zweig „Laufzeitversion“/ „Plattform“	Zieldatum für den Ruhestand
PHP 8.1 AL2023	31. März 2026
PHP 8.1 AL2	31. März 2026
Docker AL2	30. Juni 2026
ECS AL2	30. Juni 2026
Go 1 AL2	30. Juni 2026
Corretto 8 AL2	30. Juni 2026
Corretto 11 AL2	30. Juni 2026
Corretto 17 AL2	30. Juni 2026

Zweig „Laufzeitversion“/ „Plattform“	Zieldatum für den Ruhestand		
Corretto 8 with Tomcat 9 AL2	30. Juni 2026		
Corretto 11 with Tomcat 9 AL2	30. Juni 2026		
.NET Core AL2	30. Juni 2026		
Python 3.9 AL2023	31. Juli 2026		
Ruby 3.2 AL2023	31. Juli 2026		
Node.js 20 AL2023	31. Juli 2026		
IIS 10.0 on Windows Server 2016 (& Core)	30. September 2026		
PHP 8.2 AL2023	31. März 2027		
.NET 9 AL2023	31. März 2027		
.NET 8 AL2023	31. März 2027		

Bisherige eingestellte Plattformzweige

In den folgenden Tabellen sind Branches der Elastic Beanstalk-Plattform aufgeführt, die sich bereits im Status „Ausmusterung“ befinden. Eine detaillierte Historie dieser Plattform-Branches und ihrer Komponenten finden Sie im Leitfaden Plattformhistorie der AWS Elastic Beanstalk Plattformen.

Amazon Linux 2023 (AL2023)

Laufzeitversion// Plattformzweig	Datum der Einstellung		
.NET 6 AL2023	8. April 2025		
Node.js 18 AL2023	11. August 2025		

Amazon Linux (2AL2)

Laufzeitversion// Plattformzweig	Datum der Einstellung		
Corretto 11 with Tomcat 8.5 AL2	10. Oktober 2024		
Corretto 8 with Tomcat 8.5 AL2	10. Oktober 2024		
Corretto 11 with Tomcat 7 AL2	29. Juni 2022		
Corretto 8 with Tomcat 7 AL2	29. Juni 2022		
Node.js 18 AL2	11. August 2025		
Node.js 16 AL2	10. Oktober 2024		
Node.js 14 AL2	10. Oktober 2024		
Node.js 12 AL2	23. Dezember 2022		
Node.js 10 AL2	29. Juni 2022		
PHP 8.0 AL2	10. Oktober 2024		
PHP 7.4 AL2	9. Juni 2023		
PHP 7.3 AL2	29. Juni 2022		
PHP 7.2 AL2	29. Juni 2022		
Python 3.8 AL2	8. April 2025		
Python 3.7 AL2	10. Oktober 2024		
Ruby 3.0 AL2	10. Oktober 2024		
Ruby 2.7 AL2	10. Oktober 2024		

Laufzeitversion// Plattformzweig	Datum der Einstellung		
Ruby 2.6 AL2	23. Dezember 2022		
Ruby 2.5 AL2	29. Juni 2022		

Amazon Linux-AMI (AL1)

Laufzeitversion// Plattformzweig	Datum der Einstellung		
Single Container Docker	18. Juli 2022		
Multicontainer Docker	18. Juli 2022		
Preconfig ured Docker - GlassFish 5.0 with Java 8	18. Juli 2022		
Go 1	18. Juli 2022		
Java 8	18. Juli 2022		
Java 7	18. Juli 2022		
Java 8 with Tomcat 8.5	18. Juli 2022		
Java 7 with Tomcat 7	18. Juli 2022		
Node.js	18. Juli 2022		
PHP 7.2 - 7.3	18. Juli 2022		

Laufzeitversion// Plattformzweig	Datum der Einstellung		
Python 3.6	18. Juli 2022		
Ruby 2,4, 2.5, 2.6 with Passenger	18. Juli 2022		
Ruby 2.4, 2.5, 2.6 with Puma	18. Juli 2022		
Go 1.3–1.10	31. Oktober 2020		
Java 6	31. Oktober 2020		
Node.js 4.x-8.x	31. Oktober 2020		
PHP 5.4–5.6	31. Oktober 2020		
PHP 7.0–7.1	31. Oktober 2020		
Python 2.6, 2.7, 3.4	31. Oktober 2020		
Ruby 1.9.3	31. Oktober 2020		
Ruby 2.0–2.3	31. Oktober 2020		

> **Note**
>
> Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen finden Sie unter Häufig gestellte Fragen zur Einstellung von Plattformen.

Windows Server

Laufzeitversion// Plattformzweig	Datum der Einstellung		
IIS 8.5 läuft auf 64-Bit-Windows Server (& Core) 2012 R2	4. Dezember 2023		
IIS 8.5 unter 64 Bit Windows Server (& Core) 2012 R2 Version 0.1.0	29. Juni 2022		
IIS 8.5 unter 64 Bit Windows Server (& Core) 2012 R2 Version 1.2.0	29. Juni 2022		
IIS 10.0 unter 64 Bit Windows Server (& Core) 2016 Version 1.2.0	29. Juni 2022		
IIS 8 auf Windows Server 2012 R1 mit 64 Bit Plattform zweig	22. Juni 2022		
IIS 8 unter 64 Bit Windows Server 2012 R1 Version 0.1.0	22. Juni 2022		

Laufzeitversion// Plattformzweig	Datum der Einstellung		
IIS 8 unter 64 Bit Windows Server 2012 R1 Version 1.2.0	22. Juni 2022		

> **Note**
>
> Weitere Informationen zur Außerbetriebnahme der Windows-2012-R2-Plattformzweige finden Sie in den AWS Elastic Beanstalk -Versionshinweisen unter Plattformzweige von Windows Server 2012 R2 außer Betrieb genommen.

Verlauf des Servers und des Betriebssystems im Ruhestand

Die folgenden Tabellen enthalten eine Historie der Betriebssysteme, Anwendungsserver und Webserver, die von Elastic Beanstalk-Plattformen nicht mehr unterstützt werden. Alle Plattformzweige, die diese Komponenten nutzten, wurden inzwischen eingestellt. Die Daten spiegeln das Datum wider, an dem der letzte Zweig der Elastic Beanstalk-Plattform eingestellt wurde, der die Komponente enthielt.

Betriebssysteme

Betriebss ystemversion	Datum der Plattformausmusterung		
Windows Server 2012 R2 running IIS 8.5	4. Dezember 2023		
Windows Server Core 2012 R2 running IIS 8.5	4. Dezember 2023		

Betriebssystemversion	Datum der Plattformausmusterung		
Amazon Linux-AMI (AL1)	18. Juli 2022		
Windows Server 2012 R1	22. Juni 2022		
Windows Server 2008 R2	28. Oktober 2019		

Anwendungsserver

Anwendungsserver-Version	Datum der Plattformausmusterung		
Tomcat 7	29. Juni 2022 für Amazon Linux 2 (AL2) -Plattformen 18. Juli 2022 für Amazon Linux AMI (AL1) -Plattformen		
Tomcat 8	31. Oktober 2020		
Tomcat 6	31. Oktober 2020		

Web-Server

Webserver-Version	Datum der Plattformausmusterung		
Windows Server mit 64 Bit und IIS 8	22. Juni 2022		
Apache HTTP Server 2.2	31. Oktober 2020		

Webserver-Version	Datum der Plattformausmusterung		
Nginx 1.12.2	31. Oktober 2020		

Von Elastic Beanstalk unterstützte Plattformen

AWS Elastic Beanstalk bietet eine Vielzahl von Plattformen, auf denen Sie Ihre Anwendungen erstellen können. Sie gestalten Ihre Webanwendung auf einer dieser Plattformen und Elastic Beanstalk stellt Ihren Code auf der von Ihnen gewählte Plattformversion bereit, um eine aktive Anwendungsumgebung zu erstellen.

Elastic Beanstalk stellt die Ressourcen bereit, die für die Ausführung Ihrer Anwendung erforderlich sind, einschließlich einer oder mehrerer EC2 Amazon-Instances. Der Software-Stack, der auf den EC2 Amazon-Instances ausgeführt wird, hängt von der spezifischen Plattformversion ab, die Sie für Ihre Umgebung ausgewählt haben.

Der Name des Lösungsstapels für einen Plattformzweig

Sie können den Namen des Lösungsstapels für eine bestimmte Plattform-Branch-Version verwenden, um eine Umgebung mit der EB-CLI, der Elastic Beanstalk Beanstalk-API oder der AWS CLI zu starten. Im AWS Elastic Beanstalk Platforms Guide ist der Name des Solution Stacks unter der Plattform-Branch-Version sowohl in den Abschnitten Unterstützte Plattformen von Elastic Beanstalk als auch in der Plattformhistorie aufgeführt.

Um alle Namen des Lösungsstapels abzurufen, die Sie zum Erstellen einer Umgebung verwenden können, verwenden Sie die ListAvailableSolutionStacksAPI oder die `aws elasticbeanstalk list-available-solution-stacks`in der AWS CLI.

Sie können die Software, von der Ihre Anwendung auf Ihrer Plattform abhängig ist, anpassen und konfigurieren. Weitere Informationen finden Sie unter Anpassen von Software auf Linux-Servern und Anpassen von Software auf Windows-Servern. Ausführliche Versionshinweise sind für aktuelle Versionen unter den AWS Elastic Beanstalk -Versionshinweisen verfügbar.

Unterstützte Plattformen und Komponentenverlauf

Der AWS Elastic Beanstalk Platforms-Leitfaden listet alle aktuellen Plattform-Branch-Versionen im Abschnitt Unterstützte Plattformen von Elastic Beanstalk auf. Der Leitfaden für Plattformen listet auch

einen Plattformverlauf für jede Plattform auf, der auch eine Liste früherer Branch-Plattformversionen enthält. Wählen Sie einen der folgenden Links aus, um den Plattformverlauf für jede Plattform einzusehen.

- Docker
- Go
- Java SE
- Tomcat (auf Java SE)
- .NET Core auf Linux
- .NET mit Windows Server
- Node.js
- PHP
- Python
- Ruby

Elastic Beanstalk-Linux-Plattformen

Die Elastic Beanstalk Linux-Plattformen bieten eine umfangreiche, sofort einsatzbereite Funktionalität. Sie können die Plattformen auf verschiedene Arten erweitern, um Ihre Anwendung zu unterstützen. Details hierzu finden Sie unter the section called “Erweitern von Linux-Plattformen”.

Die meisten von Elastic Beanstalk unterstützten Plattformen basieren auf dem Linux-Betriebssystem. Insbesondere basieren diese Plattformen auf Amazon Linux, einer Linux-Distribution von AWS. Elastic Beanstalk Linux-Plattformen verwenden Amazon Elastic Compute Cloud (Amazon EC2) -Instances, und auf diesen Instances wird Amazon Linux ausgeführt.

Themen

- Unterstützte Amazon-Linux-Versionen
- Liste der Elastic Beanstalk-Linux-Plattformen
- Instance-Bereitstellungs-Workflow
- Workflow zur Instance-Bereitstellung für ECS auf Amazon Linux 2 und höher
- Plattform-Skripttools für Ihre Elastic Beanstalk Beanstalk-Umgebungen

Unterstützte Amazon-Linux-Versionen

AWS Elastic Beanstalk unterstützt Plattformen, die auf Amazon Linux 2 und Amazon Linux 2023 basieren.

Weitere Informationen zu Paketen für Amazon Linux 2 und Amazon Linux 2023 finden Sie unter:

- Amazon Linux 2 — Amazon Linux im EC2 Amazon-Benutzerhandbuch.
- Amazon Linux 2023 – Was ist Amazon Linux 2023? im Benutzerhandbuch für Amazon Linux 2023

Weitere Informationen zu unterstützten Plattformversionen finden Sie unter Von Elastic Beanstalk unterstützte Plattformen.

> **Note**
>
> Sie können Ihre Anwendung von einem Elastic Beanstalk AL1 - oder AL2 Plattformzweig auf den entsprechenden AL2 023-Plattformzweig migrieren. Weitere Informationen finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Amazon Linux 2023

AWS kündigte die allgemeine Verfügbarkeit von Amazon Linux 2023 im März 2023 an. Das Benutzerhandbuch für Amazon Linux 2023 fasst die wichtigsten Unterschiede zwischen Amazon Linux 2 und Amazon Linux 2023 zusammen. Weitere Informationen finden Sie unter Vergleich von Amazon Linux 2 und Amazon Linux 2023 im Benutzerhandbuch.

Es besteht ein hohes Maß an Kompatibilität zwischen den Plattformen Elastic Beanstalk Amazon Linux 2 und Amazon Linux 2023. Es gibt jedoch einige Unterschiede zu beachten:

- Instance Metadata Service Version 1 (IMDSv1) — Die IMDSv1 Option „Deaktivieren“ ist standardmäßig `true` auf AL2 023-Plattformen eingestellt. Die Standardeinstellung ist `false` auf Plattformen. AL2
- pkg-repo-Instanztool — Das Pkg-Repo Tool ist nicht für Umgebungen verfügbar, die auf AL2 023-Plattformen ausgeführt werden. Sie können Paket- und Betriebssystemupdates jedoch weiterhin manuell auf eine 023-Instanz anwenden. AL2 Weitere Informationen finden Sie unter Verwaltung von Paketen und Betriebssystem-Updates im Benutzerhandbuch für Amazon Linux 2023.

- HTTPd Apache-Konfiguration — Die `httpd.conf` Apache-Datei für AL2 023-Plattformen enthält einige Konfigurationseinstellungen, die sich von denen für folgende unterscheiden: AL2
  - Verweigern Sie standardmäßig den Zugriff auf das gesamte Dateisystem des Servers. Diese Einstellungen werden unter Serverdateien standardmäßig schützen auf der Seite mit den Sicherheitstipps der Apache-Website beschrieben.
  - Verweigert den Zugriff auf die Einrichtung von `.htaccess` in allen Verzeichnissen, mit Ausnahme der speziell aktivierten. Diese Einstellung wird unter Schützen von Systemeinstellungen auf der Seite mit den Sicherheitstipps der Apache-Website beschrieben. Auf der Seite Server-Tutorial für Apache HTTP: .htaccess-Dateien heißt es, dass diese Einstellung zur Verbesserung der Leistung beitragen kann.
  - Verweigern Sie den Zugriff auf Dateien mit einem Namensmuster `.ht*`. Diese Einstellung verhindert, dass Webclients alle `.htaccess`- und `.htpasswd`-Dateien anzeigen.

  Sie können jede der oben genannten Konfigurationseinstellungen für die Umgebung ändern. Weitere Informationen finden Sie unter Konfigurieren von Apache HTTPD.
- Unterstützung für mehrzeilige Umgebungsvariablen — AL2 023 Plattformen unterstützen mehrzeilige Werte für Umgebungsvariablen und geheime Schlüssel in Systemd-Dienstkonfigurationen. Amazon Linux 2-Plattformen unterstützen keine mehrzeiligen Umgebungsvariablenwerte. Diese Erweiterung ermöglicht es Ihnen, mehrzeilige Geheimnisse und Konfigurationswerte auf AL2 023-Plattformen zu verwenden. Weitere Hinweise zur Verwendung von Umgebungsvariablen und Geheimnissen finden Sie unter. the section called “Mehrzeilige Werte”
- CloudWatch benutzerdefinierte Protokollweiterleitung — Der veraltete CloudWatch Logs-Agent (`awslogs`Paket) ist auf AL2 023-Plattformen nicht verfügbar. Wenn Sie über benutzerdefinierte Protokollweiterleitungskonfigurationen verfügen, die den veralteten `awslogs` Agenten installieren und verwenden, müssen Sie Ihre Konfigurationsdateien aktualisieren, um den Unified CloudWatch Agent bei der Migration von Amazon Linux 2 auf 023 zu verwenden. AL2 Weitere Informationen finden Sie unter the section called “Benutzerdefiniertes Streaming von Protokolldateien”.

Plattformspezifische Unterschiede

Zusätzlich zu den grundlegenden Betriebssystemunterschieden gibt es plattformspezifische Unterschiede zwischen den Laufzeitplattformen Amazon Linux 2 und AL2 023:

- .NET-Plattformverzweigung — Die Branching-Strategie für die .NET-Plattform unterscheidet sich zwischen Amazon Linux 2 und AL2 023. Auf Amazon Linux 2 unterhält die .NET Core-Plattform

ein rotierendes Fenster mit den.NET-Hauptversionen innerhalb eines einzigen Plattformzweigs. Bei AL2 023 ist jeder Plattformzweig an eine bestimmte .NET-Hauptversion gebunden (z. B.. NET 9, .NET 10).

Wenn Sie Framework-abhängige Anwendungen (Anwendungen, die auf der installierten .NET-Laufzeit der Plattform basieren) bereitstellen, müssen Sie einen Plattformzweig auswählen, der der .NET-Zielversion Ihrer Anwendung entspricht. Wenn Sie eigenständige Anwendungen (Anwendungen, die ihre eigene .NET-Laufzeit bündeln) bereitstellen, können Sie unabhängig von der.NET-Version Ihrer Anwendung einen beliebigen AL2 023-.NET-Plattformzweig verwenden, da Ihre Anwendung nicht von der installierten Laufzeit der Plattform abhängig ist. Weitere Informationen finden Sie unter the section called “Anwendungsbündelung”.

- Versionsauswahl für Node.js — Die Plattform Node.js auf Amazon Linux 2 unterstützt die Angabe einer Node.js -Version in der `package.json` Datei Ihrer Anwendung. Die Node.js -Plattform auf AL2 Version 023 unterstützt diese Funktion nicht. Sie müssen die Standardversion von Node.js verwenden, die vom Plattformzweig bereitgestellt wird. Weitere Informationen zur Versionsverwaltung von Node.js finden Sie unterthe section called “Konfigurieren von Abhängigkeiten”.
- Ruby Puma-Serverversion — Die Ruby-Plattform auf Amazon Linux 2 ignoriert die in der `Gemfile.lock` Datei Ihrer Anwendung angegebene Puma-Version und verwendet die Puma-Standardversion der Plattform. Die Ruby-Plattform auf AL2 0.23 berücksichtigt die in angegebene Puma-Version, falls vorhanden. `Gemfile.lock` Wenn keine Version angegeben ist, installiert die Plattform die Puma-Standardversion der Plattform.
- Verfügbarkeit von PHP-Paketen — Einige Pakete, die auf Amazon Linux 2 PHP-Plattformen verfügbar sind, sind auf AL2 023 PHP-Plattformen nicht verfügbar:
  - MySQL-Client-Pakete — Die Client-Pakete `mysql` und die `mysql-devel` Befehlszeilen-Client-Pakete sind auf AL2 023 PHP-Plattformen nicht installiert. Wenn für Ihre Anwendung eine MySQL-Datenbankverbindung erforderlich ist, verwenden Sie PHP `mysqli` oder `pdo_mysql` Erweiterungen, die auf beiden Plattformen verfügbar sind.
  - Compass- und Ruby-Tools — Die `ruby-devel` `rubygems` Pakete für die Compass CSS-Framework-Unterstützung sind auf AL2 023 PHP-Plattformen nicht installiert. Compass ist veraltet. Erwägen Sie die Verwendung moderner CSS-Vorverarbeitungstools als Alternative.
- Tools zur Go-Versionskontrolle — Das Bazaar-Versionskontrollsystem (`bzr`) ist auf AL2 023 Go-Plattformen nicht verfügbar. Bazaar ist veraltet und nicht im 023-Paket-Repository enthalten. AL2 Verwenden Sie stattdessen Git, Mercurial oder Subversion für die Versionskontrolle, die alle auf AL2 023 Go-Plattformen verfügbar sind.

Liste der Elastic Beanstalk-Linux-Plattformen

In der folgenden Liste werden die Linux-Plattformen erwähnt, die Elastic Beanstalk für verschiedene Programmiersprachen sowie für Docker-Container unterstützt. Für alle davon bietet Elastic Beanstalk Plattformen an, die auf Amazon Linux 2 und Amazon Linux 2023 basieren. Um mehr über eine Plattform zu erfahren, wählen Sie den entsprechenden Link aus.

- Docker (und ECS Docker)
- Go
- Tomcat (auf Java SE)
- Java SE
- .NET Core auf Linux
- Node.js
- PHP
- Python
- Ruby

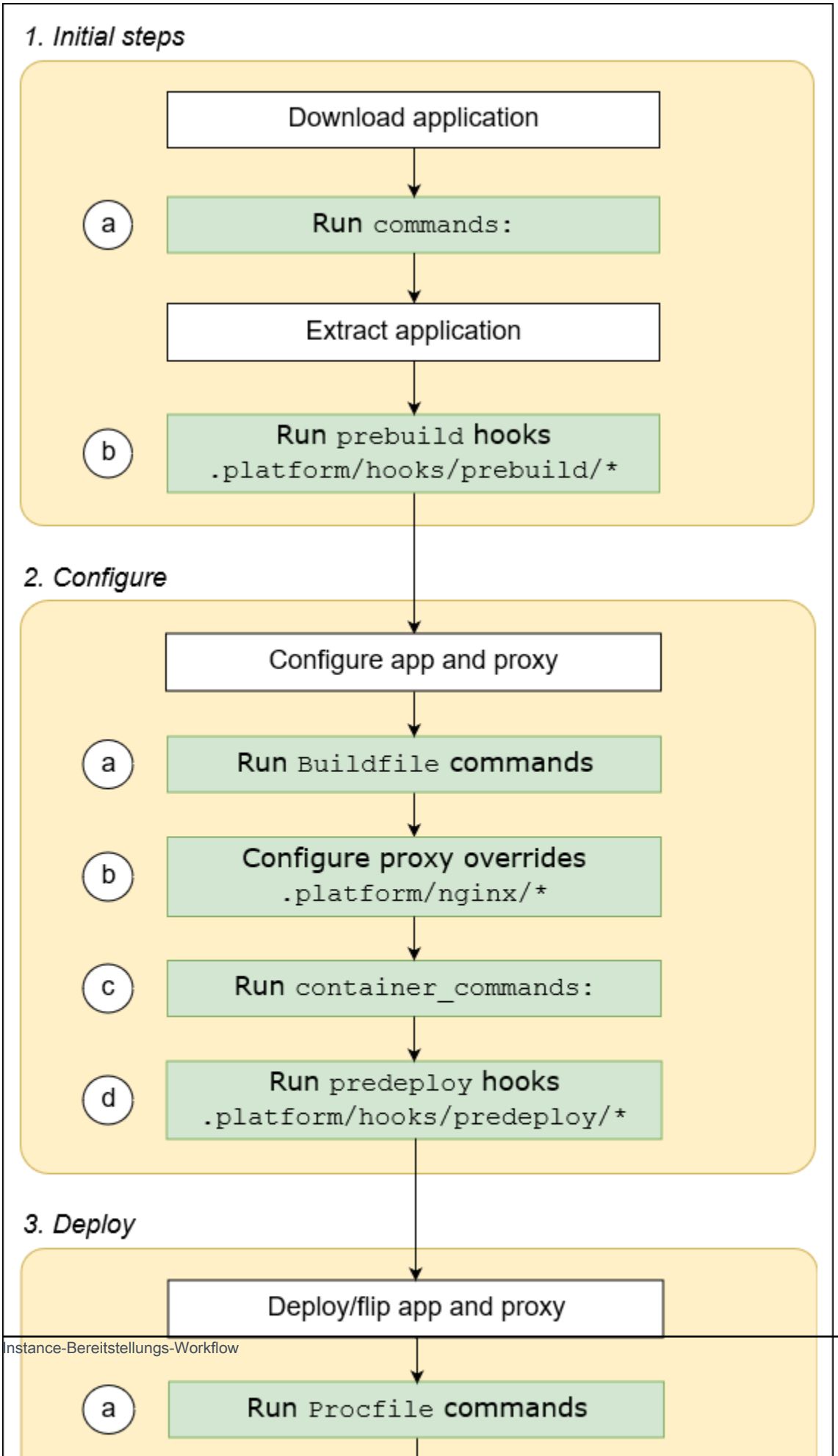
Instance-Bereitstellungs-Workflow

> **Note**
>
> Die Informationen in diesem Abschnitt gelten nicht für die Plattformzweige ECS auf Amazon Linux 2 und Amazon Linux 2023. Weitere Informationen finden Sie im folgenden Abschnitt, Workflow zur Instance-Bereitstellung für ECS auf Amazon Linux 2 und höher.

Da die Plattform Ihrer Umgebung auf viele Arten erweitert werden kann, ist es hilfreich, den Ablauf zu verstehen, wenn Elastic Beanstalk eine Instance bereitstellt oder eine Bereitstellung für eine Instance ausführt. Das folgende Diagramm zeigt diesen gesamten Bereitstellungsworkflow. Es zeigt die verschiedenen Phasen in einer Bereitstellung und die Schritte, die von Elastic Beanstalk in jeder Phase durchgeführt werden.

Hinweise

- Das Diagramm zeigt nicht den vollständigen Satz von Schritten, die Elastic Beanstalk während der Bereitstellung auf Umgebungs-Instances ausführt. Wir stellen dieses Diagramm zur Veranschaulichung bereit, um Ihnen die Reihenfolge und den Kontext für die Ausführung Ihrer Anpassungen zu liefern.
- Der Einfachheit halber zeigt das Diagramm nur die `.platform/hooks/*`-Hook-Unterverzeichnisse (für Anwendungsbereitstellungen) und nicht die `.platform/confighooks/*`-Hook-Unterverzeichnisse (für Konfigurationsbereitstellungen). Hooks in den zuletzt genannten Unterverzeichnissen werden während der gleichen Schritte ausgeführt wie die Hooks in den Unterverzeichnissen, die im Diagramm gezeigt werden.


1. Initial steps
Download application
a
Run commands:
Extract application
b
Run prebuild hooks
.platform/hooks/prebuild/*
2. Configure
Configure app and proxy
a
Run Buildfile commands
b
Configure proxy overrides
.platform/nginx/*
c
Run container_commands:
d
Run predeploy hooks
.platform/hooks/predeploy/*
3. Deploy
Deploy/flip app and proxy
a
Run Procfile commands

In der folgenden Liste werden die Bereitstellungsphasen und -schritte beschrieben.

1. Anfängliche Schritte

   Elastic Beanstalk lädt und extrahiert Ihre Anwendung. Nach jedem dieser Schritte führt Elastic Beanstalk einen Erweiterbarkeitsschritt aus.

   a. Führt Befehle aus, die im Abschnitt Befehle: einer beliebigen Konfigurationsdatei gefunden werden.

   b. Führt alle ausführbaren Dateien im Verzeichnis `.platform/hooks/prebuild` des Quellpakets aus (`.platform/confighooks/prebuild` für eine Konfigurationsbereitstellung).

2. Konfiguration

   Elastic Beanstalk konfiguriert Ihre Anwendung und den Proxy-Server.

   a. Führt die Befehle aus, die in der `Buildfile` in Ihrem Quell-Bundle gefunden wurden.

   b. Kopiert Ihre benutzerdefinierten Proxy-Konfigurationsdateien, sofern sich welche im Verzeichnis `.platform/nginx` Ihres Quell-Bundles befinden, an ihrem Laufzeitspeicherort.

   c. Führt Befehle aus, die im Abschnitt container_commands: einer beliebigen Konfigurationsdatei gefunden wurden.

   d. Führt alle ausführbaren Dateien im Verzeichnis `.platform/hooks/predeploy` des Quellpakets aus (`.platform/confighooks/predeploy` für eine Konfigurationsbereitstellung).

3. Bereitstellen

   Elastic Beanstalk stellt Ihre Anwendung und den Proxy-Server bereit und führt sie aus.

   a. Führt den Befehl aus, der in der `Procfile`-Datei in Ihrem Quell-Bundle gefunden wurde.

   b. Führt den Proxy-Server mit den benutzerdefinierten Proxy-Konfigurationsdateien aus, falls vorhanden, oder erneut aus.

   c. Führt alle ausführbaren Dateien im Verzeichnis `.platform/hooks/postdeploy` des Quellpakets aus (`.platform/confighooks/postdeploy` für eine Konfigurationsbereitstellung).

Workflow zur Instance-Bereitstellung für ECS auf Amazon Linux 2 und höher

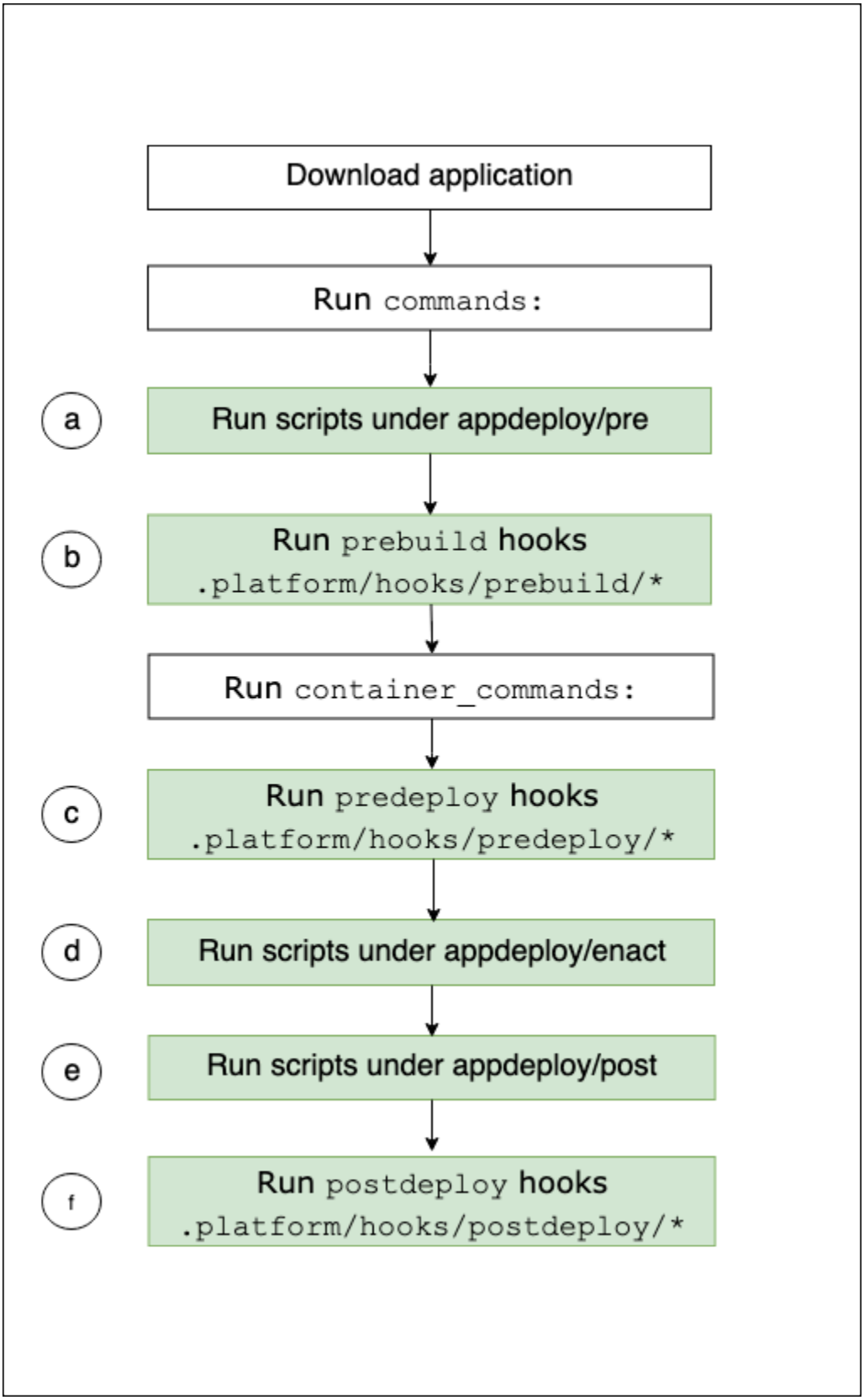
Im vorherigen Abschnitt werden die unterstützten Erweiterungsfunktionen in den Phasen des Workflows zur Anwendungsbereitstellung beschrieben. Es gibt einige Unterschiede bei den Docker-Plattformzweigen ECS auf Amazon Linux 2 und höher. In diesem Abschnitt wird erläutert, wie diese Konzepte für diesen spezifischen Plattformzweig gelten.

Da die Plattform Ihrer Umgebung auf viele Arten erweitert werden kann, ist es hilfreich, den Ablauf zu verstehen, wenn Elastic Beanstalk eine Instance bereitstellt oder eine Bereitstellung für eine Instance ausführt. Das folgende Diagramm zeigt diesen gesamten Bereitstellungs-Workflow für eine Umgebung, die auf den Plattformzweigen ECS auf Amazon Linux 2 und ECS auf Amazon Linux 2023 basiert. Es zeigt die verschiedenen Phasen in einer Bereitstellung und die Schritte, die von Elastic Beanstalk in jeder Phase durchgeführt werden.

Im Gegensatz zu dem im vorherigen Abschnitt beschriebenen Workflow unterstützt die Bereitstellungs-Konfigurations-Phase die folgenden Erweiterungsfunktionen nicht: `Buildfile`-Befehle, `Procfile`-Befehle, Reverse-Proxy-Konfiguration.

Hinweise

- Das Diagramm zeigt nicht den vollständigen Satz von Schritten, die Elastic Beanstalk während der Bereitstellung auf Umgebungs-Instances ausführt. Wir stellen dieses Diagramm zur Veranschaulichung bereit, um Ihnen die Reihenfolge und den Kontext für die Ausführung Ihrer Anpassungen zu liefern.
- Der Einfachheit halber zeigt das Diagramm nur die `.platform/hooks/*`-Hook-Unterverzeichnisse (für Anwendungsbereitstellungen) und nicht die `.platform/confighooks/*`-Hook-Unterverzeichnisse (für Konfigurationsbereitstellungen). Hooks in den zuletzt genannten Unterverzeichnissen werden während der gleichen Schritte ausgeführt wie die Hooks in den Unterverzeichnissen, die im Diagramm gezeigt werden.



Die folgende Liste enthält die Schritte des Bereitstellungs-Workflows.

a. Führt alle ausführbaren Dateien aus, die im `appdeploy/pre`-Verzeichnis unter `EBhooksDir` gefunden werden.
b. Führt alle ausführbaren Dateien im Verzeichnis `.platform/hooks/prebuild` des Quellpakets aus (`.platform/confighooks/prebuild` für eine Konfigurationsbereitstellung).
c. Führt alle ausführbaren Dateien im Verzeichnis `.platform/hooks/predeploy` des Quellpakets aus (`.platform/confighooks/predeploy` für eine Konfigurationsbereitstellung).
d. Führt alle ausführbaren Dateien aus, die im `appdeploy/enact`-Verzeichnis unter `EBhooksDir` gefunden werden.
e. Führt alle ausführbaren Dateien aus, die im `appdeploy/post`-Verzeichnis unter `EBhooksDir` gefunden werden.
f. Führt alle ausführbaren Dateien im Verzeichnis `.platform/hooks/postdeploy` des Quellpakets aus (`.platform/confighooks/postdeploy` für eine Konfigurationsbereitstellung).

Der Verweis auf `EBhooksDir` repräsentiert den Pfad des Plattform-Hooks-Verzeichnisses. Verwenden Sie zum Abrufen des Verzeichnispfad-Namens das Skripttool get-config in der Befehlszeile Ihrer Umgebungs-Instance wie gezeigt:

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k EBhooksDir
```

Plattform-Skripttools für Ihre Elastic Beanstalk Beanstalk-Umgebungen

In diesem Thema werden Tools beschrieben, die AWS Elastic Beanstalk für Umgebungen bereitgestellt werden, die Amazon Linux-Plattformen verwenden. Die Tools befinden sich auf den EC2 Amazon-Instances der Elastic Beanstalk Beanstalk-Umgebungen.

get-config

Verwenden Sie das `get-config` Tool, um Werte von Umgebungsvariablen im Klartext und andere Plattform- und Instance-Informationen abzurufen. Das Tool steht unter `/opt/elasticbeanstalk/bin/get-config` zur Verfügung.

get-config-Befehle

Jeder `get-config`-Tool-Befehl gibt einen bestimmten Typ von Informationen zurück. Verwenden Sie die folgende Syntax, um einen der Befehle des Tools auszuführen.

```
$ /opt/elasticbeanstalk/bin/get-config command [ options ]
```

Im folgenden Beispiel wird der Befehl `environment` ausgeführt:

```
$ /opt/elasticbeanstalk/bin/get-config environment -k PORT
```

Abhängig von dem gewählten Befehl und den gewählten Optionen gibt das Tool ein Objekt (JSON oder YAML) mit Schlüssel-Wert-Paaren oder einem einzelnen Wert zurück.

Sie können testen, `get-config` indem Sie SSH verwenden, um eine Verbindung zu einer EC2 Instance in Ihrer Elastic Beanstalk Beanstalk-Umgebung herzustellen.

> **Note**
>
> Wenn Sie `get-config` zum Testen ausführen, benötigen einige Befehle möglicherweise Stammbenutzerrechte, um auf die zugrunde liegenden Informationen zuzugreifen. Wenn Sie einen Zugriffsberechtigungsfehler erhalten, führen Sie den Befehl unter `sudo` erneut aus. Sie müssen `sudo` nicht hinzufügen, wenn Sie das Tool in den Skripts verwenden, die Sie in Ihrer Umgebung bereitstellen. Elastic Beanstalk führt alle Ihre Skripte als Stammbenutzer aus.

In den folgenden Abschnitten werden die Befehle des Tools beschrieben.

optionsettings – Konfigurationsoptionen

Der Befehl `get-config optionsettings` gibt ein Objekt zurück, das die Konfigurationsoptionen auflistet, die für die Umgebung festgelegt und von der Plattform auf Umgebungs-Instances verwendet werden. Sie sind nach Namespace organisiert.

```
$ /opt/elasticbeanstalk/bin/get-config optionsettings
{"aws:elasticbeanstalk:application:environment":
{"JDBC_CONNECTION_STRING":""},"aws:elasticbeanstalk:container:tomcat:jvmoptions":{"JVM
 Options":"","Xms":"256m","Xmx":"256m"},"aws:elasticbeanstalk:environment:proxy":
{"ProxyServer":"nginx","StaticFiles":
[""]},"aws:elasticbeanstalk:healthreporting:system":
{"SystemType":"enhanced"},"aws:elasticbeanstalk:hostmanager":
{"LogPublicationControl":"false"}}
```

Um einen bestimmten Wert für die Konfigurationsoption zurückzugeben, verwenden Sie die Option `--namespace` (`-n`), um einen Namespace anzugeben, und die Option `--option-name` (`-o`), um einen Optionsnamen anzugeben.

```
$ /opt/elasticbeanstalk/bin/get-config optionsettings -
n aws:elasticbeanstalk:container:php:phpini -o memory_limit
256M
```

environment – Umgebungseigenschaften

Der `get-config environment` Befehl gibt ein Objekt zurück, das eine Liste von Umgebungseigenschaften enthält, darunter sowohl benutzerkonfigurierte als auch von Elastic Beanstalk bereitgestellte Eigenschaften. Die vom Benutzer konfigurierten Eigenschaften werden in der Konsole als Klartext oder mit dem Namespace der Konfigurationsoption definiert. aws:elasticbeanstalk:application:environment

```
$ /opt/elasticbeanstalk/bin/get-config environment
{"JDBC_CONNECTION_STRING":"","RDS_PORT":"3306","RDS_HOSTNAME":"anj9aw1b0tbj6b.cijbpanmxz5u.us-
west-2.rds.amazonaws.com","RDS_USERNAME":"testusername","RDS_DB_NAME":"ebdb","RDS_PASSWORD":"te
```

Beispielsweise bietet Elastic Beanstalk Umgebungseigenschaften, mit denen die Verbindung zu einer integrierten Amazon RDS-DB-Instance (zum Beispiel `RDS_HOSTNAME`) hergestellt werden kann. Diese RDS-Verbindungseigenschaften werden in der Ausgabe von `get-config environment` angezeigt. Sie werden jedoch nicht in der Ausgabe von `get-config optionsettings`angezeigt. Dies liegt daran, dass sie in Konfigurationsoptionen nicht festgelegt wurden.

Wenn Sie eine bestimmte Umgebungseigenschaft zurückzugeben möchten, verwenden Sie die Option `--key` (`-k`), um einen Eigenschaftsschlüssel anzugeben.

```
$ /opt/elasticbeanstalk/bin/get-config environment -k TESTPROPERTY
testvalue
```

> **Note**
>
> Das `get-config` Tool kann keine Umgebungsvariablen abrufen, die Geheimnisse speichern. Weitere Hinweise zum programmgesteuerten Abrufen von Werten aus geheimen Speichern oder Parameterspeichern finden Sie unter Verwendung von Secrets Manager oder. Verwendung des Systems Manager Parameter Store

container – Konfigurationswerte auf der Instance

Der Befehl `get-config container` gibt ein Objekt zurück, das Plattform- und Umgebungskonfigurationswerte für Umgebungs-Instances auflistet.

Das folgende Beispiel zeigt die Ausgabe des Befehls in einer Amazon Linux 2-Tomcat-Umgebung.

```
$ /opt/elasticbeanstalk/bin/get-config container
{"common_log_list":["/var/log/eb-engine.log","/var/log/eb-
hooks.log"],"default_log_list":["/var/log/nginx/access.log","/var/log/nginx/
error.log"],"environment_name":"myenv-1da84946","instance_port":"80","log_group_name_prefix":"/
aws/elasticbeanstalk","proxy_server":"nginx","static_files":
[""],"xray_enabled":"false"}
```

Um den Wert eines bestimmten Schlüssels zurückzugeben, verwenden Sie die Option `--key` (`-k`), um den Schlüssel anzugeben.

```
$ /opt/elasticbeanstalk/bin/get-config container -k environment_name
myenv-1da84946
```

addons – Add-On-Konfigurationswerte

Der Befehl `get-config addons` gibt ein Objekt zurück, das Konfigurationsinformationen von Umgebungs-Add-Ons enthält. Verwenden Sie ihn, um die Konfiguration einer Amazon RDS-Datenbank abzurufen, die der Umgebung zugeordnet ist.

```
$ /opt/elasticbeanstalk/bin/get-config addons
{"rds":{"Description":"RDS Environment variables","env":
{"RDS_DB_NAME":"ebdb","RDS_HOSTNAME":"ea13k2wimu1dh8i.c18mnpu5rwvg.us-
east-2.rds.amazonaws.com","RDS_PASSWORD":"password","RDS_PORT":"3306","RDS_USERNAME":"user"}}}
```

Sie können das Ergebnis auf zwei Arten einschränken. Um Werte für ein bestimmtes Add-On abzurufen, geben Sie mit der Option `--add-on` (`-a`) den Namen des Add-On an.

```
$ /opt/elasticbeanstalk/bin/get-config addons -a rds
{"Description":"RDS Environment variables","env":
{"RDS_DB_NAME":"ebdb","RDS_HOSTNAME":"ea13k2wimu1dh8i.c18mnpu5rwvg.us-
east-2.rds.amazonaws.com","RDS_PASSWORD":"password","RDS_PORT":"3306","RDS_USERNAME":"user"}}
```

Um den Wert eines bestimmten Schlüssels in einem Add-On zurückzugeben, fügen Sie die Option `--key` (`-k`) hinzu, um den Schlüssel anzugeben.

```
$ /opt/elasticbeanstalk/bin/get-config addons -a rds -k RDS_DB_NAME
ebdb
```

platformconfig – Konstante Konfigurationswerte

Der Befehl `get-config platformconfig` gibt ein Objekt zurück, das Plattformkonfigurationsinformationen enthält, die mit der Plattformversion konstant sind. Die Ausgabe ist in allen Umgebungen, in denen dieselbe Plattformversion ausgeführt wird, gleich. Das Ausgabeobjekt des Befehls verfügt über zwei eingebettete Objekte:

- `GeneralConfig` – Enthält Informationen, die in den neuesten Versionen aller Amazon-Linux-2- und Amazon-Linux-2023-Plattformzweige konstant sind.
- `PlatformSpecificConfig` – Enthält Informationen, die für die Plattformversion konstant sind und für diese spezifisch sind.

Das folgende Beispiel zeigt die Ausgabe des Befehls in einer Umgebung, die den Corretto 11-Plattformzweig Tomcat 8.5 verwendet.

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig
{"GeneralConfig":{"AppUser":"webapp","AppDeployDir":"/var/app/
current/","AppStagingDir":"/var/app/
staging/","ProxyServer":"nginx","DefaultInstancePort":"80"},"PlatformSpecificConfig":
{"ApplicationPort":"8080","JavaVersion":"11","TomcatVersion":"8.5"}}
```

Um den Wert eines bestimmten Schlüssels zurückzugeben, verwenden Sie die Option `--key` (`-k`), um den Schlüssel anzugeben. Diese Schlüssel sind in den beiden eingebetteten Objekten eindeutig. Sie müssen das Objekt, das den Schlüssel enthält, nicht angeben.

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k AppStagingDir
/var/app/staging/
```

get-config-Ausgabeoptionen

Verwenden Sie die Option `--output`, um das Format des Ausgabeobjekts anzugeben. Gültige Werte sind `JSON` (Standard) und `YAML`. Dies ist eine globale Option. Sie müssen es vor dem Befehlsnamen angeben.

Im folgenden Beispiel werden Werte für Konfigurationsoptionen im YAML-Format zurückgegeben.

```
$ /opt/elasticbeanstalk/bin/get-config --output YAML optionsettings
aws:elasticbeanstalk:application:environment:
  JDBC_CONNECTION_STRING: ""
aws:elasticbeanstalk:container:tomcat:jvmoptions:
  JVM Options: ""
  Xms: 256m
  Xmx: 256m
aws:elasticbeanstalk:environment:proxy:
  ProxyServer: nginx
  StaticFiles:
        - ""
aws:elasticbeanstalk:healthreporting:system:
  SystemType: enhanced
aws:elasticbeanstalk:hostmanager:
  LogPublicationControl: "false"
```

Pkg-Repo

> **Note**
>
> Das `pkg-repo`-Tool ist nicht für Umgebungen verfügbar, die auf Amazon-Linux-2023-Plattformen basieren. Sie können Paket- und Betriebssystemupdates jedoch manuell auf eine AL2 023-Instanz anwenden. Weitere Informationen finden Sie unter Verwalten von Paketen und Betriebssystem-Updates im Benutzerhandbuch für Amazon Linux 2023

In dringenden Fällen müssen Sie Ihre EC2 Amazon-Instances möglicherweise mit einem Amazon Linux 2-Sicherheitspatch aktualisieren, der noch nicht mit den erforderlichen Elastic Beanstalk-Plattformversionen veröffentlicht wurde. Sie können standardmäßig keine manuelle Aktualisierung für Ihre Elastic-Beanstalk-Umgebungen durchführen. Dies liegt daran, dass die Plattformversionen an eine bestimmte Version des Amazon Linux 2-Repositorys gesperrt sind. Diese Sperre stellt sicher, dass Instances unterstützte und konsistente Softwareversionen ausführen. Für dringende Fälle erlaubt das Tool `pkg-repo` eine Problemumgehung. Sie können Yum-Pakete in Amazon Linux 2 manuell aktualisieren, wenn Sie es in einer Umgebung installieren müssen, bevor es in einer neuen Elastic-Beanstalk-Plattformversion veröffentlicht wird.

Das Tool `pkg-repo` der Amazon Linux 2-Plattformen bietet die Möglichkeit, die `yum` Paket-Repositorys zu entsperren. Sie können dann manuell eine yum update für einen Sicherheitspatch durchführen. Umgekehrt können Sie der Aktualisierung folgen, indem Sie das Tool verwenden, um

die Yum-Paket-Repositorys zu sperren, um weitere Aktualisierungen zu verhindern. Das `pkg-repo` Tool ist im `/opt/elasticbeanstalk/bin/pkg-repo` Verzeichnis aller EC2 Instances in Ihren Elastic Beanstalk Beanstalk-Umgebungen verfügbar.

Änderungen, die das `pkg-repo` Tool verwenden, werden nur an der EC2 Instance vorgenommen, auf der das Tool verwendet wird. Sie wirken sich nicht auf andere Instances aus und verhindern nicht zukünftige Aktualisierungen der Umgebung. Die Beispiele später in diesem Thema erklären, wie Sie die Änderungen auf alle Instances anwenden können, indem Sie die `pkg-repo`-Befehle aus Skripts und Konfigurationsdateien verwenden.

Warning

Wir empfehlen dieses Tool für die meistenBenutzer. Alle manuellen Änderungen, die auf eine Version der freigeschalteten Plattform angewendet werden, gelten als Out-of-Band. Diese Option ist nur für Benutzer unter dringenden Umständen geeignet, die die folgenden Risiken akzeptieren können:

- Für Paketversionen können nicht garantiert werden, dass sie über alle Instances hinweg in Ihren Umgebungen konsistent sind.
- Wir empfehlen das Tool `pkg-repo` für die meisten Benutzer. Sie wurden nicht auf Elastic Beanstalk unterstützten Plattformen getestet und überprüft.

Wir empfehlen dringend, Best Practices anzuwenden, die Test- und Backout-Pläne beinhalten. Um Best Practices zu vereinfachen, können Sie die Elastic Beanstalk Beanstalk-Konsole und die EB-CLI verwenden, um eine Umgebung zu klonen und eine Umgebung auszutauschen. URLs Weitere Informationen zur Verwendung dieser Vorgänge finden Sie unter BluBlue/Green-Bereitstellungen im Kapitel Managing environments (Verwalten von Umgebungen) in diesem Handbuch.

Wenn Sie vorhaben, Yum-Repository-Konfigurationsdateien manuell zu bearbeiten, führen Sie zuerst das Tool `pkg-repo` aus. Das Tool `pkg-repo` funktioniert in einer Amazon Linux 2-Umgebung mit manuell bearbeiteten Yum-Repository-Konfigurationsdateien möglicherweise nicht wie vorgesehen. Dies liegt daran, dass das Tool die Konfigurationsänderungen möglicherweise nicht erkennt.

Weitere Informationen zum Amazon Linux-Paket-Repository finden Sie unter dem Thema Paket-Repository im EC2 Amazon-Benutzerhandbuch.

pkg-repo-Befehle

Verwenden Sie die folgende Syntax, um einen der Befehle des Tools `pkg-repo` auszuführen.

```
$ /opt/elasticbeanstalk/bin/pkg-repo command [options]
```

Folgende `pkg-repo`-Befehle müssen ausgeführt werden:

- lock— Sperrt das `yum`-Paketrepositorys für eine bestimmte Version
- unlock— Entsperrt das `yum`-Paketrepositorys aus einer bestimmten Version
- status— Listet alle `yum`-Paket-Repositorys und ihren aktuellen Sperrstatus auf
- help— Zeigt allgemeine Hilfe oder Hilfe für einen Befehl an

Die Optionen gelten für die folgenden Befehle:

- `lock`, `unlock` und `status` — Optionen: `-h`, `--help`oder keine (Standard).
- `help` — Optionen: `lock`, `unlock`, `status` oder keine (Standard).

Im folgenden Beispiel wird der Befehl unlock ausgeführt:

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo unlock
Amazon Linux 2 core package repo successfully unlocked
Amazon Linux 2 extras package repo successfully unlocked
```

Im folgenden Beispiel wird der Befehl lock ausgeführt:

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo lock
Amazon Linux 2 core package repo successfully locked
Amazon Linux 2 extras package repo successfully locked
```

Im folgenden Beispiel wird der Befehl status ausgeführt:

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo status
Amazon Linux 2 core package repo is currently UNLOCKED
Amazon Linux 2 extras package repo is currently UNLOCKED
```

Im folgenden Beispiel wird der help-Befehl für den lock-Befehl ausgeführt.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help lock
```

Im folgenden Beispiel wird der help-Befehl für das Tool `pkg-repo` ausgeführt.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help
```

Sie können `pkg-repo` testen, indem Sie SSH verwenden, um eine Verbindung zu einer Instance in Ihrer Elastic Beanstalk-Umgebung herzustellen. Eine SSH-Option ist der EB CLI eb ssh Befehl.

> **Note**
>
> Die Tool `pkg-repo` benötigt zum Ausführen Stammbenutzer-Berechtigungen. Wenn Sie einen Zugriffsberechtigungsfehler erhalten, führen Sie den Befehl unter `sudo` erneut aus. Sie müssen `sudo` nicht hinzufügen, wenn Sie das Tool in den Skripts verwenden, die Sie in Ihrer Umgebung bereitstellen. Elastic Beanstalk führt alle Ihre Skripte als Stammbenutzer aus.

Beispiele für pkg-repo

Der vorherige Abschnitt enthält Befehlszeilenbeispiele für Tests auf einer einzelnen EC2 Instanz einer Elastic Beanstalk Beanstalk-Umgebung. Diese Vorgehensweise kann beim Testen hilfreich sein. Es aktualisiert jedoch jeweils nur eine Instance, daher ist es nicht praktisch, Änderungen an allen Instances in einer Umgebung anzuwenden.

Ein pragmatischerer Ansatz ist die Verwendung von Plattform-Hook-Skripten oder einer `.ebextensions` Konfigurationsdatei, um die Änderungen auf alle Instances konsistent anzuwenden.

Im folgenden Beispiel wird `pkg-repo` aus einer Konfigurationsdatei im `.ebextensions`-Folder aufgerufen. Elastic Beanstalk führt die Befehle in der `update_package.config`-Datei aus, wenn Sie das Quell-Bundle der Anwendung bereitstellen.

```
.ebextensions
### update_package.config
```

Um die neueste Version des Docker-Pakets zu erhalten, gibt diese Konfiguration das Docker-Paket im yum update-Befehl an.

```
### update_package.config ###
```

```
commands:
  update_package:
    command: |
      /opt/elasticbeanstalk/bin/pkg-repo unlock
      yum update docker -y
      /opt/elasticbeanstalk/bin/pkg-repo lock
      yum clean all -y
      rm -rf /var/cache/yum
```

Bei dieser Konfiguration werden keine Pakete im yum update-Befehl angegeben. Alle verfügbaren Updates werden als Ergebnis angewendet.

```
### update_package.config ###

commands:
  update_package:
    command: |
      /opt/elasticbeanstalk/bin/pkg-repo unlock
      yum update -y
      /opt/elasticbeanstalk/bin/pkg-repo lock
      yum clean all -y
      rm -rf /var/cache/yum
```

Das folgende Beispiel ruft `pkg-repo` von einem Bash-Skript als Plattform-Hook auf. Elastic Beanstalk führt die `update_package.sh`-Skriptdatei aus, die sich in dem Unterverzeichnis `prebuild` befindet.

```
.platform
### hooks
    ### prebuild
        ### update_package.sh
```

Um die neueste Version des Docker-Pakets zu erhalten, gibt dieses Skript das Docker-Paket im yum update-Befehl an. Wenn der Paketname ausgelassen wird, werden alle verfügbaren Updates angewendet. Das vorherige Beispiel der Konfigurationsdatei zeigt dies.

```
### update_package.sh ###

#!/bin/bash
```

```
/opt/elasticbeanstalk/bin/pkg-repo unlock
yum update docker -y
/opt/elasticbeanstalk/bin/pkg-repo lock
yum clean all -y
rm -rf /var/cache/yum
```

download-source-bundle (Nur Amazon Linux AMI)

Auf Amazon Linux AMI-Plattformbranchen (Vorgänger von Amazon Linux 2) bietet Elastic Beanstalk ein zusätzliches Tool, `download-source-bundle`. Verwenden Sie dieses, um den Quellcode Ihrer Anwendung während der Bereitstellung Ihrer Plattform herunterzuladen. Das Tool steht unter `/opt/elasticbeanstalk/bin/download-source-bundle` zur Verfügung.

Das Beispielskript `00-unzip.sh` befindet sich auf Umgebungs-Instances im Ordner `appdeploy/pre`. Es demonstriert, wie `download-source-bundle` verwendet wird, um den Anwendungsquellcode während der Bereitstellung in den Ordner `/opt/elasticbeanstalk/deploy/appsource` herunterzuladen.

Erweitern von Elastic Beanstalk-Linux-Plattformen

In diesem Thema wird beschrieben, wie Sie Ihre Linux-Plattformen mit Ihren eigenen Befehlen, Skripten, Software und Konfigurationen erweitern können. Möglicherweise müssen Sie Ihre Plattform erweitern, um den Standard-Proxyserver und die Konfiguration zu ändern. Oder Sie müssen möglicherweise anpassen, wie die Plattform Ihre Anwendung erstellt oder startet.

Themen

- Buildfile und Procfile
- Plattform-Hooks
- Konfigurationsdateien
- Reverse-Proxy-Konfiguration
- Anwendungsbeispiel mit Erweiterungen

Buildfile und Procfile

Auf einigen Plattformen können Sie anpassen, wie Sie Ihre Anwendung erstellen oder vorbereiten und die Prozesse angeben, die Ihre Anwendung ausführen. In jedem einzelnen Plattformthema wird

Buildfile and/or Procfile ausdrücklich erwähnt, sofern die Plattform sie unterstützt. Suchen Sie nach Ihrer spezifischen Plattform unter Plattformen.

Syntax und Semantik sind für alle unterstützenden Plattformen identisch und sind wie auf dieser Seite beschrieben. Einzelne Plattformthemen erwähnen die spezifische Verwendung dieser Dateien zum Erstellen und Ausführen von Anwendungen in ihren jeweiligen Sprachen.

Buildfile

Um einen benutzerdefinierten Build- und Konfigurationsbefehl für Ihre Anwendung anzugeben, legen Sie eine Datei mit dem Namen `Buildfile` im Stammverzeichnis Ihrer Anwendungsquelle ab. Beachten Sie beim Dateinamen die Groß- und Kleinschreibung. Verwenden Sie die folgende Syntax für Ihre `Buildfile`.

```
<process_name>: <command>
```

Der Befehl in Ihrer `Buildfile` muss folgendem regulären Ausdruck entsprechen: `^[A-Za-z0-9_-]+:\s*[^\s].*$`

Elastic Beanstalk überwacht nicht die Anwendung, die mit einer `Buildfile` ausgeführt wird. Verwenden Sie eine `Buildfile` für Befehle, die für kurze Zeiträume ausgeführt und nach Abschluss ihrer Aufgaben beendet werden. Verwenden Sie für länger dauernde Anwendungsprozesse, die nicht beendet werden sollten, eine Procfile.

Alle Pfade in der `Buildfile` sind relativ zum Stammverzeichnis des Quell-Bundle. Im folgenden Beispiel einer `Buildfile` ist `build.sh` ein Shell-Skript, das sich im Stammverzeichnis des Quell-Bundle befindet.

Example Buildfile

```
make: ./build.sh
```

Wenn Sie benutzerdefinierte Build-Schritte bereitstellen möchten, empfehlen wir Ihnen, für alles, außer für die einfachsten Befehle, anstelle einer `Buildfile` `predeploy`-Plattform-Hooks zu verwenden. Plattform-Hooks ermöglichen umfangreichere Skripts und eine bessere Fehlerbehandlung. Plattformhaken werden im nächsten Abschnitt beschrieben.

Procfile

Um benutzerdefinierte Befehle zum Starten und Ausführen der Anwendung anzugeben, legen Sie eine Datei mit dem Namen `Procfile` im Stammverzeichnis Ihrer Anwendungsquelle ab. Beachten Sie beim Dateinamen die Groß- und Kleinschreibung. Verwenden Sie die folgende Syntax für Ihre `Procfile`. Sie können einen oder mehrere Befehle angeben.

```
<process_name1>: <command1>
<process_name2>: <command2>
...
```

Jede Zeile in Ihrer `Procfile` muss folgendem regulären Ausdruck entsprechen: `^[A-Za-z0-9_-]+:\s*[^\s].*$`

Verwenden Sie eine `Procfile` für länger dauernde Anwendungsprozesse, die nicht beendet werden sollten. Elastic Beanstalk erwartet von Prozessen, die aus der `Procfile` ausgeführt werden, dass sie kontinuierlich laufen. Elastic Beanstalk überwacht diese Prozesse und startet alle Prozesse, die beendet werden, neu. Für kurz laufende Prozesse verwenden Sie eine Buildfile.

Alle Pfade in der `Procfile` sind relativ zum Stammverzeichnis des Quell-Bundle. Im folgenden Beispiel definiert `Procfile` drei Prozesse. Die erste, die im Beispiel `web` genannt wird, ist die Hauptwebanwendung.

Example Procfile

```
web: bin/myserver
cache: bin/mycache
foo: bin/fooapp
```

Elastic Beanstalk konfiguriert den Proxy-Server so, dass Anforderungen an Ihre Haupt-Web-Anwendung über Port 5000 weitergeleitet werden. Sie können diese Port-Nummer konfigurieren. Eine übliche Verwendung für `Procfile` besteht darin, diese Portnummer als Befehlsargument an Ihre Anwendung zu übergeben. Einzelheiten zur Proxykonfiguration finden Sie unter. Reverse-Proxy-Konfiguration

Elastic Beanstalk erfasst Standardausgabe- und Fehler-Streams von `Procfile`-Prozessen in Protokolldateien. Elastic Beanstalk benennt die Protokolldateien nach dem Prozess und speichert sie in `/var/log`. Beispielsweise generiert der `web`-Prozess aus dem vorherigen Beispiel Protokolle mit den Namen `web-1.log` und `web-1.error.log` für `stdout` bzw. `stderr`.

Plattform-Hooks

Plattform-Hooks wurden speziell entwickelt, um die Plattform Ihrer Umgebung zu erweitern. Es handelt sich um benutzerdefinierte Skripts und andere benutzerdefinierte ausführbare Dateien, die Sie als Teil des Quellcodes Ihrer Anwendung bereitstellen und die von Elastic Beanstalk in verschiedenen Phasen der Instance-Bereitstellung ausgeführt werden.

> **Note**
>
> Plattform-Hooks werden auf Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2) nicht unterstützt.

Plattform-Hooks für die Anwendungsbereitstellung

Eine Anwendungsbereitstellung erfolgt, wenn Sie ein neues Quellpaket zur Bereitstellung angeben oder wenn Sie eine Konfigurationsänderung ausführen, für die alle Instances der Umgebung beendet und neu erstellt werden müssen.

Um Plattform-Hooks bereitzustellen, die während einer Anwendungsbereitstellung ausgeführt werden, speichern Sie die Dateien in einem der folgenden Unterverzeichnisse des Verzeichnisses `.platform/hooks` im Quellpaket.

- `prebuild` – Die hier aufgeführten Dateien werden ausgeführt, nachdem die Elastic Beanstalk-Plattform-Engine das Anwendungs-Quell-Bundle heruntergeladen und extrahiert hat und bevor die Anwendung und der Webserver eingerichtet und konfiguriert werden.

  Die `prebuild`-Dateien werden ausgeführt, nachdem alle Befehle ausgeführt wurden, die im Abschnitt Commands einer beliebigen Konfigurationsdatei gefunden wurden, und bevor `Buildfile`-Befehle ausgeführt werden.
- `predeploy` – Die hier aufgeführten Dateien werden ausgeführt, nachdem die Elastic Beanstalk-Plattform-Engine die Anwendung und den Webserver eingerichtet und konfiguriert hat und bevor sie an ihrem endgültigen Laufzeitspeicherort bereitgestellt werden.

  Die `predeploy`-Dateien werden nach dem Ausführen von Befehlen im Abschnitt container_commands einer beliebigen Konfigurationsdatei und vor dem Ausführen von `Procfile`-Befehlen ausgeführt.
- `postdeploy` – Die hier aufgeführten Dateien werden ausgeführt, nachdem die Elastic Beanstalk-Plattform-Engine die Anwendung und den Proxyserver bereitgestellt hat.

Dies ist der letzte Schritt im Bereitstellungsworkflow.

Plattform-Hooks für die Konfigurationsbereitstellung

Eine Konfigurationsbereitstellung erfolgt, wenn Sie Konfigurationsänderungen ausführen, mit denen die Instances der Umgebung aktualisiert, jedoch nicht neu erstellt werden. Die folgenden Optionsaktualisierungen führen zu einer Konfigurationsaktualisierung.

- Umgebungseigenschaften und plattformspezifische Einstellungen
- Statische Dateien
- AWS X-Ray Daemon
- Protokollspeicher und Streamen
- Anwendungs-Port (Einzelheiten finden Sie unterReverse-Proxy-Konfiguration)

Um Hooks bereitzustellen, die während einer Konfigurationsbereitstellung ausgeführt werden, speichern Sie sie im Verzeichnis `.platform/confighooks` im Quellpaket. Es werden die gleichen drei Unterverzeichnisse wie für Hooks für die Anwendungsbereitstellung verwendet.

Weitere Informationen zu Plattform-Hooks

Bei Hook-Dateien kann es sich um Binär- oder Skriptdateien handeln, die mit einer `#!`-Zeile mit dem Interpreter-Pfad beginnen, z. B. `#!/bin/bash`. Alle Dateien müssen über die Ausführungsberechtigung verfügen. Legen Sie mit `chmod +x` die Ausführungsberechtigung für Ihre Hook-Dateien fest. Für alle auf Amazon Linux 2023 und Amazon Linux 2 basierenden Plattformversionen, die am oder nach dem 29. April 2022 veröffentlicht wurden, gewährt Elastic Beanstalk automatisch Ausführungsberechtigungen für alle Plattform-Hook-Skripts. In diesem Fall müssen Sie Ausführungsberechtigungen nicht manuell erteilen. Eine Liste dieser Plattformversionen finden Sie in den Linux-Versionshinweisen vom 29. April 2022 im AWS Elastic Beanstalk -Leitfaden für Versionshinweise.

Elastic Beanstalk führt in jedem dieser Verzeichnisse Dateien in lexikographischer Reihenfolge der Dateinamen aus. Alle Dateien werden als `root`-Benutzer ausgeführt. Das aktuelle Arbeitsverzeichnis (cwd) für Plattform-Hooks ist das Stammverzeichnis der Anwendung. Für `prebuild`- und `predeploy`-Dateien ist es das Anwendungs-Staging-Verzeichnis, und für `postdeploy`-Dateien ist es das aktuelle Anwendungsverzeichnis. Wenn eine der Dateien fehlschlägt (mit einem Exit-Code ungleich Null beendet wird), wird die Bereitstellung abgebrochen und schlägt fehl.

Ein Platform-Hooks-Textskript schlägt möglicherweise fehl, wenn es Zeilenumbruchzeichen (Windows Carriage Return/Line Feed, CRLF) enthält. Wenn eine Datei auf einem Windows-Host gespeichert und dann auf einen Linux-Server übertragen wurde, kann sie Windows-CRLF-Zeilenumbrüche enthalten. Für Plattformen, die am oder nach dem 29. Dezember 2022 veröffentlicht wurden, konvertiert Elastic Beanstalk Windows-CRLF-Zeichen automatisch in Zeilenumbruchzeichen von Linux-Line Feed(LF) in Platform-Hooks-Textdateien. Wenn Ihre Anwendung auf Amazon-Linux-2-Plattformen läuft, die vor diesem Datum veröffentlicht wurden, müssen Sie die Windows-CRLF-Zeichen in Linux-LF-Zeichen konvertieren. Eine Möglichkeit, dies zu erreichen, besteht darin, die Skriptdatei auf einem Linux-Host zu erstellen und zu speichern. Tools, die diese Zeichen konvertieren, sind auch im Internet verfügbar.

Hook-Dateien können auf alle Umgebungseigenschaften zugreifen, die Sie in den Anwendungsoptionen definiert haben, sowie auf die Systemumgebungsvariablen `HOME`, `PATH` und `PORT`.

Um die Werte von Umgebungsvariablen und anderen Konfigurationsoptionen für Ihre Skripts für Plattform-Hooks abzurufen, können Sie das `get-config`-Dienstprogramm verwenden, das Elastic Beanstalk auf Umgebungs-Instances verwendet. Details hierzu finden Sie unter the section called “Plattform-Skript-Tools”.

Konfigurationsdateien

Sie können Konfigurationsdateien zum Verzeichnis `.ebextensions` des Quellcodes Ihrer Anwendung hinzufügen, um verschiedene Aspekte Ihrer Elastic Beanstalk-Umgebung zu konfigurieren. Mit Konfigurationsdateien können Sie unter anderem Software und andere Dateien auf den Instances Ihrer Umgebung anpassen und Initialisierungsbefehle für die Instances ausführen. Weitere Informationen finden Sie unter the section called “Linux-Server”.

Sie können Konfigurationsoptionen auch mithilfe von Konfigurationsdateien festlegen. Viele der Optionen steuern das Plattformverhalten, und einige dieser Optionen sind plattformspezifisch.

Für Plattformen, die auf Amazon Linux 2 und Amazon Linux 2023 basieren, empfiehlt sich die Verwendung von Buildfile, Procfile und Plattform-Hooks zum Konfigurieren und Ausführen von benutzerdefiniertem Code auf Ihren Umgebungs-Instances während der Instance-Bereitstellung. Diese Mechanismen werden in den vorherigen Abschnitten auf dieser Seite beschrieben. Sie können in `.ebextensions`-Konfigurationsdateien weiterhin Befehle und Containerbefehle verwenden, aber sie sind nicht mehr so einfach in der Verwendung. Beispielsweise kann das Schreiben von Befehlsskripts in einer YAML-Datei vom Syntaxstandpunkt her eine Herausforderung darstellen. Sie

müssen weiterhin `.ebextensions` Konfigurationsdateien für jedes Skript verwenden, das einen Verweis auf eine AWS CloudFormation Ressource benötigt.

Reverse-Proxy-Konfiguration

Alle Amazon-Linux-2- und Amazon-Linux-2023-Plattformversionen nutzen nginx als standardmäßigen Reverse-Proxy-Server. Die Plattformen Tomcat, Node.js, PHP und Python unterstützen ebenfalls Apache HTTPD als Alternative. Um Apache auf diesen Plattformen auszuwählen, legen Sie die Option `ProxyServer` im Namespace `aws:elasticbeanstalk:environment:proxy` auf `apache` fest. Alle Plattformen ermöglichen eine einheitliche Konfiguration des Proxy-Servers wie in diesem Abschnitt beschrieben.

> **Note**
>
> Im Fall von Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2) müssen Sie Proxy-Server möglicherweise auf eine andere Art konfigurieren. Diese Legacy-Details finden Sie in den jeweiligen Plattform-Themen in diesem Handbuch.

Elastic Beanstalk konfiguriert den Proxyserver auf den Instances Ihrer Umgebung, um Webdatenverkehr an die Hauptwebanwendung über die Stamm-URL der Umgebung weiterzuleiten, z. B. `http://my-env.elasticbeanstalk.com`.

Standardmäßig wird der Proxy von Elastic Beanstalk so konfiguriert, dass alle Anforderungen an Port 80 an Ihre primäre Webanwendung unter Port 5000 weitergeleitet werden. Sie können diese Portnummer konfigurieren, indem Sie die `PORT`-Umgebungseigenschaft mithilfe des Namespace aws:elasticbeanstalk:application:environment in einer Konfigurationsdatei festlegen, wie im folgenden Beispiel gezeigt.

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application:environment
    option_name:  PORT
    value:  <main_port_number>
```

Weitere Informationen zum Festlegen von Umgebungsvariablen für Ihre Anwendung finden Sie unter the section called “Optionseinstellungen”.

Ihre Anwendung sollte den Port überwachen, der dafür im Proxy konfiguriert ist. Wenn Sie den Standardport mithilfe der Umgebungseigenschaft `PORT` ändern, kann der Code durch

Lesen des Wertes der Umgebungsvariablen `PORT` darauf zugreifen. Rufen Sie beispielsweise `os.Getenv("PORT")` in Go oder `System.getenv("PORT")` in Java auf. Wenn Sie Ihren Proxy so konfigurieren, dass Datenverkehr an mehrere Anwendungsprozesse gesendet wird, können Sie mehrere Umgebungseigenschaften konfigurieren und deren Werte sowohl in der Proxykonfiguration als auch in Ihrem Anwendungscode verwenden. Eine weitere Möglichkeit besteht darin, den Port-Wert als Befehlsargument in der `Procfile` an den Prozess zu übergeben. Weitere Informationen finden Sie unter Buildfile und Procfile.

Konfigurieren von nginx

Elastic Beanstalk verwendet nginx als Standard-Reverse-Proxy, um die Anwendung dem Elastic Load Balancing-Load Balancer zuzuweisen. Elastic Beanstalk bietet eine nginx-Standardkonfiguration, die Sie erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

> **Note**
>
> Wenn Sie eine `.conf`-Konfigurationsdatei von nginx hinzufügen oder bearbeiten, müssen Sie sie als UTF-8 codieren.

Um die nginx-Standardkonfiguration von Elastic Beanstalk zu erweitern, fügen Sie `.conf`-Konfigurationsdateien zum Ordner `.platform/nginx/conf.d/` im Quell-Bundle der Anwendung hinzu. Die nginx-Konfiguration von Elastic Beanstalk fügt `.conf`-Dateien automatisch zu diesem Ordner hinzu.

```
~/workspace/my-app/
|-- .platform
|   `-- nginx
|       `-- conf.d
|           `-- myconf.conf
`-- other source files
```

Konfigurationsdateien in `.platform/nginx/conf.d/` sind im `http` Block der Nginx-Konfiguration enthalten. Verwenden Sie diesen Speicherort für Konfigurationen, die global gelten.

Um die standardmäßige `server` Nginx-Blockkonfiguration zu erweitern, fügen Sie `.conf` Konfigurationsdateien zu einem Ordner hinzu, der `.platform/nginx/conf.d/`

`elasticbeanstalk/` in Ihrem Anwendungsquellpaket benannt ist. Die Elastic Beanstalk-Nginx-Konfiguration umfasst `.conf` Dateien in diesem Ordner innerhalb des Blocks. `server`

```
~/workspace/my-app/
|-- .platform
|   `-- nginx
|       `-- conf.d
|           `-- elasticbeanstalk
|               `-- server.conf
`-- other source files
```

Verwenden Sie diesen Speicherort, um serverspezifische Konfigurationen hinzuzufügen, z. B. zusätzliche Speicherortblöcke, benutzerdefinierte Fehlerseiten oder Direktiven auf Serverebene. Im folgenden Beispiel wird ein benutzerdefinierter Standortblock hinzugefügt.

Example. platform/nginx/conf.d/elasticbeanstalk/server.conf

```
location /test {
    return 200 "Hello World!";
    add_header Content-Type text/plain;
}
```

Um die nginx-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, fügen Sie bei `.platform/nginx/nginx.conf` eine Konfiguration in das Quellpaket ein.

```
~/workspace/my-app/
|-- .platform
|   `-- nginx
|       `-- nginx.conf
`-- other source files
```

Wenn Sie die nginx-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeile zu `nginx.conf` hinzu, damit die Elastic-Beanstalk-Konfigurationen für Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk, automatische Anwendungszuweisungen und statische Dateien einbezogen werden.

```
 include conf.d/elasticbeanstalk/*.conf;
```

Konfigurieren von Apache HTTPD

Die Plattformen Tomcat, Node.js, PHP und Python ermöglichen Ihnen die Auswahl des Apache-HTTPD-Proxy-Servers als Alternative zu nginx. Dies ist nicht die Standardoption. Im folgenden Beispiel wird Elastic Beanstalk für die Verwendung von Apache HTTPD konfiguriert.

Example.ebextensions/httpd-proxy.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

Sie können die Elastic Beanstalk-Apache-Standardkonfiguration mit Ihren zusätzlichen Konfigurationsdateien erweitern. Alternativ können Sie die Apache-Standardkonfiguration von Elastic Beanstalk vollständig überschreiben.

Um die Apache-Standardkonfiguration von Elastic Beanstalk zu erweitern, fügen Sie `.conf`-Konfigurationsdateien zum Ordner `.platform/httpd/conf.d` im Quell-Bundle der Anwendung hinzu. Die Apache-Konfiguration von Elastic Beanstalk fügt `.conf`-Dateien automatisch zu diesem Ordner hinzu.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd-proxy.config
|-- .platform
|   -- httpd
|      -- conf.d
|         -- port5000.conf
|         -- ssl.conf
-- index.jsp
```

Beispielsweise wird mit der folgenden Apache 2.4-Konfiguration ein Listener auf Port 5000 hinzugefügt:

Example. platform/httpd/conf.d/port5000.conf

```
listen 5000
<VirtualHost *:5000>
  <Proxy *>
    Require all granted
  </Proxy>
```

```
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-error_log
</VirtualHost>
```

Um die Apache-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, fügen Sie bei eine Konfiguration in das Quellpaket ei `.platform/httpd/conf/httpd.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd-proxy.config
|-- .platform
|   `-- httpd
|       `-- conf
|           `-- httpd.conf
`-- index.jsp
```

> **Note**
>
> Wenn Sie die Apache-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeilen zu `httpd.conf` hinzu, damit die Elastic-Beanstalk-Konfigurationen für Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk, automatische Anwendungszuweisungen und statische Dateien einbezogen werden.
>
> ```
> IncludeOptional conf.d/elasticbeanstalk/*.conf
> ```

Anwendungsbeispiel mit Erweiterungen

Im folgenden Beispiel wird ein Anwendungs-Quell-Bundle mit mehreren Erweiterungsfunktionen veranschaulicht, die von Amazon-Linux-2- und Amazon-Linux-2023-Plattformen für Elastic Beanstalk unterstützt werden: eine `Procfile`, `.ebextensions`-Konfigurationsdateien, benutzerdefinierte Hooks und Proxy-Konfigurationsdateien.

```
~/my-app/
|-- web.jar
|-- Procfile
```

```
|-- readme.md
|-- .ebextensions/
|   |-- options.config        # Option settings
|   `-- cloudwatch.config     # Other .ebextensions sections, for example files and
 container commands
`-- .platform/
    |-- nginx/                # Proxy configuration
    |   |-- nginx.conf
    |   `-- conf.d/
    |       |-- custom.conf
    |       `-- elasticbeanstalk/
    |           `-- server.conf
    |-- hooks/                # Application deployment hooks
    |   |-- prebuild/
    |   |   |-- 01_set_secrets.sh
    |   |   `-- 12_update_permissions.sh
    |   |-- predeploy/
    |   |   `-- 01_some_service_stop.sh
    |   `-- postdeploy/
    |       |-- 01_set_tmp_file_permissions.sh
    |       |-- 50_run_something_after_app_deployment.sh
    |       `-- 99_some_service_start.sh
    `-- confighooks/          # Configuration deployment hooks
        |-- prebuild/
        |   `-- 01_set_secrets.sh
        |-- predeploy/
        |   `-- 01_some_service_stop.sh
        `-- postdeploy/
            |-- 01_run_something_after_config_deployment.sh
            `-- 99_some_service_start.sh
```

> **Note**
>
> Einige dieser Erweiterungen werden auf Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2) nicht unterstützt.

Bereitstellen von.NET-Windows-Anwendungen mit Elastic Beanstalk

> **Schauen Sie sich das.NET im AWS Developer Center an**
>
> Haben Sie schon einmal in unserem .Net Developer Center vorbeigeschaut? Es ist unsere zentrale Anlaufstelle für alles, was mit .NET zu tun hat AWS.
> Weitere Informationen finden Sie unter.NET on AWS Developer Center.

Dieses Kapitel enthält Anweisungen zur Konfiguration und Bereitstellung Ihrer ASP.NET- und .NET Core-Windows-Webanwendungen für AWS Elastic Beanstalk. Elastic Beanstalk macht es einfach, Ihren.NET-Webanwendungen (Windows) mithilfe von Amazon Web Services bereitzustellen, zu verwalten und zu skalieren.

Sie können Ihre Anwendung in nur wenigen Minuten mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) oder der Elastic Beanstalk Beanstalk-Konsole bereitstellen. Nachdem Sie Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt haben, können Sie weiterhin die EB-CLI verwenden, um Ihre Anwendung und Umgebung zu verwalten, oder Sie können die Elastic Beanstalk Beanstalk-Konsole verwenden, AWS CLI oder die. APIs

Dieses Kapitel enthält die folgenden Tutorials:

- QuickStart für .NET Core unter Windows— Step-by-step Anweisungen zum Erstellen und Bereitstellen einer Hello World .NET Core-Windows-Anwendung mithilfe der EB CLI.
- QuickStart für ASP.NET— Step-by-step Anweisungen zum Erstellen und Bereitstellen einer Hello World-ASP.NET-Anwendung mit dem AWS Toolkit for Visual Studio.

Wenn Sie Hilfe bei der Entwicklung von Windows .NET Core-Anwendungen benötigen, können Sie sich an mehrere Stellen wenden:

- .NET-Entwicklungsforum — Stellen Sie Ihre Fragen und erhalten Sie Feedback.
- .NET Developer Center — Eine zentrale Anlaufstelle für Beispielcode, Dokumentation, Tools und zusätzliche Ressourcen.

- AWS SDK for .NET für.NET-Dokumentation — Lesen Sie mehr über das Einrichten des SDK und das Ausführen von Codebeispielen, Funktionen des SDK und detaillierte Informationen zu den API-Vorgängen für das SDK.

> **Note**
>
> Diese Plattform unterstützt keine Arbeitsumgebungen. Details hierzu finden Sie unter Elastic-Beanstalk-Worker-Umgebungen.

Themen

- QuickStart: Stellen Sie eine.NET-Core-Anwendung für Windows auf Elastic Beanstalk bereit
- QuickStart: Stellen Sie eine ASP.NET-Anwendung auf Elastic Beanstalk bereit
- Einrichten der .NET-Entwicklungsumgebung
- Verwendung der Elastic Beanstalk .NET Windows-Plattform
- Hinzufügen einer Amazon RDS-DB-Instance zur .NET-Anwendungsumgebung
- Die AWS Toolkit for Visual Studio
- Migrieren Ihrer lokalen .NET-Anwendung zu Elastic Beanstalk
- Empfehlungen für ausrangierte Windows Server-Komponenten auf Elastic Beanstalk

QuickStart: Stellen Sie eine.NET-Core-Anwendung für Windows auf Elastic Beanstalk bereit

QuickStart In diesem Tutorial erfahren Sie, wie Sie eine.NET Core-Anwendung für Windows erstellen und in einer AWS Elastic Beanstalk Umgebung bereitstellen.

> **Nicht für Produktionszwecke**
>
> Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine.NET Core-Anwendung für Windows
- Schritt 2: Führen Sie Ihre Anwendung lokal aus
- Schritt 3: Stellen Sie Ihre.NET Core on Windows-Anwendung mit der EB CLI bereit
- Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus
- Schritt 5: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center -Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden in Auflistungen angezeigt, denen ein Eingabeaufforderungssymbol (**>**) und gegebenenfalls der Name des aktuellen Verzeichnisses vorangestellt sind.

```
C:\eb-project> this is a command
this is output
```

EB-CLI

Dieses Tutorial verwendet die Befehlszeilenschnittstelle Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

.NET Core unter Windows

Wenn Sie das .NET SDK nicht auf Ihrem lokalen Computer installiert haben, können Sie es installieren, indem Sie auf der Website mit der .NET-Dokumentation auf den Link .NET herunterladen klicken.

Überprüfen Sie Ihre .NET SDK-Installation, indem Sie den folgenden Befehl ausführen.

```
C:\> dotnet --info
```

Schritt 1: Erstellen Sie eine.NET Core-Anwendung für Windows

Erstellen Sie ein Projektverzeichnis.

```
C:\> mkdir eb-dotnetcore
C:\> cd eb-dotnetcore
```

Erstellen Sie als Nächstes eine Hello RESTful World-Beispiel-Webdienstanwendung, indem Sie die folgenden Befehle ausführen.

```
C:\eb-dotnetcore> dotnet new web --name HelloElasticBeanstalk
C:\eb-dotnetcore> cd HelloElasticBeanstalk
```

Schritt 2: Führen Sie Ihre Anwendung lokal aus

Führen Sie den folgenden Befehl aus, um Ihre Anwendung lokal auszuführen.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> dotnet run
```

Die Ausgabe sollte etwa wie der folgende Text aussehen.

```
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7222
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5228
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
```

```
    Content root path: C:\Users\Administrator\eb-dotnetcore\HelloElasticBeanstalk
```

> **Note**
>
> Der `dotnet` Befehl wählt zufällig einen Port aus, wenn die Anwendung lokal ausgeführt wird. In diesem Beispiel ist der Port 5228. Wenn Sie die Anwendung in Ihrer Elastic Beanstalk Beanstalk-Umgebung bereitstellen, wird die Anwendung auf Port 5000 ausgeführt.

Geben Sie die URL-Adresse `http://localhost:`*`port`* in Ihren Webbrowser ein. Für dieses spezielle Beispiel lautet der Befehl`http://localhost:5228`. Der Webbrowser sollte „Hello World!“ anzeigen.

Schritt 3: Stellen Sie Ihre.NET Core on Windows-Anwendung mit der EB CLI bereit

Führen Sie die folgenden Befehle aus, um eine Elastic Beanstalk Beanstalk-Umgebung für diese Anwendung zu erstellen.

Um eine Umgebung zu erstellen und Ihre Anwendung für.NET Core unter Windows bereitzustellen

1. Führen Sie die folgenden Befehle im `HelloElasticBeanstalk` Verzeichnis aus, um Ihre Anwendung zu veröffentlichen und zu komprimieren.

   ```
   C:\eb-dotnetcore\HelloElasticBeasntalk> dotnet publish -o site
   C:\eb-dotnetcore\HelloElasticBeasntalk> cd site
   C:\eb-dotnetcore\HelloElasticBeasntalk\site> Compress-Archive -Path * -
   DestinationPath ../site.zip
   C:\eb-dotnetcore\HelloElasticBeasntalk\site> cd ..
   ```

2. Erstellen Sie im `HelloElasticBeanstalk` aufgerufenen Ordner eine neue Datei `aws-windows-deployment-manifest.json` mit dem folgenden Inhalt:

   ```
   {
       "manifestVersion": 1,
       "deployments": {
           "aspNetCoreWeb": [
           {
               "name": "test-dotnet-core",
   ```

```
            "parameters": {
                "appBundle": "site.zip",
                "iisPath": "/",
                "iisWebSite": "Default Web Site"
            }
        }
        ]
    }
}
```

3. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb init -p iis dotnet-windows-server-
tutorial --region us-east-2
```

Dieser Befehl erstellt eine Anwendung mit dem Namen `dotnet-windows-server-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der neuesten Version der Windows-Serverplattform erstellt werden.

4. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erstellt automatisch eine Zip-Datei für Ihre Anwendung und startet sie auf Port 5000.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb create dotnet-windows-server-env
```

Es dauert etwa fünf Minuten, bis Elastic Beanstalk Ihre Umgebung erstellt hat.

Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, öffnen Sie Ihre Website mit. eb open

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb open
```

Herzlichen Glückwunsch! Sie haben eine.NET-Core-Anwendung für Windows mit Elastic Beanstalk bereitgestellt! Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet.

Schritt 5: Bereinigen

Sie können Ihre Umgebung beenden, wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind. Elastic Beanstalk beendet alle AWS Ressourcen, die mit Ihrer Umgebung verknüpft sind.

Führen Sie den folgenden Befehl aus, um Ihre Elastic Beanstalk Beanstalk-Umgebung mit der EB-CLI zu beenden.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb terminate
```

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt. Es dient als einfache Beispielanwendung mit einer einzigen EC2-Instance, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 instance (EC2-Instance) – Eine virtuelle Amazon EC2-Maschine zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 zugelassen wird. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Web-App erreichen. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.

- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen. Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

> **Probieren Sie weitere Tutorials aus**
>
> Wenn Sie andere Tutorials mit anderen Beispielanwendungen ausprobieren möchten, finden Sie weitere Informationen unterQuickStart für ASP.NET.

Nachdem Sie ein oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, .NET Core lokal auf Windows-Anwendungen zu entwickeln und auszuführen, finden Sie weitere Informationen unter Einrichten der .NET-Entwicklungsumgebung

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

QuickStart: Stellen Sie eine ASP.NET-Anwendung auf Elastic Beanstalk bereit

QuickStart In diesem Tutorial erfahren Sie, wie Sie eine ASP.NET-Anwendung erstellen und in einer Umgebung bereitstellen. AWS Elastic Beanstalk

⚠ Nicht für Produktionszwecke

Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine ASP.NET-Anwendung
- Schritt 2: Führen Sie Ihre Anwendung lokal aus
- Schritt 3: Stellen Sie Ihre ASP.NET-Anwendung mit dem bereit AWS Toolkit for Visual Studio
- Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus
- Schritt 5: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center -Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Dieses QuickStart Tutorial führt Sie durch die Erstellung einer „Hello World“ -Anwendung und deren Bereitstellung in einer Elastic Beanstalk Beanstalk-Umgebung mit Visual Studio und dem. AWS Toolkit for Visual Studio

Visual Studio

Folgen Sie den Anweisungen auf der Visual Studio-Downloadseite, um Visual Studio herunterzuladen und zu installieren. In diesem Beispiel wird Visual Studio 2022 verwendet. Wählen Sie während der Visual Studio-Installation die folgenden spezifischen Elemente aus:

- Wählen Sie auf der Registerkarte Workloads die Option ASP.NET und Webentwicklung aus.
- Wählen Sie auf der Registerkarte Einzelne Komponenten die Entwicklungstools für .NET Framework 4.8 sowie Projekt- und Elementvorlagen für .NET Framework aus.

AWS Toolkit for Visual Studio

AWS Toolkit for Visual Studio Folgen Sie zum Herunterladen und Einrichten den Anweisungen im Abschnitt Erste Schritte des AWS Toolkit for Visual Studio Benutzerhandbuchs.

Schritt 1: Erstellen Sie eine ASP.NET-Anwendung

Als Nächstes erstellen Sie eine Anwendung, die Sie in einer Elastic Beanstalk Beanstalk-Umgebung bereitstellen. Wir erstellen eine ASP.NET-Webanwendung „Hello World“.

Um eine ASP.NET-Anwendung zu erstellen

1. Starten Sie Visual Studio. Wählen Sie im Menü Datei die Option Neu und dann Projekt aus.
2. Das Dialogfeld Neues Projekt erstellen wird angezeigt. Wählen Sie ASP.NET-Webanwendung (.NET Framework) und dann Weiter aus.
3. Geben **`eb-aspnet`**Sie im Dialogfeld „Neues Projekt konfigurieren“ Ihren Projektnamen ein. Wählen Sie im Dropdownmenü Framework die Option .NET Framework 4.8 und dann Create aus.

   Notieren Sie sich das Projektverzeichnis. In diesem Beispiel ist das Projektverzeichnis`C:\Users\Administrator\source\repos\eb-aspnet\eb-aspnet`.
4. Das Dialogfeld Neue ASP.NET-Webanwendung erstellen wird angezeigt. Wählen Sie die Vorlage Leer aus. Wählen Sie als Nächstes Erstellen aus.

   Zu diesem Zeitpunkt haben Sie mit Visual Studio ein leeres ASP.NET-Webanwendungsprojekt erstellt. Als Nächstes erstellen wir ein Webformular, das als Einstiegspunkt für die ASP.NET-Webanwendung dient.
5. Wählen Sie im Menü Projekt die Option Neues Element hinzufügen aus. Wählen Sie auf der Seite Neues Element hinzufügen die Option Webformular aus und geben Sie dem Objekt einen Namen`Default.aspx`. Wählen Sie als Nächstes Hinzufügen aus.
6. Fügen Sie Folgendes hinzu `Default.aspx:`

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
 Inherits="eb_aspnet.Default" %>

<!DOCTYPE html>

<html xmlns="https://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Hello Elastic Beanstalk!</title>
</head>
<body>
    <form id="body" runat="server">
        <div>
            Hello Elastic Beanstalk! This is an ASP.NET on Windows Server
 application.
        </div>
    </form>
</body>
</html>
```

Schritt 2: Führen Sie Ihre Anwendung lokal aus

Wählen Sie in Visual Studio im Menü Debug die Option Debugging starten aus, um Ihre Anwendung lokal auszuführen. Auf der Seite sollte „Hello Elastic Beanstalk! Dies ist eine ASP.NET-Anwendung auf Windows Server.“

Schritt 3: Stellen Sie Ihre ASP.NET-Anwendung mit dem bereit AWS Toolkit for Visual Studio

Gehen Sie wie folgt vor, um eine Elastic Beanstalk Beanstalk-Umgebung zu erstellen und Ihre neue Anwendung darin bereitzustellen.

Um eine Umgebung zu erstellen und Ihre ASP.NET-Anwendung bereitzustellen

1. Klicken Sie im Solution Explorer mit der rechten Maustaste auf Ihre Anwendung und wählen Sie dann Veröffentlichen in AWS Elastic Beanstalk aus.
2. Wählen Sie einen Namen für Ihre neue Elastic Beanstalk Beanstalk-Anwendung und Umgebung.
3. Danach können Sie mit den von Elastic Beanstalk bereitgestellten Standardeinstellungen fortfahren oder die Optionen und Einstellungen nach Ihren Wünschen ändern.

4. Wählen Sie auf der Seite „Überprüfen“ die Option Bereitstellen aus. Dadurch wird Ihre ASP.NET-Webanwendung gepackt und auf Elastic Beanstalk bereitgestellt.

   Es dauert etwa fünf Minuten, bis Elastic Beanstalk Ihre Umgebung erstellt hat. Mit der Bereitstellungsfunktion von Elastic Beanstalk wird die Umgebung überwacht, bis sie mit dem neu bereitgestellten Code verfügbar ist. Auf der Registerkarte Env: **< `environment name` >** sehen Sie den Status Ihrer Umgebung.

Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, werden auf der Registerkarte Env: **< `environment name` >** Informationen zu Ihrer Umgebung und Anwendung angezeigt, einschließlich der Domain-URL zum Starten Ihrer Anwendung. Wählen Sie diese URL auf dieser Registerkarte aus oder kopieren Sie sie und fügen Sie sie in Ihren Webbrowser ein.

Herzlichen Glückwunsch! Sie haben eine ASP.NET-Anwendung mit Elastic Beanstalk bereitgestellt!

Schritt 5: Bereinigen

Wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind, können Sie Ihre Umgebung im beenden. AWS Toolkit for Visual Studio

So beenden Sie Ihre Umgebung

1. Erweitern Sie den Elastic-Beanstalk-Knoten und den Anwendungsknoten in AWS Explorer. Klicken Sie mit der rechten Maustaste auf die Anwendungsumgebung und wählen Sie Terminate Environment (Umgebung beenden) aus.
2. Wenn Sie dazu aufgefordert werden, wählen Sie Ja, um zu bestätigen, dass Sie die Umgebung beenden möchten. Es wird einige Minuten dauern, bis Elastic Beanstalk die in der Umgebung laufenden AWS Ressourcen beendet hat.

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt. Es dient als einfache Beispielanwendung mit einer einzigen EC2-Instance, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 instance (EC2-Instance) – Eine virtuelle Amazon EC2-Maschine zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 zugelassen wird. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Web-App erreichen. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen. Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

> **Probieren Sie weitere Tutorials aus**
>
> Wenn Sie andere Tutorials mit anderen Beispielanwendungen ausprobieren möchten, finden Sie weitere Informationen unterQuickStart für .NET Core unter Windows.

Nachdem Sie eine oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, ASP.NET-Anwendungen lokal zu entwickeln und auszuführen, finden Sie weitere Informationen unter Einrichten der .NET-Entwicklungsumgebung

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

Einrichten der .NET-Entwicklungsumgebung

Dieses Thema enthält Anweisungen zum Einrichten einer.NET-Windows-Entwicklungsumgebung, in der Sie Ihre Anwendung lokal testen können, bevor Sie sie bereitstellen AWS Elastic Beanstalk. Außerdem wird auf Websites verwiesen, die Installationsanweisungen für nützliche Tools bereitstellen.

Sections

- Installieren einer IDE
- Installation des AWS Toolkit for Visual Studio

Wenn Sie AWS Ressourcen von Ihrer Anwendung aus verwalten müssen, installieren Sie das AWS SDK for .NET. Beispielsweise können Sie mit Amazon S3 Daten speichern und abrufen.

Mit dem AWS SDK for .NET können Sie in wenigen Minuten mit einem einzigen herunterladbaren Paket mit Visual Studio-Projektvorlagen, der AWS .NET-Bibliothek, C#-Codebeispielen und Dokumentation loslegen. Praktische Beispiele dazu, wie mit den Bibliotheken Anwendungen erstellt werden können, sind in C# enthalten. Online-Videotutorials und Referenzdokumentationen werden bereitgestellt, um Sie dabei zu unterstützen, die Verwendung der Bibliotheken und Beispielcodes zu erlernen.

Weitere Informationen und Installationsanleitungen finden Sie auf der AWS SDK for .NET-Homepage.

Installieren einer IDE

Integrierte Entwicklungsumgebungen (IDEs) bieten eine Vielzahl von Funktionen, die die Anwendungsentwicklung erleichtern. Wenn Sie noch nie eine IDE für die .NET-Entwicklung verwendet haben, können Sie mit Visual Studio Community erste Schritte unternehmen.

Besuchen Sie die Visual Studio Community-Webseite, um Visual Studio Community herunterzuladen und zu installieren.

Installation des AWS Toolkit for Visual Studio

Das AWS Toolkit for Visual Studioist ein Open-Source-Plug-In für die Visual Studio-IDE, das Entwicklern das Entwickeln, Debuggen und Bereitstellen von.NET-Anwendungen erleichtert. AWS Installationsanleitungen finden Sie auf der Toolkit for Visual Studio-Homepage.

Verwendung der Elastic Beanstalk .NET Windows-Plattform

In diesem Thema wird beschrieben, wie Sie Ihre ASP.NET- und .NET Core-Windows-Webanwendungen auf Elastic Beanstalk konfigurieren, erstellen und ausführen.

AWS Elastic Beanstalk unterstützt eine Reihe von Plattformen für verschiedene Versionen des.NET-Programmierframeworks und Windows Server. Eine komplette Liste finden Sie unter .NET on Windows Server mit IIS im Dokument AWS Elastic Beanstalk -Plattformen.

Elastic Beanstalk bietet Konfigurationsoptionen, mit denen Sie die Software anpassen können, die auf den EC2 Instances in Ihrer Elastic Beanstalk Beanstalk-Umgebung ausgeführt wird. Sie können die für Ihre Anwendung erforderlichen Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren und .NET-Framework-Einstellungen festlegen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das Ändern der Konfiguration einer ausgeführten Umgebung verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie gespeicherte Konfigurationen verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie Konfigurationsdateien einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung

erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe Konfigurationsoptionen.

Konfigurieren Ihrer .NET-Umgebung in der Elastic Beanstalk-Konsole

Mit der Elastic Beanstalk-Konsole können Sie die Protokollrotation für Amazon S3 aktivieren, Variablen konfigurieren, die die Anwendung in der Umgebung lesen kann, und die .NET-Framework-Einstellungen ändern.

So konfigurieren Sie Ihre .NET-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

Container-Optionen

- Target .NET runtime (.NET-Ziellaufzeit) – Legen Sie den Wert `2.0` fest, um CLR v2 auszuführen.
- Enable 32-bit applications (32-Bit-Anwendungen aktivieren) – Legen Sie den Wert `True` fest, um 32-Bit-Anwendungen auszuführen.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.

- Rotation der Protokolldateien nach Amazon S3 aktivieren — Gibt an, ob Protokolldateien für die EC2 Amazon-Instances Ihrer Anwendung in den Amazon S3-Bucket kopiert werden, der Ihrer Anwendung zugeordnet ist.

Umgebungseigenschaften

Im Abschnitt Umgebungseigenschaften können Sie die Umgebungskonfigurationseinstellungen für die EC2 Amazon-Instances angeben, auf denen Ihre Anwendung ausgeführt wird. Diese Einstellungen werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet. Verwenden Sie `System.GetEnvironmentVariable`, um diese zu lesen. Identische Schlüssel können in `web.config` und als Umgebungseigenschaften vorhanden sein. Verwenden Sie den `System.Configuration`-Namespace zum Lesen von Werten aus `web.config`.

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;
string endpoint = appConfig["API_ENDPOINT"];
```

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Der Namespace aws:elasticbeanstalk:container:dotnet:apppool

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Über die .NET-Plattform werden Optionen im Namespace `aws:elasticbeanstalk:container:dotnet:apppool` festgelegt, mithilfe derer Sie die .NET-Laufzeitumgebung konfigurieren können.

In der folgenden Beispielkonfigurationsdatei werden die Einstellungen der einzelnen Optionen angezeigt, die in diesem Namespace verfügbar sind:

Example.ebextensions/dotnet-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:dotnet:apppool:
    Target Runtime: 2.0
    Enable 32-bit Applications: True
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Migration über die Hauptversionen der Elastic Beanstalk-Windows Server-Plattform hinweg

AWS Elastic Beanstalk hatte mehrere Hauptversionen seiner Windows Server-Plattform. Diese Seite geht auf die wichtigsten Verbesserungen für jede Hauptversion und darauf ein, was Sie bedenken müssen, bevor Sie auf eine neuere Version migrieren.

Die Windows Server-Plattform liegt derzeit in Version 2 (v2) vor. Wenn Ihre Anwendung eine Version der Windows Server-Plattform vor v2 verwendet, empfehlen wir die Migration auf v2.

Neues in den Hauptversionen der Windows Server-Plattform

Windows Server-Plattform V2

Version 2 (v2) der Elastic Beanstalk Windows Server Plattform wurde im Februar 2019 veröffentlicht. Mit V2 ist das Verhalten der Windows Server-Plattform in mehreren wichtigen Punkten demjenigen der Linux-basierten Plattformen von Elastic Beanstalk ähnlicher. V2 ist vollständig abwärtskompatibel zu v1, wodurch die Migration von v1 ganz einfach ist.

Die Windows Server-Plattform unterstützt jetzt Folgendes:

- Versioning – Jede Version erhält eine neue Versionsnummer, und Sie können auf frühere Versionen (die noch verfügbar sind) verweisen, wenn Sie Umgebungen erstellen und verwalten.
- Erweiterter Zustandsbericht – Weitere Informationen finden Sie unter Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk.
- Bereitstellungen, die unveränderlich und fortlaufend mit zusätzlichem Batch sind – Weitere Informationen zu Bereitstellungsrichtlinien finden Sie unter Anwendungen in Elastic Beanstalk-Umgebungen bereitstellen.
- Unveränderliche Updates – Weitere Informationen zu den Update-Typen finden Sie unter Konfigurationsänderungen.
- Verwaltete Plattformaktualisierungen – Weitere Informationen finden Sie unter Verwaltete Plattformaktualisierungen.

> **Note**
>
> Die neuen Bereitstellungs- und Update-Funktionen sind vom erweiterten Zustandsbericht abhängig. Um sie verwenden zu können, müssen Sie den erweiterten Zustandsbericht aktivieren. Details hierzu finden Sie unter Aktivieren der erweiterten Elastic-Beanstalk-Integritätsberichte.

Windows Server-Plattform V1

Version 1.0.0 (v1) der Elastic Beanstalk-Windows Server-Plattform wurde im Oktober 2015 veröffentlicht. In dieser Version ändert sich die Reihenfolge, in der die Elastic Beanstalk-Prozesse Befehle in Konfigurationsdateien bei der Erstellung und Aktualisierungen der Umgebung verarbeitet werden.

Vorherige Plattformversionen haben keine Versionsnummer im Lösungs-Stack-Namen:

- Windows Server 2012 R2 mit 64 Bit und IIS 8.5
- Windows Server Core 2012 R2 mit 64 Bit und IIS 8.5
- Windows Server 2012 mit 64 Bit und IIS 8
- Windows Server 2008 R2 mit 64 Bit und IIS 7.5

In früheren Versionen war die Verarbeitungsreihenfolge für Konfigurationsdateien inkonsistent. Während der Umgebungserstellung wurde `Container Commands` ausgeführt, nachdem die Anwendungsquelle für IIS bereitgestellt wurde. Bei einer Bereitstellung in einer ausgeführten Umgebung werden Container-Befehle ausgeführt, bevor die neue Version bereitgestellt wird. Bei einer Skalierung nach oben werden Konfigurationsdateien nicht verarbeitet.

Darüber hinaus wird IIS gestartet, bevor Container-Befehle ausgeführt werden. Dieses Verhalten hat einige Kunden zur Implementierung von Umgehungslösungen in Container-Befehlen veranlasst, wobei der IIS-Server angehalten wird, bevor Befehle ausgeführt werden, und nach Abschluss wieder gestartet wird.

Version 1 behebt die Inkonsistenzen. Damit ist das Verhalten der Windows Server-Plattform demjenigen der Linux-basierten Plattformen von Elastic Beanstalk ähnlicher. Bei der v1-Plattform führt Elastic Beanstalk immer Container-Befehle vor dem Start des IIS-Servers aus.

Die Lösungs-Stacks der v1-Plattform haben ein `v1` nach der Windows Server-Version:

- Windows Server 2012 R2 v1.1.0 mit 64 Bit und IIS 8.5
- Windows Server Core 2012 R2 v1.1.0 mit 64 Bit und IIS 8.5
- Windows Server 2012 v1.1.0 mit 64 Bit und IIS 8
- Windows Server 2008 R2 v1.1.0 mit 64 Bit und IIS 7.5

Außerdem extrahiert die v1-Plattform den Inhalt Ihres Anwendungs-Quell-Bundles in `C:\staging\`, bevor Container-Befehle ausgeführt werden. Nachdem die Container-Befehle abgeschlossen wurden, wird der Inhalt dieses Ordners als ZIP-Datei komprimiert und für IIS bereitgestellt. In diesem Workflow können Sie den Inhalt Ihres Anwendungs-Quell-Bundles mit Befehlen oder einem Skript vor der Bereitstellung ändern.

Migration von früheren Hauptversionen der Windows Server-Plattform

Lesen Sie diesen Abschnitt mit Hinweisen zur Migration, bevor Sie Ihre Umgebung aktualisieren. Weitere Informationen zur Aktualisierung Ihrer Umgebungsplattform auf eine neuere Version finden Sie unter Aktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung.

Kopieren von V1 in V2

.NET Core 1.x und 2.0 werden von der v2-Plattform von Windows Server nicht unterstützt. Wenn Sie Ihre Anwendung von Windows Server v1 auf v2 migrieren und Ihre Anwendung eine dieser .NET Core-Versionen nutzt, aktualisieren Sie Ihre Anwendung auf eine .NET-Core Version, die von v2 unterstützt wird. Eine Liste der unterstützten Versionen finden Sie unter .NET mit Windows Server und IIS in den AWS Elastic Beanstalk Plattformen.

Wenn Ihre Anwendung ein benutzerdefiniertes Amazon Machine Image (AMI) verwendet, erstellen Sie ein neues benutzerdefiniertes AMI basierend auf einer Windows Server-Plattform v2 AMI. Weitere Informationen hierzu finden Sie unter Verwenden eines benutzerdefinierten Amazon Machine Images (AMI) in Ihrer Elastic Beanstalk Beanstalk-Umgebung.

> **Note**
>
> Welche Bereitstellungs- und Update-Funktionen in Windows Server v2 neu sind, ist vom erweiterten Zustandsbericht abhängig. Wenn Sie eine Umgebung auf v2 migrieren, ist der erweiterte Zustandsbericht deaktiviert. Sie müssen sie aktivieren, um diese Funktionen verwenden zu können. Details hierzu finden Sie unter Aktivieren der erweiterten Elastic-Beanstalk-Integritätsberichte.

Von Pre-V1-Versionen

Wenn Sie Ihre Anwendung von einem Windows Server-Lösungs-Stack vor v1 migrieren und derzeit Containerbefehle verwenden, müssen Sie zusätzlich zu den Überlegungen, die auf eine Migration von v1 zutreffen, alle Befehle entfernen, die Sie hinzugefügt haben, um die Verarbeitungsinkonsistenzen beim Migrieren auf eine neuere Version zu umgehen. Ab v1 werden Container-Befehle garantiert vollständig ausgeführt, bevor die Anwendungsquelle bereitgestellt und bevor IIS gestartet wird. Auf diese Weise können Sie Änderungen an der Quelle in `C:\staging` durchführen und IIS-Konfigurationsdateien in diesem Schritt ohne Probleme ändern.

Sie können beispielsweise die verwenden, AWS CLI um eine DLL-Datei von Amazon S3 in Ihre Anwendungsquelle herunterzuladen:

`.ebextensions\copy-dll.config`

```
container_commands:
  copy-dll:
    command: aws s3 cp s3://amzn-s3-demo-bucket/dlls/large-dll.dll .\lib\
```

Weitere Informationen zur Verwendung von Konfigurationsdateien finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

Ausführen mehrerer Anwendungen und ASP.NET Core-Anwendungen mit einem Bereitstellungsmanifest

Sie können mit einem Bereitstellungsmanifest Elastic Beanstalk anweisen, wie Ihre Anwendung bereitgestellt werden soll. Wenn Sie diese Methode verwenden, müssen Sie kein `MSDeploy` für eine einzelne ASP.NET-Anwendung generieren, die im Root-Pfad Ihrer Website ausgeführt wird. Stattdessen können Sie eine Manifestdatei verwenden, um mehrere Anwendungen auf verschiedenen Pfaden auszuführen. Alternativ können Sie Elastic Beanstalk anweisen, die App mit ASP.NET Core bereitzustellen und auszuführen. Sie können ein Bereitstellungsmanifest auch zum Konfigurieren eines Anwendungspools verwenden, in dem Sie Ihre Anwendungen ausführen.

Bereitstellungsmanifeste fügen Unterstützung für .NET Core-Anwendungen zu Elastic Beanstalk hinzu. Sie können eine .NET Framework-Anwendung ohne Bereitstellungsmanifest bereitstellen. .NET-Core-Anwendungen benötigen jedoch ein Bereitstellungsmanifest, um auf Elastic Beanstalk ausgeführt zu werden. Wenn Sie ein Bereitstellungsmanifest verwenden, können Sie ein Website-Archiv für jede erstellen und die Website-Archive dann in einem zweiten ZIP-Archiv mit dem Bereitstellungsmanifest bündeln.

Bereitstellungsmanifeste bieten auch die Möglichkeit, mehrere Anwendungen auf verschiedenen Pfaden auszuführen. Ein Bereitstellungsmanifest definiert eine Reihe von Bereitstellungszielen, jeweils mit einem Website-Archiv und einem Pfad, auf dem IIS es ausführen soll. Sie können beispielsweise einen Web-API auf dem `/api`-Pfad ausführen, um asynchrone Anfragen zu verarbeiten, und eine Web-App auf dem Stammpfad ausführen, die die API in Anspruch nimmt.

Sie können ein Bereitstellungsmanifest verwenden, um IIS-Websites mit benutzerdefinierten Bindungen und physischen Pfaden zu konfigurieren. Auf diese Weise können Sie Websites einrichten, die bestimmte Ports oder Hostnamen abhören, bevor Sie Ihre Anwendungen bereitstellen.

Sie können ein Bereitstellungsmanifest auch verwenden, um mehrere Anwendungen mit Anwendungspools in IIS oder Kestrel auszuführen. Sie können einen Anwendungspool so konfigurieren, dass Ihre Anwendungen in regelmäßigen Abständen neu gestartet werden, 32-Bit-Anwendungen ausführen oder eine bestimmte Version der .NET Framework-Laufzeit verwenden.

Für eine vollständige Anpassung können Sie Ihre eigenen Deployment-Skripts in Windows schreiben PowerShell und Elastic Beanstalk mitteilen, welche Skripts ausgeführt werden müssen, um Ihre Anwendung zu installieren, zu deinstallieren und neu zu starten.

Bereitstellungsmanifeste und verwandte Funktionen erfordern als Windows Server-Plattformkonfiguration Version 1.2.0 oder höher.

Detaillierte Informationen zu allen verfügbaren Konfigurationsoptionen, Eigenschaften und erweiterten Funktionen wie dem Überspringen von IIS-Resets finden Sie in der Schemareferenz für das Deployment-Manifest.

Sections

- .NET Core-Apps
- Ausführen mehrerer Anwendungen
- Konfigurieren Sie IIS-Websites
- Verwenden von Application Request Routing (ARR)
- Konfigurieren der Anwendungspools
- Definieren von benutzerdefinierten Bereitstellungen
- Schemareferenz für das Bereitstellungsmanifest

.NET Core-Apps

Sie können ein Bereitstellungsmanifest verwenden, um .NET-Core-Anwendungen auf Elastic Beanstalk auszuführen. .NET-Core-Version ist eine plattformübergreifende Version von .NET, die ein Befehlszeilentool (`dotnet`) enthält. Sie können damit eine Anwendung generieren, lokal ausführen und für die Veröffentlichung vorbereiten.

Zum Ausführen einer .NET-Core-Anwendung auf Elastic Beanstalk können Sie `dotnet publish` ausführen und die Ausgabe in ein ZIP-Archiv (ohne enthaltene Verzeichnisse) packen. Platzieren Sie das Website-Archiv in einem Quell-Bundle mit einem Bereitstellungsmanifest mit einem Bereitstellungsziel des Typs `aspNetCoreWeb`.

Das folgende Bereitstellungsmanifest führt eine .NET Core-Anwendung von einem Website-Archiv mit dem Namen `dotnet-core-app.zip` auf dem Stammpfad aus.

Example aws-windows-deployment-manifest.json — .NET-Kern

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "archive": "dotnet-core-app.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

Bündeln Sie das Manifest und Website-Archiv in einem ZIP-Archiv zum Erstellen eines Quell-Bundle.

Example dotnet-core-bundle.zip

```
.
|-- aws-windows-deployment-manifest.json
`-- dotnet-core-app.zip
```

Die Website-Archiv enthält den kompilierten Anwendungscode, Abhängigkeiten und `web.config`-Dateien.

Example dotnet-core-app.zip

```
.
|-- Microsoft.AspNetCore.Hosting.Abstractions.dll
|-- Microsoft.AspNetCore.Hosting.Server.Abstractions.dll
|-- Microsoft.AspNetCore.Hosting.dll
|-- Microsoft.AspNetCore.Http.Abstractions.dll
|-- Microsoft.AspNetCore.Http.Extensions.dll
|-- Microsoft.AspNetCore.Http.Features.dll
|-- Microsoft.AspNetCore.Http.dll
|-- Microsoft.AspNetCore.HttpOverrides.dll
|-- Microsoft.AspNetCore.Server.IISIntegration.dll
|-- Microsoft.AspNetCore.Server.Kestrel.dll
|-- Microsoft.AspNetCore.WebUtilities.dll
|-- Microsoft.Extensions.Configuration.Abstractions.dll
|-- Microsoft.Extensions.Configuration.EnvironmentVariables.dll
|-- Microsoft.Extensions.Configuration.dll
|-- Microsoft.Extensions.DependencyInjection.Abstractions.dll
|-- Microsoft.Extensions.DependencyInjection.dll
|-- Microsoft.Extensions.FileProviders.Abstractions.dll
|-- Microsoft.Extensions.FileProviders.Physical.dll
|-- Microsoft.Extensions.FileSystemGlobbing.dll
|-- Microsoft.Extensions.Logging.Abstractions.dll
|-- Microsoft.Extensions.Logging.dll
|-- Microsoft.Extensions.ObjectPool.dll
|-- Microsoft.Extensions.Options.dll
|-- Microsoft.Extensions.PlatformAbstractions.dll
|-- Microsoft.Extensions.Primitives.dll
|-- Microsoft.Net.Http.Headers.dll
|-- System.Diagnostics.Contracts.dll
|-- System.Net.WebSockets.dll
|-- System.Text.Encodings.Web.dll
|-- dotnet-core-app.deps.json
|-- dotnet-core-app.dll
|-- dotnet-core-app.pdb
|-- dotnet-core-app.runtimeconfig.json
`-- web.config
```

Ausführen mehrerer Anwendungen

Sie können mehrere Anwendungen mit einem Bereitstellungsmanifest ausführen, indem Sie mehrere Bereitstellungsziele definieren.

Das folgende Bereitstellungsmanifest konfiguriert zwei .NET Core-Anwendungen. Die `WebApiSampleApp` Anwendung implementiert eine einfache Web-API und bedient asynchrone Anfragen im `/api` Pfad. Die `DotNetSampleApp`-Anwendung ist eine Webanwendung, die Anfragen im Root-Pfad bedient.

Example aws-windows-deployment-manifest.json — mehrere Apps

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "WebAPISample",
        "parameters": {
          "appBundle": "WebApiSampleApp.zip",
          "iisPath": "/api"
        }
      },
      {
        "name": "DotNetSample",
        "parameters": {
          "appBundle": "DotNetSampleApp.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

Eine Beispielanwendung mit mehreren Anwendungen finden Sie hier:

- Bereitstellbares Quellpaket — -v2.zip dotnet-multiapp-sample-bundle
- Quellcode - -v2.zip dotnet-multiapp-sample-source

Konfigurieren Sie IIS-Websites

Mithilfe des Bereitstellungsmanifests können Sie IIS-Websites mit benutzerdefinierten Bindungen und physischen Pfaden konfigurieren. Dies ist nützlich, wenn Sie Websites einrichten müssen, die bestimmte Ports abhören, benutzerdefinierte Hostnamen verwenden oder Inhalte aus bestimmten Verzeichnissen bereitstellen.

Das folgende Bereitstellungsmanifest konfiguriert eine benutzerdefinierte IIS-Website, die HTTP mit einer bestimmten Portnummer und einem benutzerdefinierten physischen Pfad abhört:

Example aws-windows-deployment-manifest.json — Konfiguration der IIS-Website

```
{
  "manifestVersion": 1,
  "iisConfig": {
    "websites": [
      {
        "name": "MyCustomSite",
        "physicalPath": "C:\inetpub\wwwroot\mysite",
        "bindings": [
          {
            "protocol": "http",
            "port": 8080,
            "hostName": "mysite.local"
          }
        ]
      }
    ]
  },
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "appBundle": "dotnet-core-app.zip",
          "iisWebSite": "MyCustomSite",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

In diesem Beispiel:

- Eine Website mit dem Namen MyCustomSite "" wird mit einem benutzerdefinierten physischen Pfad erstellt
- Die Website hat eine HTTP-Bindung an Port 8080 mit einem bestimmten Hostnamen

- Die ASP.NET Core-Anwendung wird mithilfe des Parameters auf dieser benutzerdefinierten Website bereitgestellt `iisWebSite`

Verwenden von Application Request Routing (ARR)

Die Module Application Request Routing (ARR) und URL Rewrite sind vorinstalliert und in Elastic Beanstalk Windows verfügbar. AMIs Diese Module ermöglichen erweiterte Routing-Szenarien und URL-Manipulation über die IIS-Konfiguration mithilfe von EExtensions oder Anwendungskonfiguration.

Das folgende Beispiel zeigt ein einfaches Bereitstellungsmanifest, das eine Website mit einem benutzerdefinierten Port konfiguriert, kombiniert mit einer EExtensions-Konfiguration, die das grundlegende ARR-Routing einrichtet:

Example aws-windows-deployment-manifest.json — Einfache ARR-Einrichtung

```
{
  "manifestVersion": 1,
  "iisConfig": {
    "websites": [
      {
        "name": "ARRSite",
        "physicalPath": "C:\\inetpub\\wwwroot\\arrsite",
        "bindings": [
          {
            "protocol": "http",
            "port": 8080,
            "hostName": "localhost"
          }
        ]
      }
    ]
  },
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "BackendApp",
        "parameters": {
          "appBundle": "backend-app.zip",
          "iisWebSite": "ARRSite",
          "iisPath": "/backend"
        }
```

```
      }
    ]
  }
}
```

Die ARR-Konfiguration erfolgt über EExtensions. Die folgende Konfiguration richtet grundlegende ARR-Routing-Regeln ein:

Example.ebextensions/arr-config.config — Grundlegende ARR-Konfiguration

```
files:
  "C:\\temp\\configure-arr.ps1":
    content: |
      # Enable ARR proxy at server level
      Set-WebConfigurationProperty -PSPath 'MACHINE/WEBROOT/APPHOST' -Filter
 'system.webServer/proxy' -Name 'enabled' -Value 'True'

      # Clear any existing global rules to avoid conflicts
      Clear-WebConfiguration -PSPath 'MACHINE/WEBROOT/APPHOST' -Filter
 'system.webServer/rewrite/globalRules'

      # Add global rule to route all requests to backend
      Add-WebConfigurationProperty -PSPath 'MACHINE/WEBROOT/APPHOST' `
        -Filter 'system.webServer/rewrite/globalRules' `
        -Name '.' `
        -Value @{
          name = 'Route_to_Backend'
          stopProcessing = 'True'
          match = @{ url = '^(?!backend/)(.*)' }
          action = @{
            type = 'Rewrite'
            url = 'http://localhost:8080/backend/{R:1}'
          }
        }

container_commands:
  01_configure_arr:
    command: powershell -ExecutionPolicy Bypass -File "C:\\temp\\configure-arr.ps1"
    waitAfterCompletion: 0
```

Diese Konfiguration erstellt eine Website auf Port 8080 und richtet ARR so ein, dass alle eingehenden Anfragen an die Backend-Anwendung weitergeleitet werden, die auf dieser Site ausgeführt wird.

Konfigurieren der Anwendungspools

Sie können mehrere Anwendungen in Ihrer Windows-Umgebung unterstützen. Es stehen zwei Ansätze zur Verfügung:

- Sie können das out-of-process Hosting-Modell mit dem Kestrel-Webserver verwenden. Bei diesem Modell konfigurieren Sie mehrere Anwendungen für die Ausführung in einem Anwendungspool.
- Sie können das In-Process-Hosting-Modell verwenden. Bei diesem Modell verwenden Sie mehrere Anwendungspools, um mehrere Anwendungen mit nur einer Anwendung in jedem Pool auszuführen. Wenn Sie den IIS-Server verwenden und mehrere Anwendungen ausführen müssen, müssen Sie diesen Ansatz verwenden.

Um Kestrel für die Ausführung mehrerer Anwendungen in einem Anwendungspool zu konfigurieren, fügen Sie `hostingModel="OutofProcess"` zur Datei `web.config` hinzu. Betrachten Sie die folgenden Beispiele:

Example web.config — für das Kestrel-Hosting-Modell out-of-process

```
<configuration>
<location path="." inheritInChildApplications="false">
<system.webServer>
<handlers>
<add
    name="aspNetCore"
    path="*" verb="*"
    modules="AspNetCoreModuleV2"
    resourceType="Unspecified" />
</handlers>
<aspNetCore
    processPath="dotnet"
    arguments=".\CoreWebApp-5-0.dll"
    stdoutLogEnabled="false"
    stdoutLogFile=".\logs\stdout"
    hostingModel="OutofProcess" />
</system.webServer>
</location>
</configuration>
```

Example aws-windows-deployment-manifest.json — mehrere Anwendungen

```
{
```

```
"manifestVersion": 1,
  "deployments": {"msDeploy": [
      {"name": "Web-app1",
        "parameters": {"archive": "site1.zip",
          "iisPath": "/"
        }
      },
      {"name": "Web-app2",
        "parameters": {"archive": "site2.zip",
          "iisPath": "/app2"
        }
      }
    ]
  }
}
```

IIS unterstützt nicht mehrere Anwendungen in einem Anwendungspool, da es das In-Process-Hosting-Modell verwendet. Daher müssen Sie mehrere Anwendungen konfigurieren, indem Sie jede Anwendung einem Anwendungspool zuweisen. Mit anderen Worten: Weisen Sie einem Anwendungspool nur eine Anwendung zu.

Sie können IIS so konfigurieren, dass verschiedene Anwendungspools in der Datei `aws-windows-deployment-manifest.json` verwendet werden. Nehmen Sie die folgenden Aktualisierungen vor, wenn Sie auf die nächste Beispieldatei verweisen:

- Fügen Sie einen Abschnitt `iisConfig` hinzu, der einen Unterabschnitt mit dem Namen `appPools` enthält.
- Listen Sie im Block `appPools` die Anwendungspools auf.
- Definieren Sie im Abschnitt `deployments` einen Abschnitt `parameters` für jede Anwendung.
- Für jede Anwendung gibt der Abschnitt `parameters` ein Archiv, einen Pfad zum Ausführen und einen `appPool` an, in dem die Ausführung erfolgt.

Das folgende Bereitstellungsmanifest konfiguriert zwei Anwendungspools, die ihre Anwendung alle 10 Minuten neu starten. Sie hängen ihre Anwendungen auch an eine .NET Framework-Webanwendung an, die unter dem angegebenen Pfad ausgeführt wird.

Example aws-windows-deployment-manifest.json — eine Anwendung pro Anwendungspool

```
{
```

```
"manifestVersion": 1,
  "iisConfig": {"appPools": [
      {"name": "MyFirstPool",
       "recycling": {"regularTimeInterval": 10}
      },
      {"name": "MySecondPool",
       "recycling": {"regularTimeInterval": 10}
      }
     ]
    },
  "deployments": {"msDeploy": [
      {"name": "Web-app1",
        "parameters": {
           "archive": "site1.zip",
           "iisPath": "/",
           "appPool": "MyFirstPool"
           }
      },
      {"name": "Web-app2",
        "parameters": {
           "archive": "site2.zip",
           "iisPath": "/app2",
           "appPool": "MySecondPool"
         }
      }
     ]
    }
}
```

Definieren von benutzerdefinierten Bereitstellungen

Für noch mehr Kontrolle können Sie eine Anwendungsbereitstellung komplett anpassen, indem Sie eine benutzerdefinierte Bereitstellung definieren.

Dieses Deployment-Manifest weist Elastic Beanstalk an, PowerShell Skripts im 32-Bit-Modus auszuführen. Es spezifiziert drei Skripte: ein `install` Skript (`siteInstall.ps1`), das beim Start und bei der Bereitstellung von Instances ausgeführt wird, ein `uninstall` Skript (`siteUninstall.ps1`), das vor der Installation neuer Versionen während der Bereitstellung ausgeführt wird, und ein `restart` Skript (`siteRestart.ps1`), das ausgeführt wird, wenn Sie App Server neu starten in der Managementkonsole auswählen. AWS

Example aws-windows-deployment-manifest.json — benutzerdefinierte Bereitstellung

```
{
  "manifestVersion": 1,
  "deployments": {
    "custom": [
      {
        "name": "Custom site",
        "architecture" : 32,
        "scripts": {
          "install": {
            "file": "siteInstall.ps1"
          },
          "restart": {
            "file": "siteRestart.ps1"
          },
          "uninstall": {
            "file": "siteUninstall.ps1"
          }
        }
      }
    ]
  }
}
```

Fügen Sie alle Artefakte hinzu, die erforderlich sind, um die Anwendung in Ihrem Quell-Bundle mit dem Manifest und Skripts auszuführen.

Example C ustom-site-bundle — .zip

```
.
|-- aws-windows-deployment-manifest.json
|-- siteInstall.ps1
|-- siteRestart.ps1
|-- siteUninstall.ps1
`-- site-contents.zip
```

Schemareferenz für das Bereitstellungsmanifest

Das Deployment-Manifest ist eine JSON-Datei, die definiert, wie Elastic Beanstalk Ihre Windows-Anwendungen bereitstellen und konfigurieren soll. Dieser Abschnitt enthält eine umfassende Referenz für alle unterstützten Eigenschaften und Konfigurationsoptionen im Manifestschema.

Manifest-Struktur

Das Bereitstellungsmanifest folgt einem bestimmten JSON-Schema mit der folgenden Struktur auf oberster Ebene:

Example Grundlegende Manifeststruktur

```
{
  "manifestVersion": 1,
  "skipIISReset": false,
  "iisConfig": {
    "websites": [...],
    "appPools": [...]
  },
  "deployments": {
    "msDeploy": [...],
    "aspNetCoreWeb": [...],
    "custom": [...]
  }
}
```

Eigenschaften der obersten Ebene

`manifestVersion` (Erforderlich)

Typ: Zahl

Standard: 1

Gültige Werte: 1

Gibt die Version des Manifestschemas an. Derzeit wird nur Version 1 unterstützt.

`skipIISReset` (optional)

Typ: Boolesch

Standard: false

Steuert, ob IIS während der Anwendungsbereitstellung zurückgesetzt wird. Dieses Flag betrifft sowohl Bereitstellungstypen als `msDeploy` auch `aspNetCoreWeb` Bereitstellungstypen.

Verhalten:

- Nicht angegeben oder *`false`* (Standard): IIS-Resets werden bei Installations-, Deinstallations- und Aktualisierungsvorgängen durchgeführt. Dies ist das traditionelle Verhalten.
- *`true`*: IIS-Resets werden bei Bereitstellungsvorgängen übersprungen.

Vorteile:

- Geringere Ausfallzeiten — Bei Anwendungen kommt es während der Bereitstellung zu kürzeren Serviceunterbrechungen.
- Schnellere Bereitstellungen — Eliminiert die Zeit, die IIS benötigt, um vollständig neu zu starten und neu zu initialisieren.

> **Note**
>
> Bei Verwendung `skipIISReset` führt der RestartAppServerVorgang unabhängig von dieser Flag-Einstellung einen IIS-Reset durch.

Beispiel:

```
{
  "manifestVersion": 1,
  "skipIISReset": true,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "archive": "dotnet-core-app.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

`deployments` (Erforderlich)

Typ: Objekt

Enthält die Bereitstellungskonfigurationen für Ihre Anwendungen. Dieses Objekt kann `custom` Bereitstellungstypen `msDeployaspNetCoreWeb`, und enthalten.

`iisConfig` (optional)

Typ: Objekt

Definiert IIS-Konfigurationseinstellungen, die vor der Bereitstellung von Anwendungen angewendet werden sollen. Unterstützt sowohl die Website- als auch die Anwendungspoolkonfiguration.

IIS-Konfiguration

In `iisConfig` diesem Abschnitt können Sie IIS-Einstellungen konfigurieren, bevor Sie Ihre Anwendungen bereitstellen. Dazu gehören die Einrichtung von Anwendungspools mit bestimmten Konfigurationen und die Konfiguration von IIS-Websites mit benutzerdefinierten Bindungen.

IIS-Websites

IIS-Websites ermöglichen es Ihnen, benutzerdefinierte Website-Einstellungen, einschließlich physischer Pfade und Netzwerkbindungen, zu konfigurieren, bevor Sie Ihre Anwendungen bereitstellen.

Wichtige Überlegungen zur Erstellung verschiedener IIS-Websites

- Reihenfolge der Einrichtung der Website: Websites werden sequentiell in der Reihenfolge konfiguriert, in der sie im `websites` Array erscheinen. Die Plattform verarbeitet jede Website-Konfiguration der Reihe nach. Achten Sie daher auf die richtige Reihenfolge, wenn Sie Abhängigkeiten zwischen Websites haben.
- Firewall und Portzugriff: Nur Port 80 wird durch die standardmäßige Windows-Firewallkonfiguration von Elastic Beanstalk automatisch verfügbar gemacht. Wenn Sie Websites so konfigurieren, dass sie nicht standardmäßige Ports verwenden, müssen Sie mithilfe von EExtensions oder benutzerdefinierten Deployment-Skripten benutzerdefinierte Firewall-Regeln definieren, um externen Zugriff auf diese Ports zu ermöglichen.

Example Konfiguration der Website

```
{
  "iisConfig": {
    "websites": [
```

```
      {
        "name": "MyCustomSite",
        "physicalPath": "C:\inetpub\wwwroot\mysite",
        "bindings": [
          {
            "protocol": "http",
            "port": 8080,
            "hostName": "mysite.local"
          },
          {
            "protocol": "https",
            "port": 8443
          }
        ]
      }
    ]
  }
}
```

Eigenschaften der Website

`name` (Erforderlich)

Typ: Zeichenfolge

Der Name der IIS-Website. Dieser Name wird zur Identifizierung der Website im IIS-Manager verwendet und muss innerhalb der IIS-Konfiguration eindeutig sein.

`physicalPath` (Erforderlich)

Typ: Zeichenfolge

Der physische Pfad auf dem Server, auf dem die Website-Dateien gespeichert sind. Dieser Pfad muss für den IIS-Arbeitsprozess zugänglich sein.

`bindings` (Erforderlich)

Typ: Array

Mindestanzahl der Artikel: 1

Eine Reihe von Bindungskonfigurationen, die definieren, wie die Website auf Netzwerkanfragen reagiert. Jede Bindung spezifiziert ein Protokoll, einen Port und einen optionalen Hostnamen.

Webseiten-Bindungen

Website-Bindungen definieren die Netzwerkendpunkte, an denen Ihre IIS-Website auf eingehende Anfragen wartet.

`protocol` (Erforderlich)

Typ: Zeichenfolge

Gültige Werte: „http“, „https“

Das für die Bindung verwendete Protokoll.

`port` (Erforderlich)

Typ: Ganzzahl

Gültiger Bereich: 1-65535

Die Portnummer, über die die Website Anfragen abhört.

`hostName` (optional)

Typ: Zeichenfolge

Der Hostname (Domainname) für die Bindung.

Anwendungspools

Anwendungspools bieten eine Isolierung zwischen Anwendungen und ermöglichen es Ihnen, Laufzeiteinstellungen für Anwendungsgruppen zu konfigurieren.

Example Konfiguration des Anwendungspools

```
{
  "iisConfig": {
    "appPools": [
      {
        "name": "MyAppPool",
        "enable32Bit": false,
        "managedPipelineMode": "Integrated",
        "managedRuntimeVersion": "v4.0",
        "queueLength": 1000,
```

```
        "cpu": {
          "limitPercentage": 80,
          "limitAction": "Throttle",
          "limitMonitoringInterval": 5
        },
        "recycling": {
          "regularTimeInterval": 1440,
          "requestLimit": 10000,
          "memory": 1048576,
          "privateMemory": 524288
        }
      }
    ]
  }
}
```

Eigenschaften des Anwendungspools

`name` (Erforderlich)

Typ: Zeichenfolge

Der Name des Anwendungspools. Dieser Name wird verwendet, um in Bereitstellungskonfigurationen auf den Pool zu verweisen.

`enable32Bit` (optional)

Typ: Boolesch

Ermöglicht die Ausführung einer 32-Bit-Anwendung auf einer 64-Bit-Version von Windows. Wird `true` für ältere Anwendungen, die 32-Bit-Kompatibilität erfordern, auf gesetzt.

`managedPipelineMode` (optional)

Typ: Zeichenfolge

Gültige Werte: „Integrated“, „Classic“

Gibt den Modus zur Anforderungsverarbeitung für den Anwendungspool an.

`managedRuntimeVersion` (optional)

Typ: Zeichenfolge

Gültige Werte: „Kein verwalteter Code“, „v2.0", „v4.0"

Gibt die .NET Framework-Version für den Anwendungspool an.

`queueLength` (optional)

Typ: Ganzzahl

Maximale Anzahl von Anfragen, die HTTP.sys für den Anwendungspool in die Warteschlange stellt, bevor weitere Anfragen zurückgewiesen werden.

CPU-Konfiguration

Das `cpu` Objekt konfiguriert die CPU-Nutzungslimits und die Überwachung für den Anwendungspool.

`limitPercentage` (optional)

Typ: Zahl

Maximaler Prozentsatz der CPU-Zeit, den Arbeitsprozesse im Anwendungspool verbrauchen können.

`limitAction` (optional)

Typ: Zeichenfolge

Gültige Werte: "NoAction„, „KillW3WP“, „Throttle“, "“ ThrottleUnderLoad

Aktion, die ergriffen werden soll, wenn das CPU-Limit erreicht ist.

`limitMonitoringInterval` (optional)

Typ: Zahl

Reset-Zeitraum (in Minuten) für CPU-Überwachungs- und Drosselungsgrenzwerte.

Konfiguration recyceln

Das `recycling` Objekt konfiguriert, wann und wie Worker-Prozesse im Anwendungspool wiederverwendet werden.

`regularTimeInterval` (optional)

Typ: Ganzzahl

Zeitintervall (in Minuten), nach dem der Anwendungspool wiederverwendet wird. Auf 0 setzen, um das zeitbasierte Recycling zu deaktivieren.

`requestLimit` (optional)

Typ: Ganzzahl

Maximale Anzahl von Anfragen, die der Anwendungspool vor dem Recycling verarbeitet.

`memory` (optional)

Typ: Ganzzahl

Menge an virtuellem Speicher (in Kilobyte), die das Recycling von Arbeitsprozessen auslöst.

`privateMemory` (optional)

Typ: Ganzzahl

Menge an privatem Speicher (in Kilobyte), der das Recycling von Arbeitsprozessen auslöst.

Bereitstellungstypen

Das `deployments` Objekt enthält Arrays von Bereitstellungskonfigurationen für verschiedene Anwendungstypen. Jeder Bereitstellungstyp hat spezifische Eigenschaften und Anwendungsfälle.

MSDeploy Bereitstellungen

MSDeploy Bereitstellungen werden für herkömmliche.NET-Framework-Anwendungen verwendet, die mit Web Deploy () MSDeploy bereitgestellt werden können.

Example MSDeploy Bereitstellungskonfiguration

```
{
  "deployments": {
    "msDeploy": [
      {
        "name": "WebApp",
        "description": "Main web application",
        "parameters": {
          "appBundle": "webapp.zip",
          "iisPath": "/",
          "appPool": "DefaultAppPool"
```

```
        }
      }
    ]
  }
}
```

MSDeploy Eigenschaften der Bereitstellung

`name` (Erforderlich)

Typ: Zeichenfolge

Eindeutiger Name für die Bereitstellung. Dieser Name muss für alle Bereitstellungen im Manifest eindeutig sein.

`description` (optional)

Typ: Zeichenfolge

Für Menschen lesbare Beschreibung der Bereitstellung.

`parameters` (Erforderlich)

Typ: Objekt

Konfigurationsparameter für den MSDeploy Vorgang.

`scripts` (optional)

Typ: Objekt

PowerShell Skripts, die in verschiedenen Phasen des Bereitstellungslebenszyklus ausgeführt werden.

MSDeploy Parameter

`appBundle` (Erforderlich)

Typ: Zeichenfolge

Pfad zum Anwendungspaket (ZIP-Datei) relativ zur Manifestdatei. Dieses Paket enthält die Anwendungsdateien, die bereitgestellt werden sollen.

`iisWebSite` (optional)

Typ: Zeichenfolge

Standard: „Standardwebsite“

Die IIS-Website, auf der die Anwendung bereitgestellt werden soll. Standardmäßig werden Anwendungen auf der „Standardwebsite“ bereitgestellt. Optional können Sie einen anderen Namen für die Website angeben, z. B. einen, der im `iisConfig.websites` Abschnitt konfiguriert wurde.

`iisPath` (optional)

Typ: Zeichenfolge

Standard: „/“

Virtueller Verzeichnispfad in IIS, in dem die Anwendung bereitgestellt wird. Verwenden Sie „/“ für den Stammpfad oder „/api“ für ein Unterverzeichnis.

`appPool` (optional)

Typ: Zeichenfolge

Name des Anwendungspools, in dem diese Anwendung ausgeführt werden soll.

ASP.NET Core-Bereitstellungen

ASP.NET Core-Bereitstellungen wurden speziell für .NET Core- und .NET 5+-Anwendungen entwickelt.

Example Konfiguration der ASP.NET Core-Bereitstellung

```
{
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "CoreAPI",
        "description": "ASP.NET Core Web API",
        "parameters": {
          "appBundle": "coreapi.zip",
          "iisPath": "/api",
          "appPool": "CoreAppPool"
```

```
        }
      }
    ]
  }
}
```

ASP.NET Core-Bereitstellungen verwenden dieselbe Eigenschaftsstruktur wie MSDeploy Bereitstellungen, wobei der Hauptunterschied in der für die Anwendung verwendeten Laufzeitumgebung und dem Hostingmodell besteht.

ASP.NET Core-Bereitstellungsparameter

`appBundle` (Erforderlich)

Typ: Zeichenfolge

Pfad zum Anwendungspaket relativ zur Manifestdatei. Dies kann entweder ein ZIP-Archiv oder ein Verzeichnispfad sein, der die veröffentlichte ASP.NET Core-Anwendung enthält.

`iisWebSite` (optional)

Typ: Zeichenfolge

Standard: „Standardwebsite“

Die IIS-Website, auf der die ASP.NET Core-Anwendung bereitgestellt werden soll. Standardmäßig werden Anwendungen auf der „Standardwebsite“ bereitgestellt. Optional können Sie einen anderen Namen für die Website angeben, z. B. einen, der im `iisConfig.websites` Abschnitt konfiguriert wurde.

`iisPath` (optional)

Typ: Zeichenfolge

Standard: „/“

Virtueller Verzeichnispfad in IIS für die ASP.NET Core-Anwendung.

`appPool` (optional)

Typ: Zeichenfolge

Anwendungspool für die ASP.NET Core-Anwendung. Der Pool wird entsprechend für das ASP.NET Core-Hosting konfiguriert.

Benutzerdefinierte Bereitstellungen

Benutzerdefinierte Bereitstellungen bieten mithilfe PowerShell von Skripten die vollständige Kontrolle über den Bereitstellungsprozess. Dieser Bereitstellungstyp ist für komplexe Szenarien nützlich, die eine benutzerdefinierte Installation, Konfiguration oder Bereitstellungslogik erfordern.

Example Benutzerdefinierte Bereitstellungskonfiguration

```
{
  "deployments": {
    "custom": [
      {
        "name": "CustomService",
        "description": "Custom Windows service deployment",
        "architecture": 32,
        "scripts": {
          "install": {
            "file": "install-service.ps1"
          },
          "restart": {
            "file": "restart-service.ps1"
          },
          "uninstall": {
            "file": "uninstall-service.ps1",
            "ignoreErrors": true
          }
        }
      }
    ]
  }
}
```

Benutzerdefinierte Bereitstellungseigenschaften

`name` (Erforderlich)

Typ: Zeichenfolge

Eindeutiger Name für die benutzerdefinierte Bereitstellung.

`description` (optional)

Typ: Zeichenfolge

Beschreibung der benutzerdefinierten Bereitstellung.

`architecture` (optional)

Typ: Ganzzahl

Standard: 32

Gültige Werte: 32, 64

Die Architekturspezifikation für den Ausführungsmodus von Powershell-Skripten

`scripts` (Erforderlich)

Typ: Objekt

PowerShell Skripten, die das Bereitstellungsverhalten definieren. Benutzerdefinierte Bereitstellungen unterstützen im Vergleich zu anderen Bereitstellungstypen zusätzliche Skripttypen.

Bereitstellungsskripte

PowerShell Bereitstellungsskripten sind Skripten, die zu bestimmten Zeitpunkten während des Bereitstellungszyklus ausgeführt werden. Verschiedene Bereitstellungstypen unterstützen unterschiedliche Gruppen von Skriptereignissen.

Skriptereignisse

Die folgenden Skriptereignisse sind je nach Bereitstellungstyp verfügbar:

Standardbereitstellungsskripts (MSDeploy und aspNetCore Web)

`preInstall`

Wird ausgeführt, bevor die Anwendung installiert oder aktualisiert wird.

`postInstall`

Wird ausgeführt, nachdem die Anwendung installiert oder aktualisiert wurde.

`preRestart`

Wird ausgeführt, bevor die Anwendung neu gestartet wird.

`postRestart`

Wird ausgeführt, nachdem die Anwendung neu gestartet wurde.

`preUninstall`

Wird ausgeführt, bevor die Anwendung deinstalliert wird.

`postUninstall`

Wird ausgeführt, nachdem die Anwendung deinstalliert wurde.

Benutzerdefinierte Bereitstellungsskripten (nur benutzerdefinierte Bereitstellungen)

`install`

Primäres Installationsskript für benutzerdefinierte Bereitstellungen. Dieses Skript ist für die Installation der Anwendung oder des Dienstes verantwortlich.

`restart`

Skript zum Neustarten der Anwendung oder des Dienstes. Wird aufgerufen, wenn die Umgebung neu gestartet wird.

`uninstall`

Skript zur Deinstallation der Anwendung oder des Dienstes. Wird beim Beenden der Umgebung oder beim Entfernen der Anwendung aufgerufen.

Eigenschaften des Skripts

Jedes Skript ist als Objekt mit den folgenden Eigenschaften definiert:

`file` (Erforderlich)

Typ: Zeichenfolge

Pfad zur PowerShell Skriptdatei relativ zur Manifestdatei. Das Skript sollte eine `.ps1` Erweiterung haben.

`ignoreErrors` (optional)

Typ: Boolesch

Standard: false

Wenn diese Option auf gesetzt ist`true`, wird die Bereitstellung auch dann fortgesetzt, wenn das Skript fehlschlägt. Verwenden Sie diese Option für unkritische Skripts oder Bereinigungsvorgänge.

Example Beispiel für eine Skriptkonfiguration

```
{
  "scripts": {
    "preInstall": {
      "file": "backup-config.ps1",
      "ignoreErrors": true
    },
    "postInstall": {
      "file": "configure-app.ps1"
    }
  }
}
```

Verwenden von EC2 Fast Launch mit Windows-Plattformzweigen

Die EC2 Schnellstartfunktion reduziert die Startzeiten von Windows-Instances in Ihren Elastic Beanstalk Beanstalk-Umgebungen. In diesem Thema erfahren Sie, wie Sie diese Funktion in Ihren Elastic Beanstalk Beanstalk-Umgebungen verwenden können. Beginnend mit der Windows-Plattformversion 2.16.2, veröffentlicht am 22. Januar 2025, enthalten Elastic Beanstalk-Plattformversionen Basisversionen AMIs mit EC2 aktiviertem Fast Launch.

Standardmäßige Verfügbarkeit von Fast Launch EC2

Die neuesten Versionen der Elastic Beanstalk Windows-Plattform enthalten eine Basisversion AMIs mit automatisch aktiviertem EC2 Fast Launch, ohne dass zusätzliche Kosten anfallen. Wenn jedoch neuere Plattformversionen veröffentlicht werden, bleibt EC2 Fast Launch möglicherweise nicht automatisch auf der Basis AMIs älterer Plattformversionen aktiviert.

Wir empfehlen, auf die neueste Windows-Plattformversion zu aktualisieren, um die Basisversion AMIs mit automatisch aktiviertem EC2 Fast Launch zu verwenden. Wenn Sie jedoch Ihre bestehende Plattformversion weiterhin verwenden müssen, können Sie EC2 Fast Launch auf dem Basis-AMI Ihrer Umgebung manuell aktivieren. Detaillierte Anweisungen finden Sie unter Manuelles Konfigurieren von EC2 Fast Launch.

Manuelles Konfigurieren von EC2 Fast Launch

> **Note**
>
> Die manuelle Aktivierung von EC2 Fast Launch kann im Vergleich zur Verwendung von Plattformversionen mit automatisch aktiviertem EC2 Fast Launch zusätzliche Kosten verursachen. Weitere Informationen zu den EC2 Fast Launch-Kosten finden Sie auf der Seite Kosten für EC2 Fast Launch verwalten im EC2 Amazon-Benutzerhandbuch.

Gehen Sie wie folgt vor, um EC2 Fast Launch auf einem Windows-Basis-AMI zu aktivieren, das von Ihrer Elastic Beanstalk Beanstalk-Umgebung verwendet wird:

So aktivieren Sie EC2 Fast Launch manuell für Ihre Elastic Beanstalk Beanstalk-Umgebung

1. Identifizieren Sie das Basis-AMI Ihrer Umgebung:

   Folgen Sie den Schritten unter Benutzerdefiniertes AMI erstellen, um die Basis-AMI-ID Ihrer Umgebung zu ermitteln. Beachten Sie, dass Sie kein benutzerdefiniertes AMI erstellen müssen — Sie müssen nur die Schritte befolgen, um Ihre aktuelle Basis-AMI-ID zu finden.
2. Aktivieren Sie EC2 Fast Launch auf dem AMI:

   Verwenden Sie die Anweisungen unter Enable EC2 Fast Launch im EC2 Amazon-Benutzerhandbuch, um EC2 Fast Launch für Ihr AMI zu konfigurieren.

Hinzufügen einer Amazon RDS-DB-Instance zur .NET-Anwendungsumgebung

Dieses Thema enthält Anweisungen zum Erstellen eines Amazon RDS mit der Elastic-Beanstalk-Konsole. Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann an Ihre Umgebung gekoppelt sein und von Elastic Beanstalk verwaltet werden oder sie kann als entkoppelte Datenbank erstellt und extern von einem anderen Service verwaltet werden. In dieser Anleitung wird die Datenbank an Ihre Umgebung gekoppelt und von Elastic Beanstalk verwaltet. Weitere Informationen zur Integration eines Amazon RDS mit Elastic Beanstalk finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Sections

- Hinzufügen einer DB-Instance zu Ihrer Umgebung
- Herunterladen eines Treibers
- Verbinden mit einer Datenbank

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftenwert
`RDS_HOSTNAME`	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
`RDS_PORT`	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.

Eigenschaftenname	Beschreibung	Eigenschaftenwert
`RDS_DB_NAME`	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
`RDS_USERNAME`	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
`RDS_PASSWORD`	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.

Weitere Informationen zur Konfiguration einer Datenbank-Instance in Verbindung mit einer Elastic-Beanstalk-Umgebung finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Herunterladen eines Treibers

Laden Sie das `EntityFramework`-Paket und einen Datenbanktreiber für Ihre Entwicklungsumgebung mit `NuGet` herunter und installieren sie es bzw. ihn.

Gängige Entitäts-Framework-Datenbankanbieter für .NET

- SQL Server – `Microsoft.EntityFrameworkCore.SqlServer`
- MySQL – `Pomelo.EntityFrameworkCore.MySql`
- PostgreSQL – `Npgsql.EntityFrameworkCore.PostgreSQL`

Verbinden mit einer Datenbank

Elastic Beanstalk stellt Verbindungsinformationen für zugehörige DB-Instances in Umgebungseigenschaften bereit. Mit `ConfigurationManager.AppSettings` können Sie die Eigenschaften lesen und eine Datenbankverbindung konfigurieren.

Example Helpers.cs – Verbindungszeichenfolgen-Methode

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Web;

namespace MVC5App.Models
{
  public class Helpers
  {
    public static string GetRDSConnectionString()
    {
      var appConfig = ConfigurationManager.AppSettings;

      string dbname = appConfig["RDS_DB_NAME"];

      if (string.IsNullOrEmpty(dbname)) return null;

      string username = appConfig["RDS_USERNAME"];
      string password = appConfig["RDS_PASSWORD"];
      string hostname = appConfig["RDS_HOSTNAME"];
      string port = appConfig["RDS_PORT"];

      return "Data Source=" + hostname + ";Initial Catalog=" + dbname + ";User ID=" +
 username + ";Password=" + password + ";";
    }
  }
}
```

Verwenden Sie die Verbindungszeichenfolge, um Ihren Datenbankkontext zu initialisieren.

Example DBContext.cs

```
using System.Data.Entity;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;

namespace MVC5App.Models
{
```

```
  public class RDSContext : DbContext
  {
    public RDSContext()
      : base(GetRDSConnectionString())
    {
    }

    public static RDSContext Create()
    {
      return new RDSContext();
    }
  }
}
```

Die AWS Toolkit for Visual Studio

Visual Studio bietet Vorlagen für verschiedene Programmiersprachen und Anwendungstypen. Sie können mit einer beliebigen Vorlage beginnen. Das AWS Toolkit for Visual Studio bietet außerdem drei Projektvorlagen, die die Entwicklung Ihrer Anwendung beschleunigen: AWS Console Project, AWS Web Project und AWS Empty Project. In diesem Beispiel erstellen Sie eine neue ASP.NET-Webanwendung.

So erstellen Sie ein neues ASP.NET-Webanwendungsprojekt:

1. Klicken Sie in Visual Studio im Menü Datei auf Neu und dann auf Projekt.
2. Klicken Sie im Dialogfeld Neues Projekt auf Installierte Vorlagen, auf Visual C# und auf Web. Klicken Sie auf Leere ASP.NET-Webanwendung, geben Sie einen Projektnamen ein und klicken Sie auf OK.

So führen Sie ein Projekt aus

Führen Sie eine der folgenden Aufgaben aus:

1. Drücken Sie F5.
2. Wählen Sie im Menü Debuggen die Option Debuggen starten aus.

Ausführen lokaler Tests

Mit Visual Studio können Sie die Anwendung einfach lokal testen. Zum Testen oder Ausführen von ASP.NET-Webanwendungen ist ein Webserver erforderlich. Visual Studio bietet mehrere Optionen, z. B. Internet Information Services (IIS), IIS Express oder den integrierten Visual Studio Development Server. Für weitere Informationen zu den einzelnen Optionen und zur Auswahl der am besten für Sie geeigneten Option rufen Sie Webserver in Visual Studio für ASP.NET-Webprojekte auf.

Erstellen einer Elastic Beanstalk-Umgebung

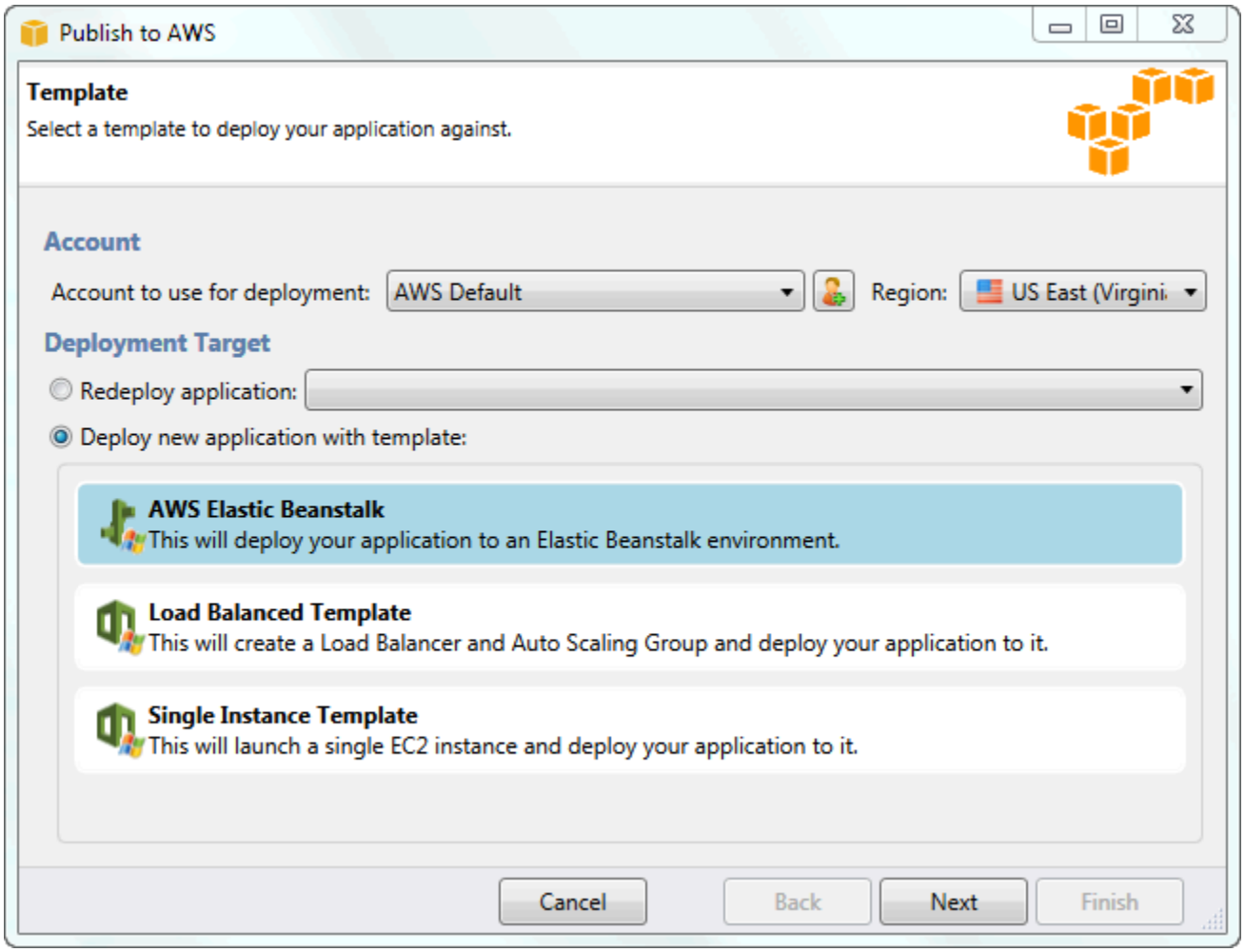
Nachdem Sie die Anwendung getestet haben, können Sie diese nun in Elastic Beanstalk bereitstellen.

> **Note**
>
> Die Konfigurationsdatei muss Teil des Projekts sein und in das Archiv eingebunden werden. Anstatt die Konfigurationsdateien in das Projekt einzubinden, können Sie mit Visual Studio alternativ alle Dateien im Projektordner bereitstellen. Klicken Sie im Projektmappen-Explorer mit der rechten Maustaste auf den Projektnamen und klicken Sie dann auf Eigenschaften. Klicken Sie auf die Registerkarte Web packen/veröffentlichen. Wählen Sie unter Bereitzustellende Elemente in der Dropdown-Liste die Option Alle Dateien in diesem Projektordner aus.

So stellen Sie Ihre Anwendung mithilfe des AWS Toolkits für Visual Studio auf Elastic Beanstalk bereit

1. Klicken Sie im Solution Explorer mit der rechten Maustaste auf Ihre Anwendung und wählen Sie dann Veröffentlichen in. AWS
2. Geben Sie im Assistenten In AWS veröffentlichen Ihre Kontoinformationen ein.
   a. Wählen Sie unter AWS -Konto für Bereitstellung Ihr Konto aus. Alternativ wählen Sie Anderes aus, um neue Kontoinformationen anzugeben.
   b. Wählen Sie unter Region die Region aus, in der die Anwendung bereitgestellt werden soll. Informationen zu verfügbaren AWS Regionen finden Sie unter AWS Elastic Beanstalk Endpunkte und Kontingente in der Allgemeine AWS-Referenz. Wenn Sie eine Region auswählen, die nicht von Elastic Beanstalk unterstützt wird, ist die Option zur Bereitstellung in Elastic Beanstalk nicht verfügbar.

c. Klicken Sie auf Deploy new application with template (Neue Anwendung mit Vorlage bereitstellen) und wählen Sie Elastic Beanstalk aus. Klicken Sie dann auf Next (Weiter).



3. Geben Sie auf der Seite Application (Anwendung) die Anwendungsdetails an.

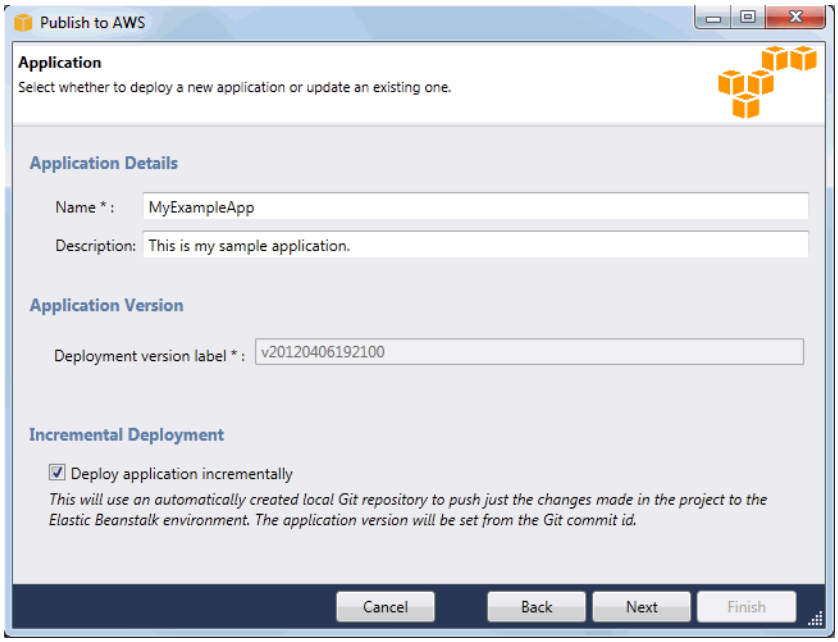
   a. Geben Sie im Feld Name den Namen der Anwendung ein.

   b. Geben Sie im Feld Description eine Beschreibung der Anwendung ein. Dieser Schritt ist optional.

   c. Die Versionsbezeichnung der Anwendung wird automatisch im Feld Deployment version label (Versionsbezeichnung der Bereitstellung) angezeigt.

   d. Wählen Sie Deploy application incrementally (Anwendung inkrementell bereitstellen) aus, wenn nur die geänderten Dateien bereitgestellt werden sollen. Eine inkrementelle Bereitstellung ist schneller, da anstelle von allen Dateien nur solche aktualisiert werden, die geändert wurden. Bei Auswahl dieser Option wird die Anwendungsversion über die Git-Commit-ID festgelegt. Soll die Anwendung nicht inkrementell bereitgestellt werden, können

Sie die Versionsbezeichnung im Feld Deployment version label (Versionsbezeichnung der Bereitstellung) aktualisieren.



e. Klicken Sie auf Weiter.

4. Beschreiben Sie auf der Seite Environment (Umgebung) die Details der Umgebung.

   a. Wählen Sie Create a new environment for this application (Neue Umgebung für diese Anwendung erstellen) aus.

   b. Geben Sie unter Name einen Namen für die Umgebung ein.

   c. Geben Sie im Feld Description (Beschreibung) eine Beschreibung der Umgebung ein. Dieser Schritt ist optional.

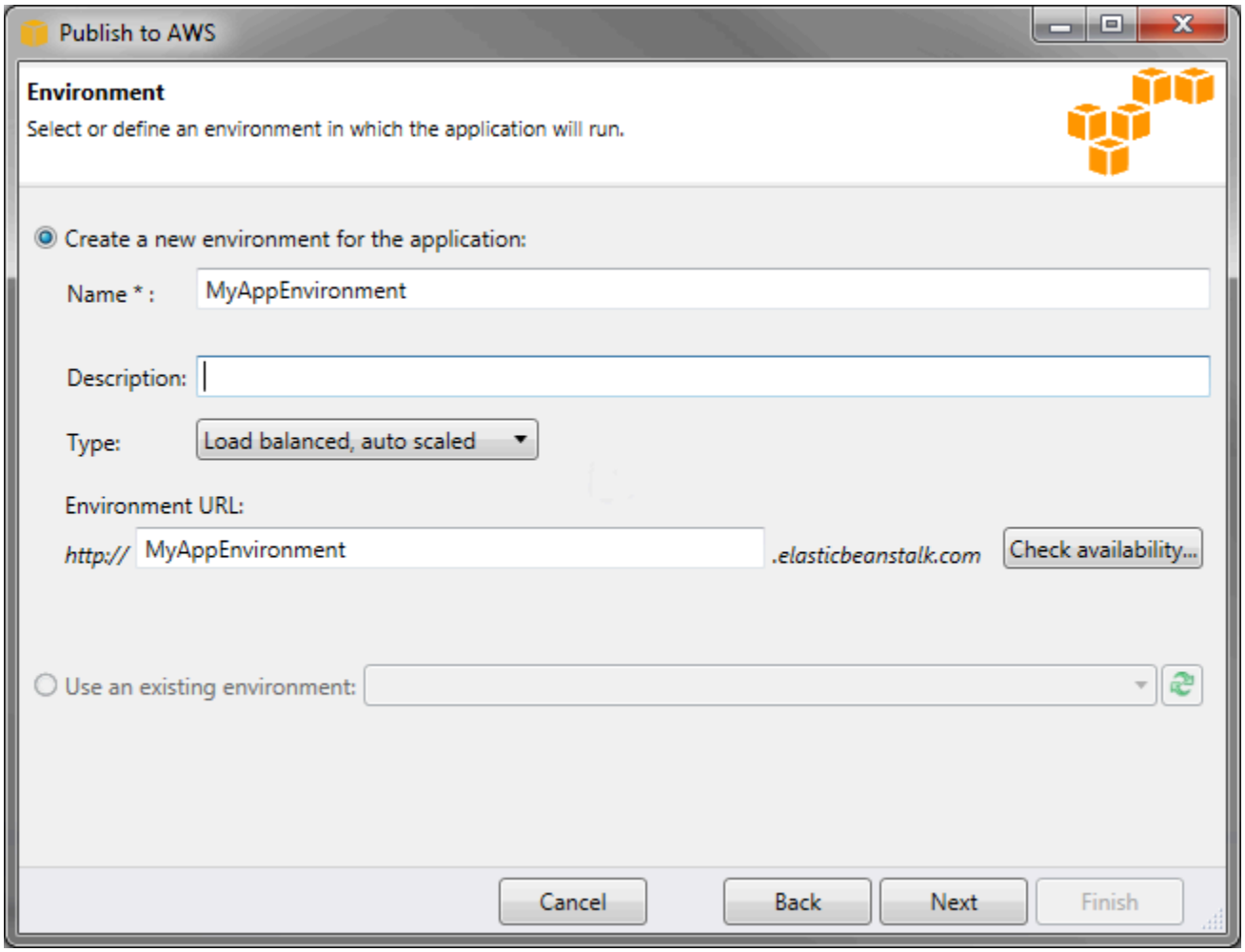
   d. Wählen Sie unter Type (Typ) den gewünschten Umgebungstyp aus.

      Sie können als Umgebungstyp entweder Load balanced, auto scaled (lastenverteilt, automatisch skaliert) oder Single instance (Einzelne Instance) auswählen. Weitere Informationen finden Sie unter Typen von Umgebungen.

> **Note**
>
> Die Einstellungen für Lastausgleich, Auto Scaling und Integritätsprüfungs-URL können nicht auf Single-Instance-Umgebungen angewendet werden.

e. Im Feld Environment URL (Umgebungs-URL) wird die Umgebungs-URL automatisch angezeigt, sobald Sie den Mauszeiger zu diesem Feld bewegen.

f. Klicken Sie auf Check availability (Verfügbarkeit prüfen), um sicherzustellen, dass die URL verfügbar ist.



g. Klicken Sie auf Weiter.

5. Konfigurieren Sie auf der Seite AWS -Optionen zusätzliche Optionen und Sicherheitsinformationen für die Bereitstellung.

a. Wählen Sie im Feld Container Type (Container-Typ) den Eintrag 64bit Windows Server 2012 running IIS 8 (Windows Server 2012 mit 64 Bit und IIS 8) oder 64bit Windows Server 2008 running IIS 7.5 (Windows Server 2008 mit 64 Bit und IIS 7.5) aus.

b. Wählen Sie im Feld Instance Type (Instance-Typ) die Option Micro aus.

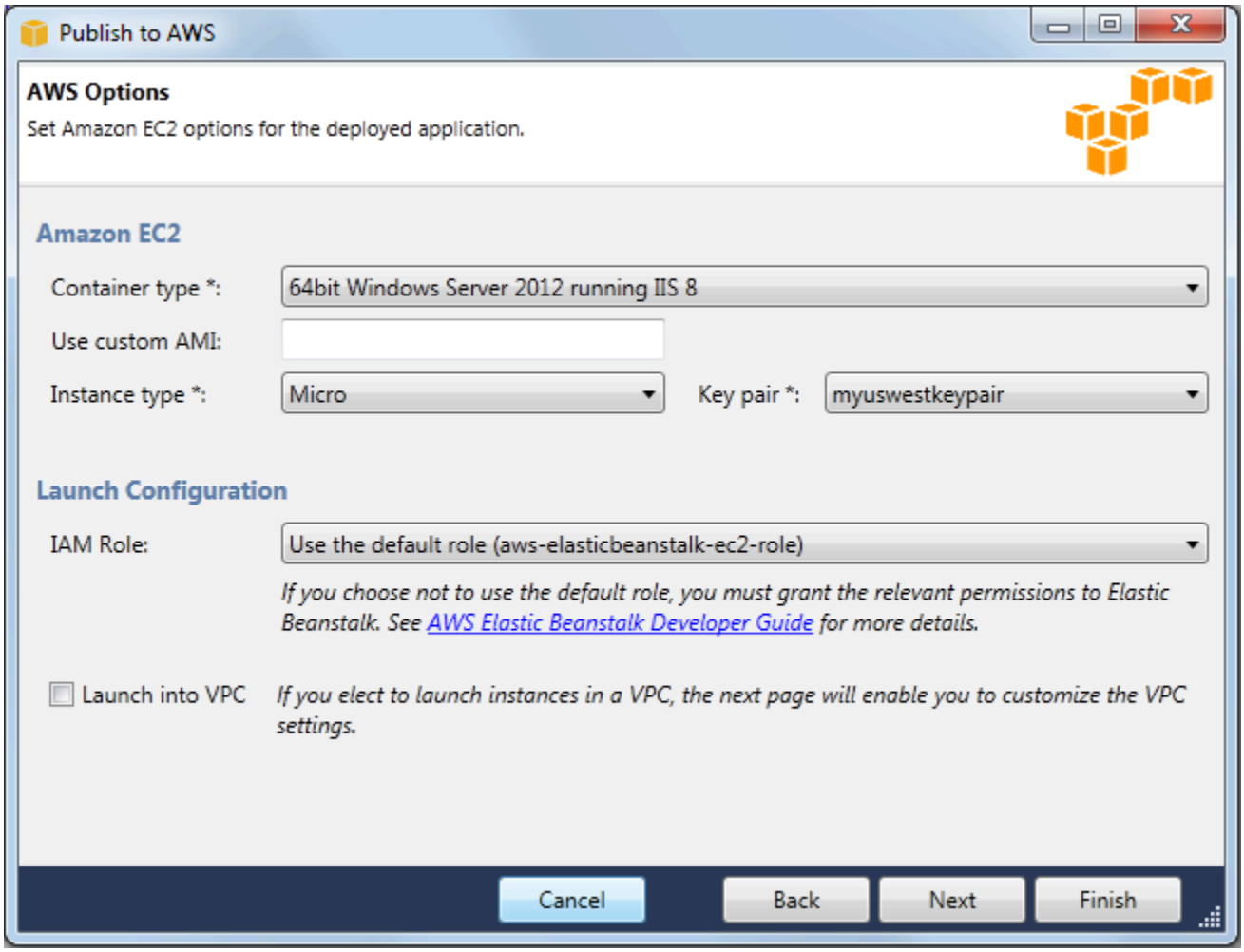
c. Wählen Sie im Feld Key pair (Schlüsselpaar) die Option Create new key pair (Neues Schlüsselpaar erstellen) aus. Geben Sie einen Namen für das neue Schlüsselpaar ein –

in diesem Beispiel wird **`myuswestkeypair`** verwendet – und klicken Sie auf OK. Ein key pair ermöglicht den Remote-Desktop-Zugriff auf Ihre EC2 Amazon-Instances. Weitere Informationen zu EC2 Amazon-Schlüsselpaaren finden Sie unter Using Credentials im Amazon Elastic Compute Cloud-Benutzerhandbuch.

d. Wählen Sie ein Instance-Profil aus.

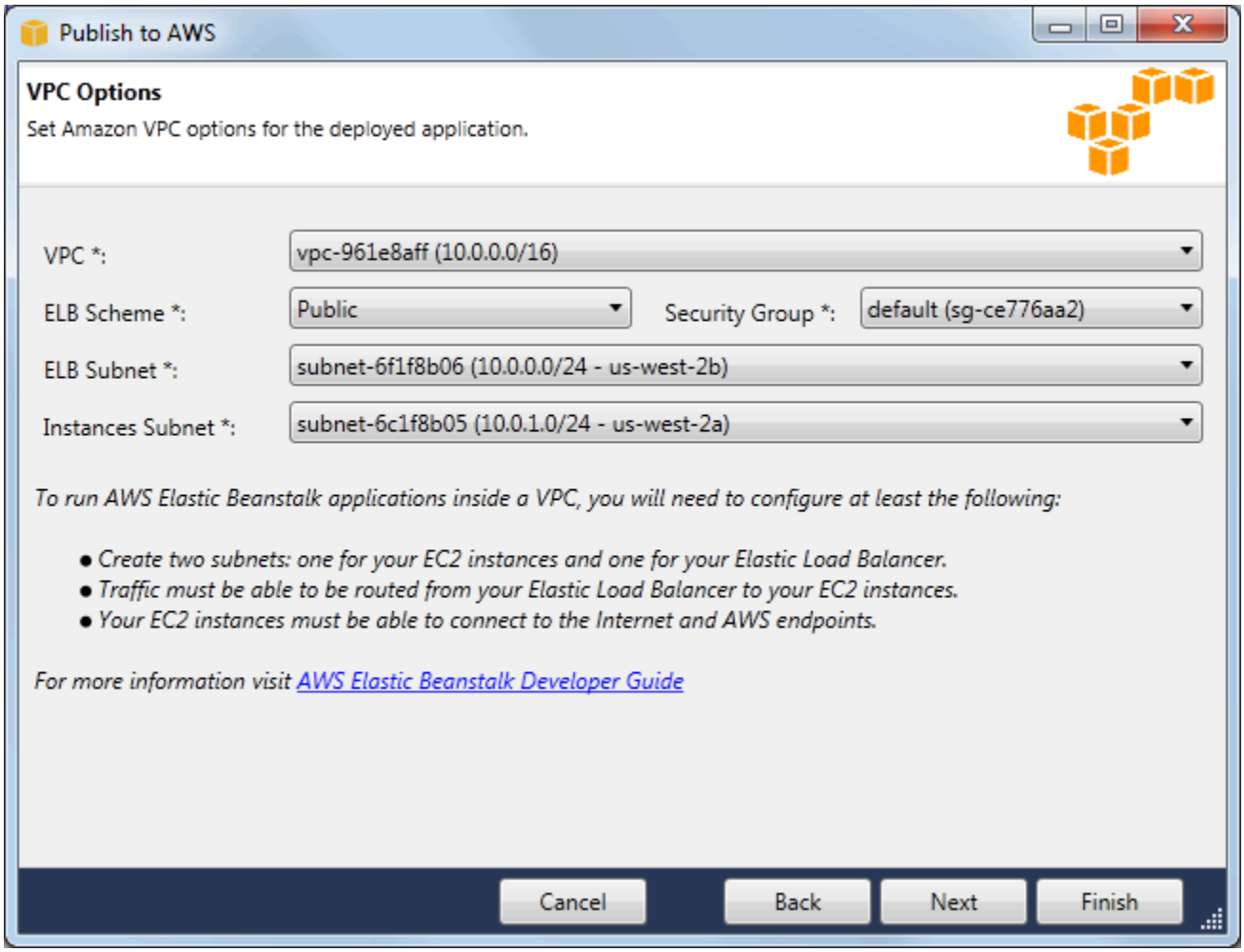
   Falls Sie nicht über ein Instance-Profil verfügen, wählen Sie Instance-Standardprofil erstellen aus. Weitere Informationen zum Verwenden von Instance-Profilen in Elastic Beanstalk finden Sie unter Elastic Beanstalk Instance-Profile verwalten.

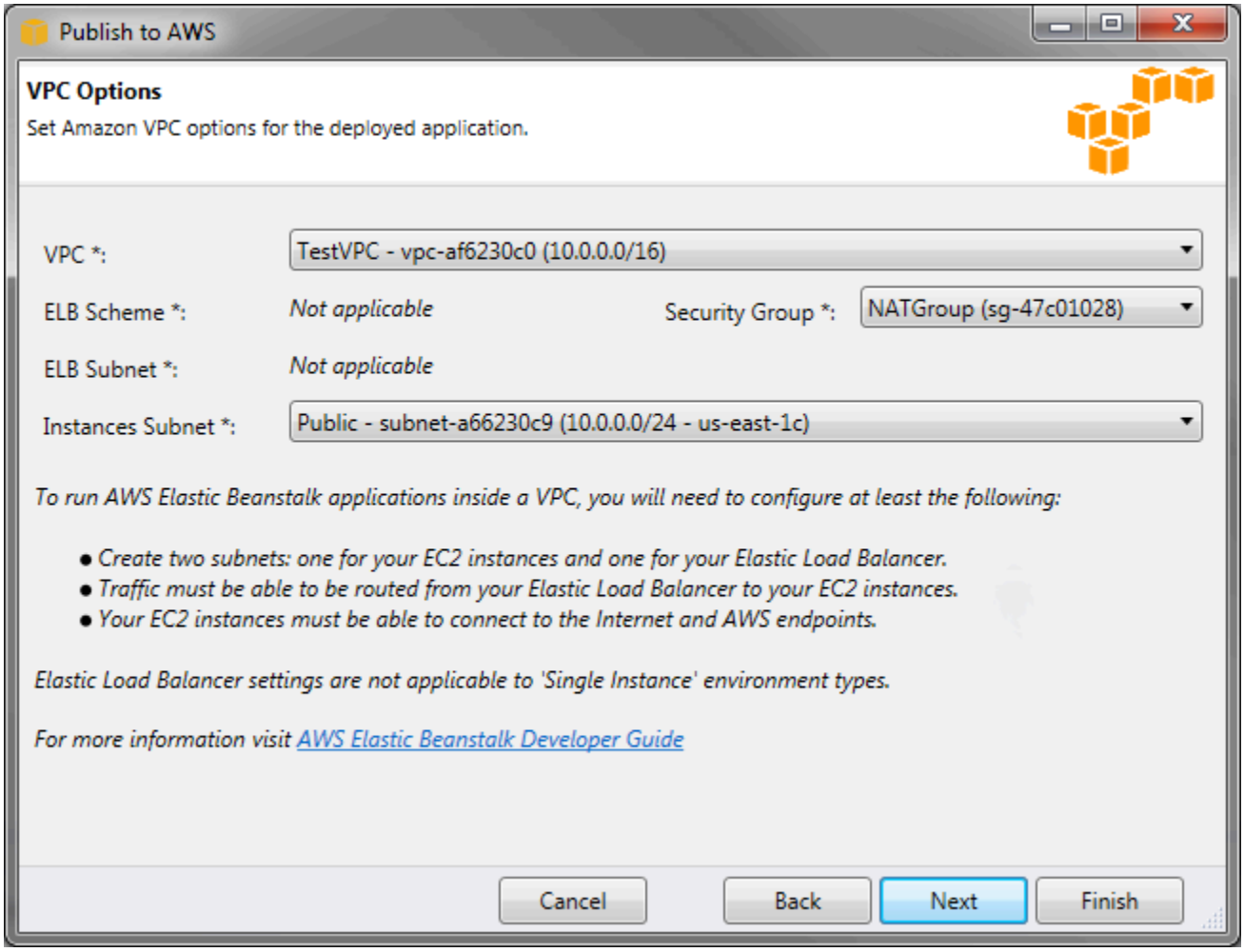
e. Wenn Sie eine benutzerdefinierte VPC mit der Umgebung nutzen möchten, klicken Sie auf Launch into VPC (In VPC starten). Sie können die VPC-Informationen auf der nächsten Seite konfigurieren. Weitere Informationen zu Amazon VPC finden Sie unter Amazon Virtual Private Cloud (Amazon VPC). Eine Liste mit unterstützten aktuellen Containertypen finden Sie unter the section called “Warum sind einige Plattformversionen als Legacy gekennzeichnet?”



f. Klicken Sie auf Weiter.

6. Wenn Sie ausgewählt haben, dass die Umgebung in einer VPC gestartet werden soll, wird die Seite VPC Options (VPC-Optionen) eingeblendet. Andernfalls wird die Seite Additional Options (Weitere Optionen) angezeigt. Hier können Sie die VPC-Optionen konfigurieren.





a. Wählen Sie die VPC-ID der VPC aus, in der die Umgebung gestartet werden soll.

b. Für eine skalierbare Umgebung mit Lastausgleich wählen Sie im Feld ELB Scheme (ELB-Schema) die Option private (privat) aus, wenn der Elastic Load Balancer nicht für das Internet verfügbar sein soll.

   Bei einer Umgebung mit einer einzelnen Instance ist diese Option nicht auswählbar, da kein Load Balancer vorhanden ist. Weitere Informationen finden Sie unter Typen von Umgebungen.

c. Wählen Sie für eine skalierbare Umgebung mit Lastenausgleich die Subnetze für den Elastic Load Balancer und die Instances aus. EC2 Wenn Sie öffentliche und private Subnetze erstellt haben, stellen Sie sicher, dass der Elastic Load Balancer und die EC2 Instances dem richtigen Subnetz zugeordnet sind. Standardmäßig erstellt Amazon VPC ein öffentliches Standardsubnetz unter 10.0.0.0/24 und ein privates Subnetz unter 10.0.1.0/24. Sie können Ihre vorhandenen Subnetze in der Amazon VPC-Konsole unter einsehen. https://console.aws.amazon.com/vpc/

Bei einer Umgebung mit einer einzelnen Instance benötigt die VPC nur ein öffentliches Subnetz für die Instance. Die Auswahl eines Subnetzes für den Load Balancer ist nicht möglich, da für diese Umgebung kein Load Balancer vorhanden ist. Weitere Informationen finden Sie unter Typen von Umgebungen.

d. Für eine skalierbare Umgebung mit Lastausgleich wählen Sie die Sicherheitsgruppe aus, die Sie für Ihre Instances erstellt haben (wenn zutreffend).

   Für eine Umgebung mit einer einzelnen Instance ist kein NAT-Gerät erforderlich. Wählen Sie die Standardsicherheitsgruppe aus. Elastic Beanstalk weist der Instance eine Elastic IP-Adresse zu, sodass sie Zugang zum Internet hat.

e. Klicken Sie auf Weiter.

7. Konfigurieren Sie auf der Seite Application Options (Anwendungsoptionen) die Anwendungsoptionen.

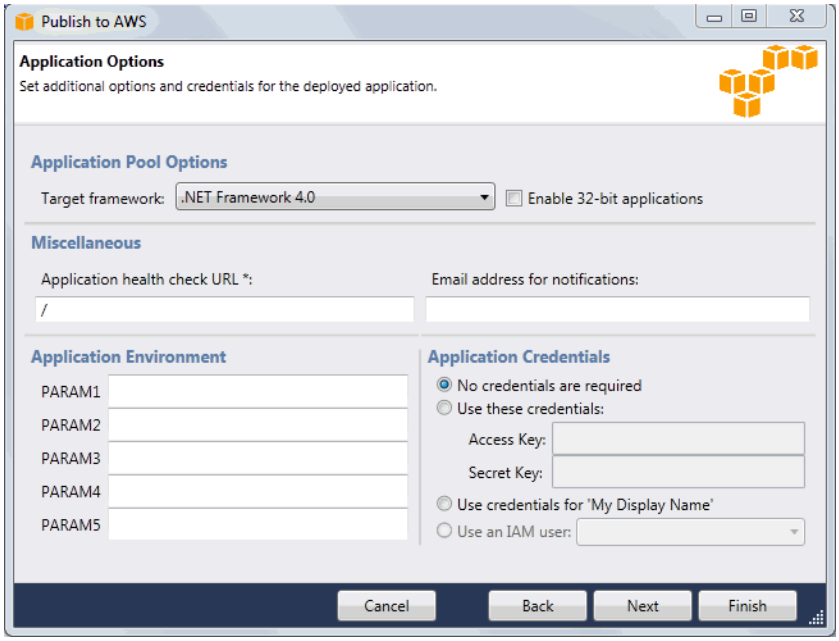
   a. Wählen Sie im Feld "Target framework" die Option .NET Framework 4.0 aus.

   b. Elastic Load Balancing verwendet eine Zustandsprüfung, um festzustellen, ob die EC2 Amazon-Instances, auf denen Ihre Anwendung ausgeführt wird, fehlerfrei sind. Bei der Zustandsprüfung wird der Instance-Zustand durch Überprüfung einer vorgegebenen URL in festgelegten Intervallen bestimmt. Sie können die Standard-URL überschreiben, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/index.aspx`). Dazu geben Sie diese im Feld Application health check URL (Zustandsprüfungs-URL für die Anwendung) ein. Weitere Informationen zu Anwendungszustandsprüfungen finden Sie unter Gesundheitscheck.

   c. Geben Sie eine E-Mail-Adresse an, wenn Sie Amazon Simple Notification Service (Amazon SNS)-Benachrichtigungen zu wichtigen Ereignissen für die Anwendung erhalten möchten.

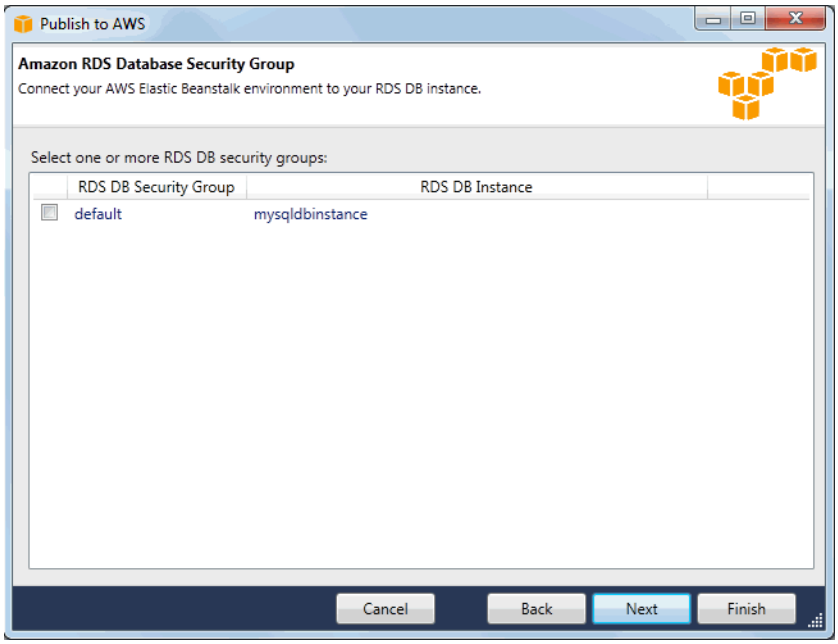
   d. Im Abschnitt Anwendungsumgebung können Sie Umgebungsvariablen für die EC2 Amazon-Instances angeben, auf denen Ihre Anwendung ausgeführt wird. Diese Einstellung ermöglicht eine bessere Portabilität, da der Quellcode beim Wechseln zwischen den Umgebungen nicht neu kompiliert werden muss.

   e. Wählen Sie die für die Anwendungsbereitstellung zu verwendende Option für die Anmeldeinformationen aus.

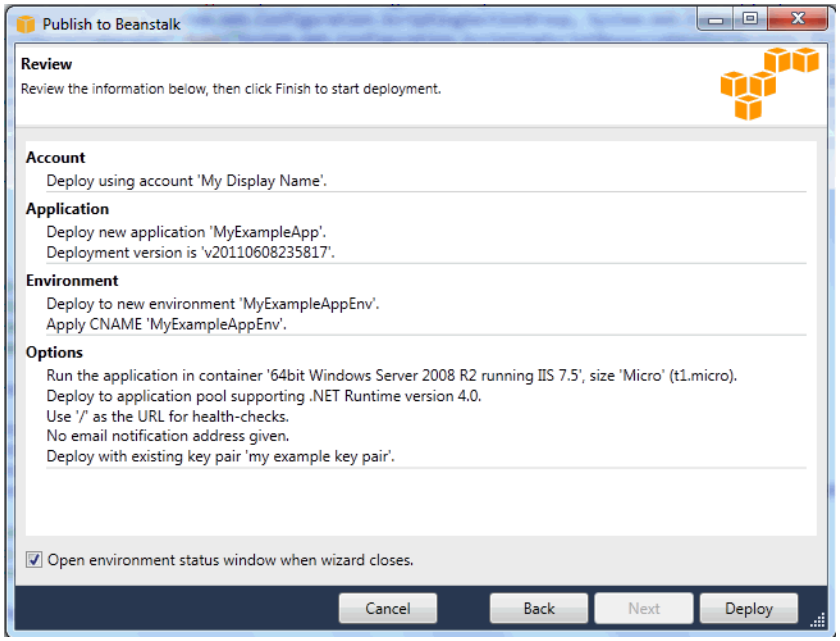


f. Klicken Sie auf Weiter.

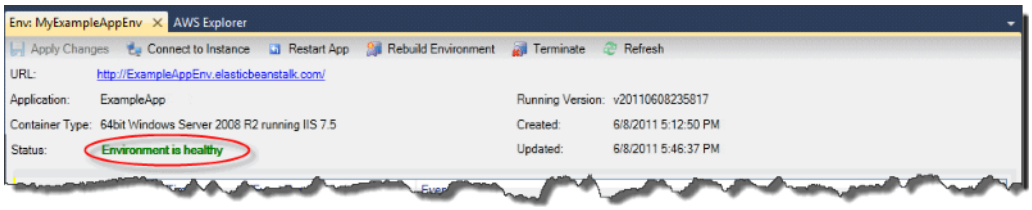
8. Sofern Sie bereits eine Amazon RDS-Datenbank eingerichtet haben, wird die Seite Amazon RDS DB Security Group (Amazon RDS DB-Sicherheitsgruppe) angezeigt. Wenn die Elastic Beanstalk-Umgebung mit der Amazon RDS-DB-Instance verbunden werden soll, wählen Sie eine oder mehrere Sicherheitsgruppen aus. Andernfalls fahren Sie mit dem nächsten Schritt fort. Sobald Sie fertig sind, klicken Sie auf Next (Weiter).



9. Überprüfen Sie die Bereitstellungsoptionen. Wenn alles korrekt ist, klicken Sie auf Deploy (Bereitstellen).



Das ASP.NET-Projekt wird als Webbereitstellungsdatei exportiert, in Amazon S3 hochgeladen und als neue Anwendungsversion in Elastic Beanstalk registriert. Mit der Bereitstellungsfunktion von Elastic Beanstalk wird die Umgebung überwacht, bis sie mit dem neu bereitgestellten Code verfügbar ist. Auf der Registerkarte "env (Umg.):<environment name (Umgebungsname)>" können Sie den Status der Umgebung anzeigen.



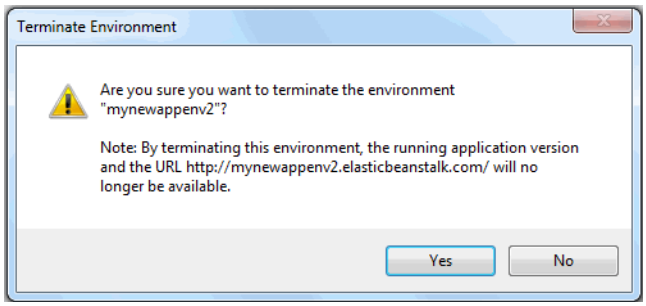
Beenden einer Umgebung

Um zu vermeiden, dass Gebühren für ungenutzte AWS Ressourcen anfallen, können Sie eine laufende Umgebung mit dem AWS Toolkit for Visual Studio beenden.

> **Note**
>
> Sie können zu einem späteren Zeitpunkt problemlos eine neue Umgebung mit derselben Version starten.

So beenden Sie eine Umgebung

1. Erweitern Sie den Elastic-Beanstalk-Knoten und den Anwendungsknoten in AWS Explorer. Klicken Sie mit der rechten Maustaste auf die Anwendungsumgebung und wählen Sie Terminate Environment (Umgebung beenden) aus.
2. Wenn Sie aufgefordert werden, klicken Sie auf Ja, um zu bestätigen, dass die Umgebung beendet werden soll. Es dauert einige Minuten, bis Elastic Beanstalk die in der Umgebung laufenden AWS Ressourcen beendet.



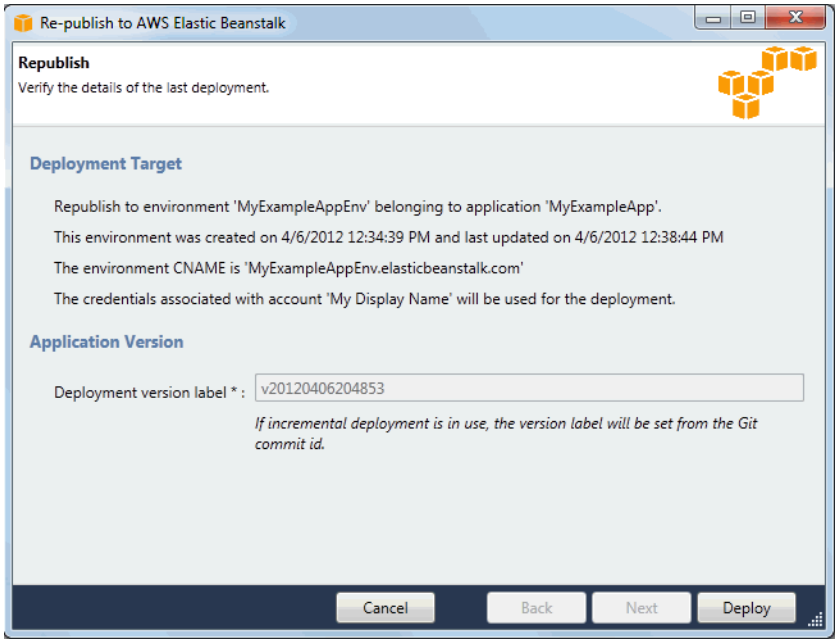
> **Note**
>
> Nachdem die Umgebung beendet ist, steht ihr zugehöriger CNAME wieder zur Verfügung und kann von allen verwendet werden.

Bereitstellen für Ihre Umgebung

Nachdem Sie Ihre Anwendung getestet haben, können Sie Ihre Anwendung einfach bearbeiten und erneut bereitstellen und die Ergebnisse in wenigen Augenblicken anzeigen.

So bearbeiten Sie Ihre ASP.NET-Webanwendung und stellen Sie erneut bereit

1. Klicken Sie im Solution Explorer mit der rechten Maustaste auf Ihre Anwendung, und klicken Sie dann auf In Umgebung erneut veröffentlichen < ***your environment name*** >. Der Assistent Erneut in AWS Elastic Beanstalk veröffentlichen wird geöffnet.



2. Überprüfen Sie die Bereitstellungsdetails und klicken Sie dann auf Deploy (Bereitstellen).

> **Note**
>
> Wenn Sie Ihre Einstellungen ändern möchten, klicken Sie auf Abbrechen und verwenden Sie stattdessen den In AWS veröffentlichen-Assistenten. Detaillierte Anweisungen finden Sie unter Erstellen einer Elastic Beanstalk-Umgebung.

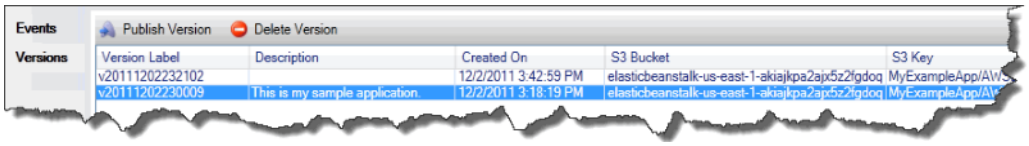
   Ihr aktualisiertes ASP.NET-Webprojekt wird als Webbereitstellungsdatei mit der neuen Versionsbezeichnung exportiert, in Amazon S3 hochgeladen und als neue Anwendungsversion in Elastic Beanstalk registriert. Mit der Bereitstellungsfunktion von Elastic Beanstalk wird die vorhandene Umgebung überwacht, bis sie mit dem neu bereitgestellten Code verfügbar ist. Auf der Registerkarte env: < ***environment name*** > sehen Sie den Status Ihrer Umgebung.

Sie können auch eine vorhandene Anwendung in einer vorhandenen Umgebung bereitstellen, wenn Sie zum Beispiel auf eine frühere Anwendungsversion zurücksetzen.

So stellen Sie eine Anwendungsversion in einer vorhandenen Umgebung bereit

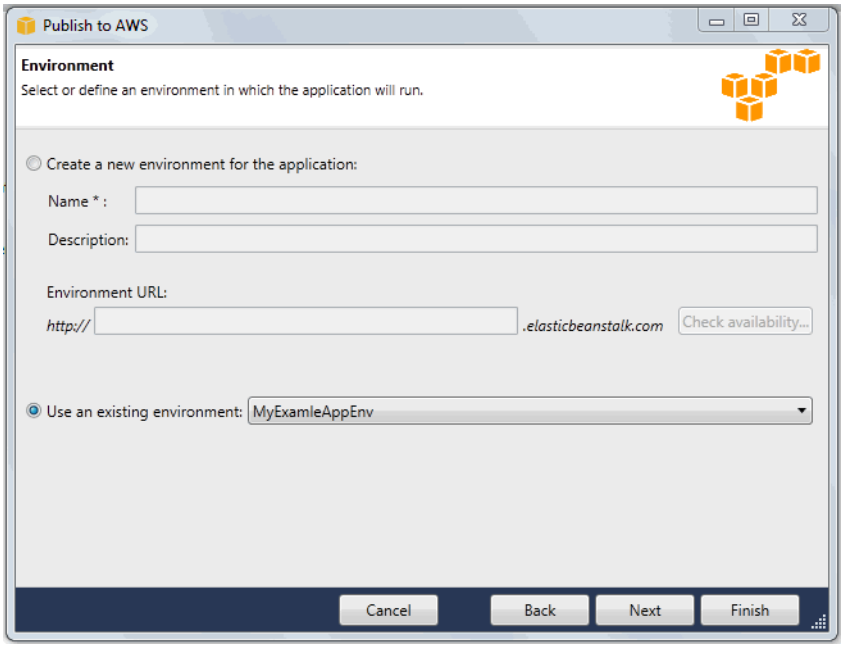
1. Klicken Sie mit der rechten Maustaste auf Ihre Elastic-Beanstalk-Anwendung, indem Sie den Elastic-Beanstalk-Knoten in AWS Explorer erweitern. Wählen Sie View Status (Status anzeigen) aus.

2. Klicken Sie auf der Registerkarte App: < ***application name*** > auf Versionen.

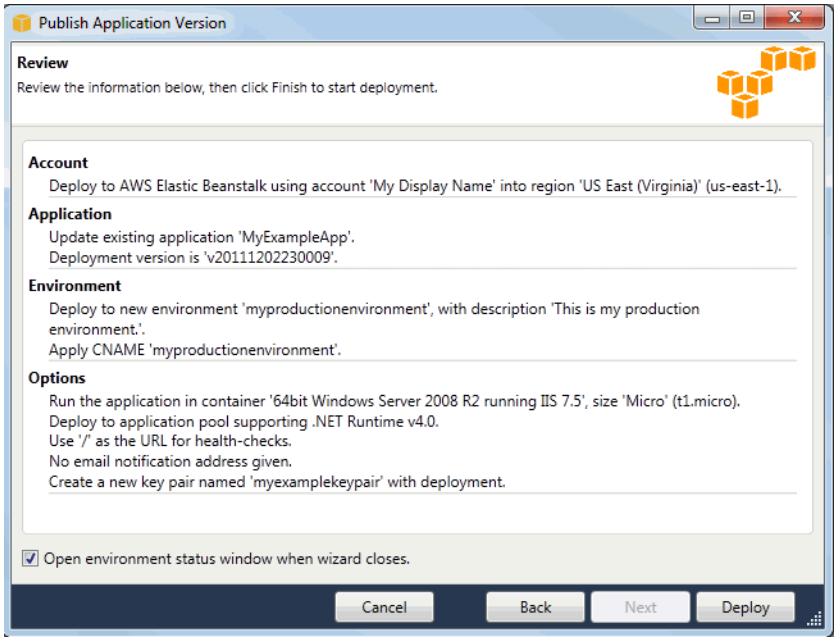


3. Klicken Sie auf die Anwendungsversion, die Sie bereitstellen möchten, und dann auf Publish Version (Version veröffentlichen).

4. Klicken Sie im Assistenten Publish Application Version (Anwendungsversion veröffentlichen) auf Next (Weiter).



5. Überprüfen Sie die Bereitstellungsoptionen und klicken Sie dann auf Deploy (Bereitstellen).



Ihr ASP.NET-Projekt wird als eine Webbereitstellungsdatei exportiert und in Amazon S3 hochgeladen. Mit der Bereitstellungsfunktion von Elastic Beanstalk wird die Umgebung überwacht, bis sie mit dem neu bereitgestellten Code verfügbar ist. Auf der Registerkarte env: < ***environment name*** > sehen Sie den Status Ihrer Umgebung.

Verwalten Ihrer Elastic Beanstalk-Anwendungsumgebungen

Mit dem AWS Toolkit for Visual Studio und der AWS Management Console können Sie die Bereitstellung und Konfiguration der von Ihren AWS Anwendungsumgebungen verwendeten Ressourcen ändern. Informationen zur Verwaltung Ihrer Anwendungsumgebungen mithilfe der AWS Management Console finden Sie unter. Elastic Beanstalk Beanstalk-Umgebungen verwalten In diesem Abschnitt werden die spezifischen Service-Einstellungen beschrieben, die Sie im AWS Toolkit for Visual Studio als Teil Ihrer Anwendungsumgebungs-Konfiguration bearbeiten können.

Ändern der Umgebungskonfigurationseinstellungen

Wenn Sie Ihre Anwendung bereitstellen, konfiguriert Elastic Beanstalk eine Reihe von AWS Cloud-Computing-Diensten. Sie können mit dem AWS Toolkit for Visual Studio steuern, wie diese einzelnen Dienste konfiguriert werden.

So bearbeiten Sie die Umgebungseinstellungen einer Anwendung

- Erweitern Sie den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten. Klicken Sie anschließend im AWS -Explorer mit der rechten Maustaste auf Ihre Elastic-Beanstalk-Umgebung. Wählen Sie View Status (Status anzeigen) aus.

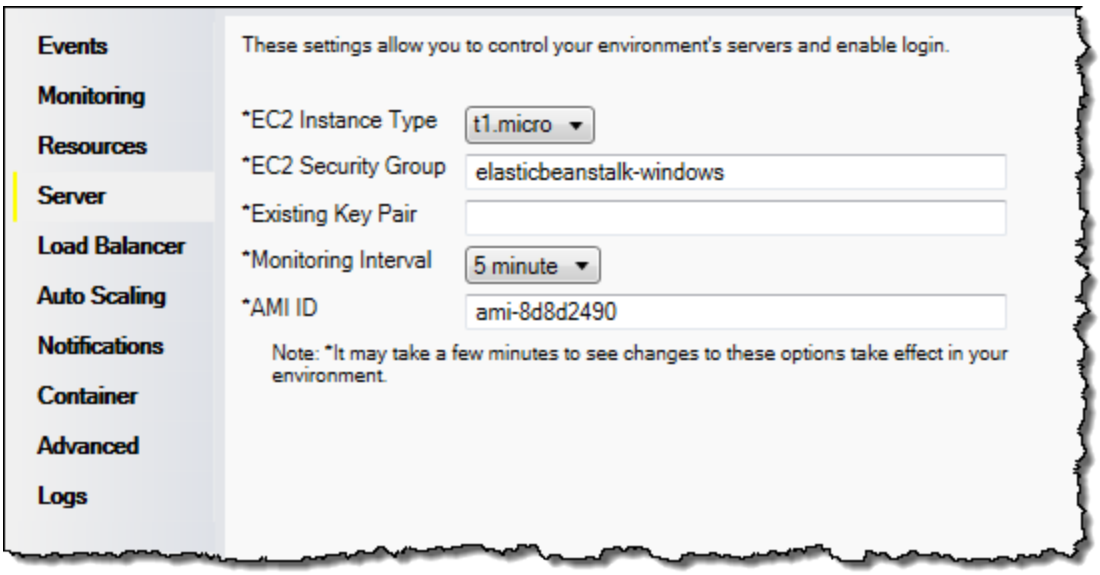
  Sie können nun Einstellungen für Folgendes konfigurieren:

  - herstellen
  - Load Balancing
  - Auto Scaling
  - Benachrichtigungen
  - Umgebungseigenschaften

Konfiguration von EC2 Serverinstanzen mit dem AWS Toolkit für Visual Studio

Amazon Elastic Compute Cloud (Amazon EC2) ist ein Webservice, mit dem Sie Server-Instances in den Rechenzentren von Amazon starten und verwalten. Sie können EC2 Amazon-Server-Instances jederzeit, so lange Sie benötigen und für jeden legalen Zweck verwenden. Instances sind in verschiedenen Größen und Konfigurationen verfügbar. Weitere Informationen finden Sie auf Amazon EC2.

Sie können die EC2 Amazon-Instance-Konfiguration der Elastic Beanstalk Beanstalk-Umgebung mit der Registerkarte Server innerhalb Ihrer Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.



EC2 Amazon-Instance-Typen

Instance type (Instance-Typ) zeigt die für Ihre Elastic Beanstalk-Anwendung verfügbaren Instance-Typen an. Ändern Sie den Instance-Typ, um einen Server mit den Merkmalen (einschließlich Speichergröße und CPU-Leistung) auszuwählen, die für Ihre Anwendung am besten geeignet sind. Anwendungen mit aufwändigen und lange dauernden Operationen erfordern mehr CPU oder Speicher.

Weitere Informationen zu den EC2 Amazon-Instance-Typen, die für Ihre Elastic Beanstalk Beanstalk-Anwendung verfügbar sind, finden Sie unter Instance-Typen im Amazon Elastic Compute Cloud-Benutzerhandbuch.

EC2 Amazon-Sicherheitsgruppen

Sie können den Zugriff auf Ihre Elastic Beanstalk Beanstalk-Anwendung mithilfe einer Amazon EC2 Security Group kontrollieren. Eine Sicherheitsgruppe definiert Firewall-Regeln für Ihre Instances. Diese Regeln legen fest, welcher eingehende Netzwerkverkehr an Ihre Instance übertragen werden soll. Anderer eingehender Datenverkehr wird verworfen. Sie können die Regeln für eine Gruppe jederzeit ändern. Die neuen Regeln werden automatisch für alle laufenden und künftigen Instances durchgesetzt.

Sie können Ihre EC2 Amazon-Sicherheitsgruppen mit der AWS Management Console oder mit dem AWS Toolkit for Visual Studio einrichten. Sie können angeben, welche EC2 Amazon-Sicherheitsgruppen den Zugriff auf Ihre Elastic Beanstalk Beanstalk-Anwendung kontrollieren, indem Sie die Namen einer oder mehrerer EC2 Amazon-Sicherheitsgruppen (durch Kommas getrennt) in das EC2 Textfeld Sicherheitsgruppen eingeben.

> **Note**
>
> Stellen Sie sicher, dass Port 80 (HTTP) über 0.0.0.0/0 als Quell-CIDR-Bereich zugreifbar ist, wenn Sie Zustandsprüfungen für Ihre Anwendung aktivieren möchten. Weitere Informationen zu Zustandsprüfungen finden Sie unter Zustandsprüfungen.

Um eine Sicherheitsgruppe mit dem Toolkit für Visual Studio zu erstellen AWS

1. Erweitern Sie in Visual Studio im AWS Explorer den EC2Amazon-Knoten, und doppelklicken Sie dann auf Sicherheitsgruppen.
2. Klicken Sie auf Create Security Group (Sicherheitsgruppe erstellen) und geben Sie einen Namen und eine Beschreibung für Ihre Sicherheitsgruppe ein.

3. Klicken Sie auf OK.

Weitere Informationen zu Amazon EC2 Security Groups finden Sie unter Using Security Groups im Amazon Elastic Compute Cloud-Benutzerhandbuch.

EC2 Amazon-Schlüsselpaare

Sie können sich mit einem Amazon-Schlüsselpaar sicher bei den EC2 Amazon-Instances anmelden, die für Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt wurden. EC2

> **Important**
>
> Sie müssen ein EC2 Amazon-Schlüsselpaar erstellen und Ihre von Elastic Beanstalk bereitgestellten EC2 Amazon-Instances so konfigurieren, dass sie das EC2 Amazon-Schlüsselpaar verwenden, bevor Sie auf Ihre von Elastic Beanstalk bereitgestellten Amazon-Instances zugreifen können. EC2 Sie können Ihr key pair mit dem Publish to AWS Wizard im AWS Toolkit for Visual Studio erstellen, wenn Sie Ihre Anwendung auf Elastic Beanstalk bereitstellen. Wenn Sie zusätzliche Schlüsselpaare mit dem Toolkit erstellen möchten, führen Sie die folgenden Schritte aus. Alternativ können Sie Ihre EC2 Amazon-Schlüsselpaare über die AWS Management Console einrichten. Anweisungen zum Erstellen eines key pair für Amazon finden Sie im Amazon EC2 Elastic Compute Cloud Getting Started Guide.

Im Textfeld Bestehendes key pair können Sie den Namen eines EC2 Amazon-Schlüsselpaars angeben, mit dem Sie sich sicher bei den EC2 Amazon-Instances anmelden können, auf denen Ihre Elastic Beanstalk Beanstalk-Anwendung ausgeführt wird.

Um den Namen eines EC2 Amazon-Schlüsselpaars anzugeben

1. Erweitern Sie den EC2Amazon-Knoten und doppelklicken Sie auf Schlüsselpaare.
2. Klicken Sie auf Create Key Pair (Schlüsselpaar erstellen) und geben Sie den Namen des Schlüsselpaares ein.
3. Klicken Sie auf OK.

Weitere Informationen zu EC2 Amazon-Schlüsselpaaren finden Sie unter Using Amazon EC2 Credentials im Amazon Elastic Compute Cloud-Benutzerhandbuch. Weitere Informationen zum Herstellen einer Verbindung zu EC2 Amazon-Instances finden Sie unterAuflisten von Server-Instances/Verbinden mit Server-Instances.

Überwachungsintervall

Standardmäßig sind nur grundlegende CloudWatch Amazon-Metriken aktiviert. Sie geben Daten in 5-Minuten-Intervallen zurück. Sie können detailliertere CloudWatch Ein-Minuten-Metriken aktivieren, indem Sie im Bereich Server auf der Registerkarte Konfiguration für Ihre Umgebung unter 1 Minute für das Überwachungsintervall auswählen. AWS Toolkit for Eclipse

> **Note**
>
> CloudWatch Amazon-Servicegebühren können für Metriken im Abstand von einer Minute anfallen. Weitere Informationen erhalten Sie CloudWatchbei Amazon.

Benutzerdefinierte AMI-ID

Sie können das Standard-AMI, das für Ihre EC2 Amazon-Instances verwendet wird, durch Ihr eigenes benutzerdefiniertes AMI überschreiben, indem Sie die ID Ihres benutzerdefinierten AMI in das Feld Benutzerdefinierte AMI-ID im Abschnitt Server der Registerkarte Konfiguration für Ihre Umgebung in der eingeben AWS Toolkit for Eclipse.

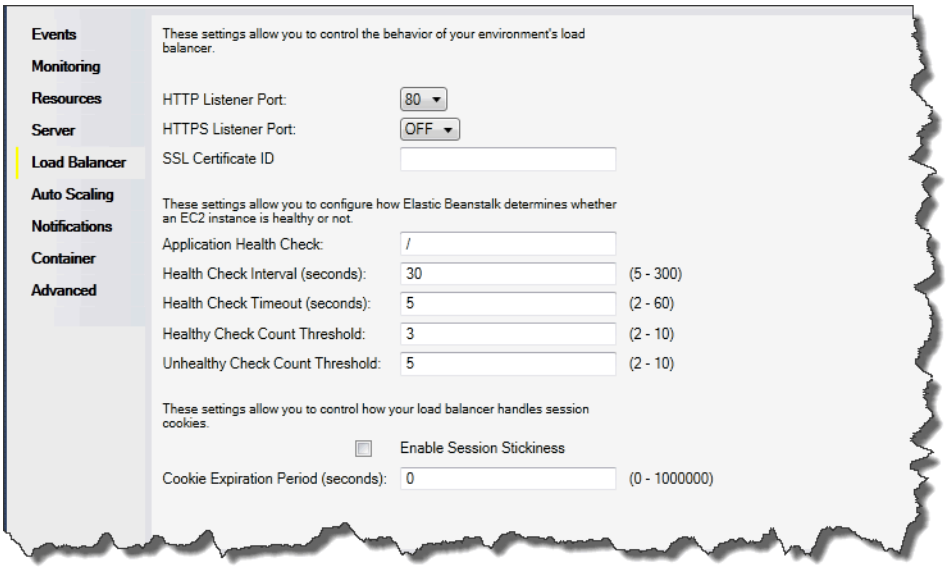
> **Important**
>
> Die Verwendung eines eigenen AMI ist eine Aufgabe für Fortgeschrittene und sollte mit Bedacht ausgeführt werden. Wenn Sie ein benutzerdefiniertes AMI benötigen, empfehlen wir, das standardmäßige Elastic Beanstalk-AMI als Ausgangspunkt zu nehmen und es zu ändern. Um als fehlerfrei angesehen zu werden, geht Elastic Beanstalk davon aus, dass EC2 Amazon-Instances eine Reihe von Anforderungen erfüllen, einschließlich eines laufenden Host-Managers. Werden diese Anforderungen nicht erfüllt, funktioniert Ihre Umgebung möglichweise nicht ordnungsgemäß.

Konfiguration von Elastic Load Balancing mit dem AWS Toolkit für Visual Studio

Elastic Load Balancing ist ein Amazon-Webservice, der Ihnen hilft, die Verfügbarkeit und Skalierbarkeit Ihrer Anwendung zu verbessern. Dieser Service macht es Ihnen leicht, Anwendungslasten auf zwei oder mehr EC2 Amazon-Instances zu verteilen. Elastic Load Balancing ermöglicht die Verfügbarkeit durch Redundanz und unterstützt wachsenden Datenverkehr für Ihre Anwendung.

Mit Elastic Load Balancer können Sie den eingehenden Datenverkehr der Anwendung automatisch auf sämtliche Instances verteilen, die Sie ausführen. Mit dem Service ist es auch ganz einfach, neue Instances hinzuzufügen, wenn Sie die Kapazität Ihrer Anwendung erhöhen.

Elastic Beanstalk stellt Elastic Load Balancing automatisch bereit, wenn Sie eine Anwendung bereitstellen. Sie können die EC2 Amazon-Instance-Konfiguration der Elastic Beanstalk Beanstalk-Umgebung mit der Registerkarte Load Balancer innerhalb Ihrer Anwendungsumgebung in AWS Toolkit for Visual Studio bearbeiten.



In den folgenden Abschnitten werden die Elastic Load Balancing-Parameter beschrieben, die Sie für Ihre Anwendung konfigurieren können.

Ports

Der Load Balancer, der für die Bearbeitung von Anfragen für Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt wurde, sendet Anfragen an die EC2 Amazon-Instances, auf denen Ihre Anwendung ausgeführt wird. Der bereitgestellte Load Balancer kann Anfragen an HTTP- und HTTPS-Ports abhören und Anfragen an die EC2 Amazon-Instances in Ihrer AWS Elastic Beanstalk Anwendung weiterleiten. Standardmäßig verarbeitet der Load Balancer Anfragen auf dem HTTP-Port. Mindestens einer der Ports (HTTP oder HTTPS) muss aktiviert werden.



> **Important**
>
> Stellen Sie sicher, dass der von Ihnen angegebene Port nicht gesperrt ist. Andernfalls können die Benutzer keine Verbindung zu Ihrer Elastic Beanstalk-Anwendung herstellen.

Steuern des HTTP-Ports

Um den HTTP-Port auszuschalten, wählen Sie OFF (AUS) für HTTP Listener Port aus. Zum Aktivieren des HTTP-Ports wählen Sie einen HTTP-Port (z. B. 80) aus der Liste aus.

> **Note**
>
> Für den Zugriff auf Ihre Umgebung mit einem anderen als dem Standardport 80, z. B. Port 8080, fügen Sie dem vorhandenen Load Balancer einen Listener hinzu und konfigurieren Sie den neuen Listener für die Überwachung des entsprechenden Ports.
> Geben Sie beispielsweise AWS CLI für Classic Load Balancer den folgenden Befehl ein und *LOAD_BALANCER_NAME* ersetzen Sie ihn durch den Namen Ihres Load Balancers für Elastic Beanstalk.
>
> ```
> aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
>  --listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
>  InstancePort=80"
> ```
>
> Geben Sie beispielsweise AWS CLI für Application Load Balancers den folgenden Befehl ein und *LOAD_BALANCER_ARN* ersetzen Sie ihn durch den ARN Ihres Load Balancers für Elastic Beanstalk.
>
> ```
> aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
>  --port 8080
> ```
>
> Wenn Elastic Beanstalk Ihre Umgebung überwachen soll, entfernen Sie den Listener für Port 80 nicht.

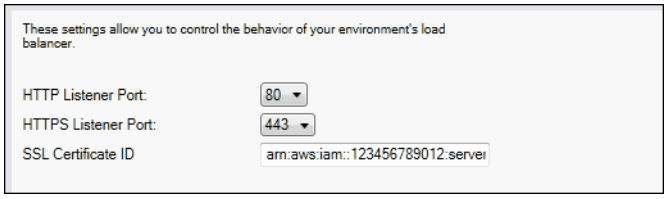
Steuern des HTTPS-Ports

Elastic Load Balancing unterstützt das HTTPS/TLS Protokoll, um die Verschlüsselung des Datenverkehrs für Client-Verbindungen zum Load Balancer zu ermöglichen. Verbindungen vom Load

Balancer zu den EC2 Instances verwenden Klartextverschlüsselung. Standardmäßig ist der HTTPS-Port deaktiviert.

So aktivieren Sie den HTTPS-Port

1. Erstellen Sie ein neues Zertifikat mit AWS Certificate Manager (ACM) oder laden Sie ein Zertifikat und einen Schlüssel zu AWS Identity and Access Management (IAM) hoch. Weitere Informationen zum Anfordern eines ACM-Zertifikats finden Sie unter Zertifikat anfordern im AWS Certificate Manager -Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in ACM finden Sie unter Importieren von Zertifikaten im AWS Certificate Manager -Benutzerhandbuch. Wenn ACM in Ihrer Region nicht verfügbar ist, verwenden Sie AWS Identity and Access Management (IAM), um ein Zertifikat eines Drittanbieters hochzuladen. Die ACM- und IAM-Services speichern das Zertifikat und stellen einen Amazon-Ressourcennamen (ARN) für das SSL-Zertifikat bereit. Weitere Informationen zum Erstellen und Hochladen von Zertifikaten in IAM finden Sie unter Arbeiten mit Serverzertifikaten im IAM-Benutzerhandbuch.
2. Geben Sie den HTTPS-Port an, indem Sie einen Port für HTTPS Listener Port auswählen.



3. Geben Sie für SSL Certificate ID (SSL-Zertifikat-ID) den Amazon-Ressourcennamen (ARN) Ihres SSL-Zertifikats ein. Beispiel: **`arn:aws:iam::123456789012:server-certificate/abc/certs/build`** oder **`arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`**. Verwenden Sie das SSL-Zertifikat, das Sie in Schritt 1 erstellt oder hochgeladen haben.

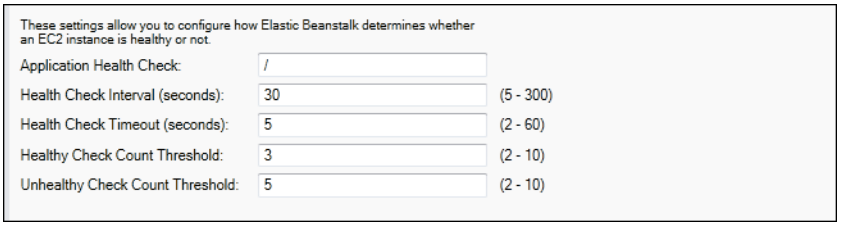
Um den HTTPS-Port auszuschalten, wählen Sie OFF (AUS) für HTTPS Listener Port.

Zustandsprüfungen

Die Definition der Zustandsprüfung enthält einen URL, die für den Instance-Zustand abgefragt wird. Standardmäßig nutzt Elastic Beanstalk TCP:80 für Nonlegacy-Container und HTTP:80 für Legacy-Container. Sie können die Standard-URL überschreiben, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/default.aspx`). Dazu geben Sie diese im Feld Application Health Check URL (Zustandsprüfungs-URL für die Anwendung) ein. Wenn Sie die Standard-URL überschreiben, verwendet Elastic Beanstalk HTTP zum Abfragen der

Ressource. Wenn Sie überprüfen möchten, ob Sie einen Legacy-Container verwenden, erhalten Sie Informationen unter the section called “Warum sind einige Plattformversionen als Legacy gekennzeichnet?”

Sie können die Einstellungen für die Integritätsprüfung im Bereich EC2 Instance Health Check im Bereich Load Balancing steuern.



Die Definition der Zustandsprüfung enthält einen URL, die für den Instance-Zustand abgefragt wird. Überschreiben Sie die Standard-URL, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/index.jsp`). Dazu geben Sie diese im Feld Application Health Check URL (Zustandsprüfungs-URL für die Anwendung) ein.

In der folgenden Liste werden die Zustandsprüfungs-Parameter beschrieben, die Sie für Ihre Anwendung festlegen können.

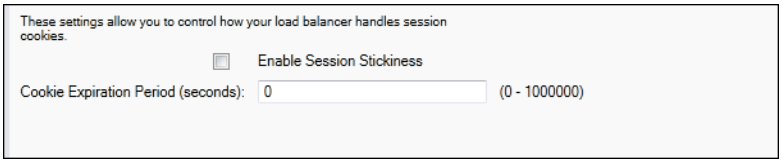
- Geben Sie für Health Check Interval (Sekunden) die Anzahl der Sekunden ein, die Elastic Load Balancing zwischen den Zustandsprüfungen für die EC2 Amazon-Instances Ihrer Anwendung wartet.
- Für Health Check Timeout (seconds) (Zustandsprüfungs-Timeout (Sekunden)) geben Sie die Anzahl der Sekunden ein, die Elastic Load Balancing auf eine Antwort wartet, bevor die Instance als nicht reagierend eingestuft wird.
- Für Healthy Check Count Threshold (Schwellenwert Anzahl erfolgreicher Zustandsprüfungen) und Unhealthy Check Count Threshold (Schwellenwert Anzahl fehlgeschlagener Zustandsprüfungen) geben Sie die Anzahl der aufeinanderfolgenden erfolgreichen oder nicht erfolgreichen URL-Proben ein, bevor Elastic Load Balancing den Status des Instance-Zustands ändert. Beispiel: Die Angabe von **5** für Unhealthy Check Count Threshold (Schwellenwert Anzahl fehlgeschlagener Zustandsprüfungen) bedeutet, dass die URL fünfmal hintereinander eine Fehlermeldung oder eine Zeitüberschreitung zurückgeben müssten, bevor Elastic Load Balancing die Zustandsprüfung als fehlgeschlagen einstufen würde.

Sitzungen

Standardmäßig leitet ein Load Balancer jede Anforderung getrennt an die Server-Instance mit der kleinsten Last weiter. Im Vergleich dazu bindet eine Sticky Session eine Benutzersitzung an eine bestimmte Server-Instance, sodass alle vom Benutzer während der Sitzung gesendeten Anforderungen an die gleiche Server-Instance übermittelt werden.

Elastic Beanstalk nutzt von Load Balancer generierte HTTP-Cookies, wenn Sticky Sessions für eine Anwendung aktiviert sind. Der Load Balancer verwendet ein vom Load-Balancer generiertes spezielles Cookie, um die Anwendungs-Instance für jede Anforderung nachzuverfolgen. Wenn der Load Balancer eine Anforderung empfängt, prüft er zunächst, ob dieses Cookie in der Anforderung vorhanden ist. Wenn ja, wird die Anforderung an die im Cookie angegebene Anwendungs-Instance gesendet. Wenn kein Cookie vorhanden ist, wählt der Load Balancer eine Anwendungs-Instance basierend auf dem vorhandenen Load Balancing-Algorithmus aus. Ein Cookie wird in die Antwort eingefügt, um nachfolgende Anforderungen von demselben Benutzer an diese Anwendungs-Instance zu binden. Mit der Konfiguration der Richtlinie wird ein Cookie-Ablauf definiert, der die Dauer der Gültigkeit für jedes Cookie festlegt.

Sie können über den Abschnitt Sessions (Sitzungen) auf der Registerkarte Load Balancer angeben, ob der Load Balancer für Ihre Anwendung "Sticky Sessions" erlaubt.



Weitere Informationen zu Elastic Load Balancing finden Sie im Elastic Load Balancing-Entwicklerhandbuch.

Konfiguration von Auto Scaling mit dem AWS Toolkit für Visual Studio

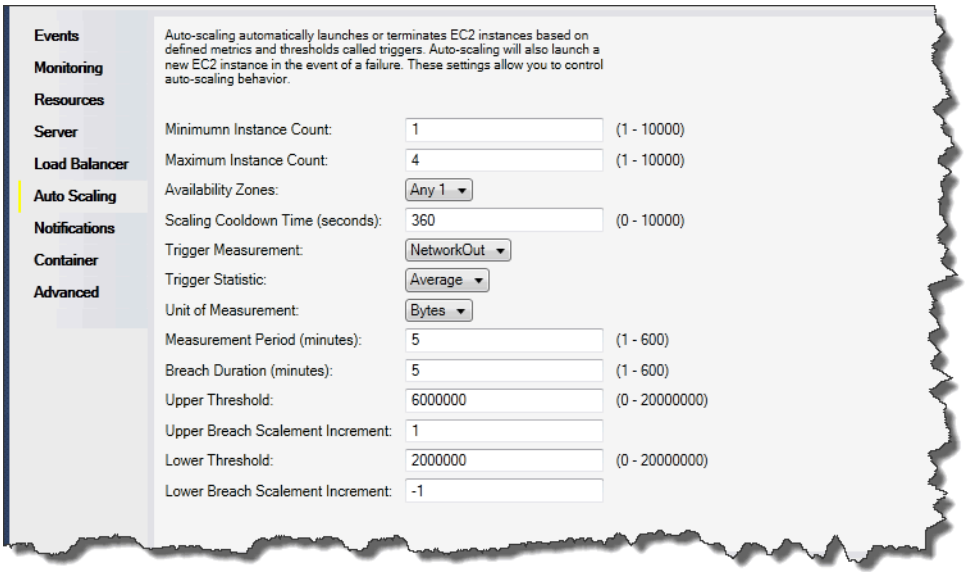
Amazon EC2 Auto Scaling ist ein Amazon-Webservice, der EC2 Amazon-Instances auf der Grundlage benutzerdefinierter Trigger automatisch startet oder beendet. Benutzer können Auto Scaling-Gruppen einrichten und diesen Gruppen Auslöser zuweisen, um Datenverarbeitungsressourcen automatisch basierend auf Metriken, wie Bandbreitennutzung oder CPU-Auslastung, zu skalieren. Amazon EC2 Auto Scaling arbeitet mit Amazon zusammen CloudWatch , um Metriken für die Server-Instances abzurufen, auf denen Ihre Anwendung ausgeführt wird.

Mit Amazon EC2 Auto Scaling können Sie eine Gruppe von EC2 Amazon-Instances nehmen und verschiedene Parameter festlegen, sodass die Anzahl dieser Gruppen automatisch erhöht oder

verringert wird. Amazon EC2 Auto Scaling kann EC2 Amazon-Instances zu dieser Gruppe hinzufügen oder daraus entfernen, damit Sie problemlos mit Verkehrsänderungen in Ihrer Anwendung umgehen können.

Amazon EC2 Auto Scaling überwacht auch den Zustand jeder EC2 Amazon-Instance, die es startet. Wenn eine Instance unerwartet beendet wird, erkennt Amazon EC2 Auto Scaling die Kündigung und startet eine Ersatz-Instance. Mit dieser Funktion können Sie automatisch eine feste, gewünschte Anzahl von EC2 Amazon-Instances verwalten.

Elastic Beanstalk stellt Amazon EC2 Auto Scaling für Ihre Anwendung bereit. Sie können die EC2 Amazon-Instance-Konfiguration der Elastic Beanstalk Beanstalk-Umgebung mit der Registerkarte Auto Scaling innerhalb Ihrer Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.

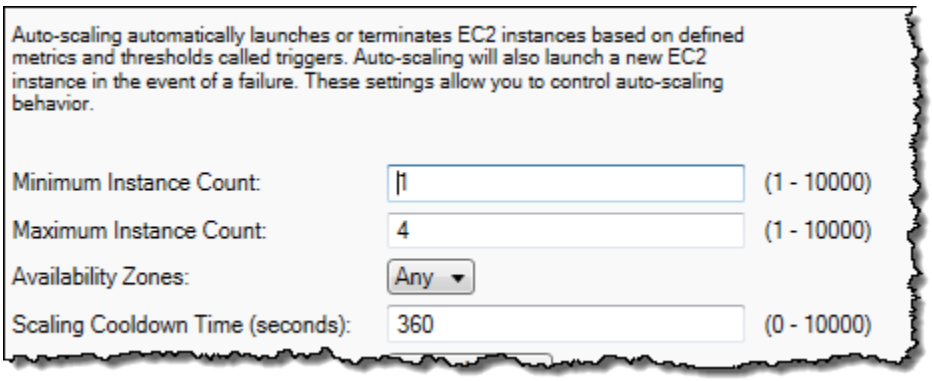


Im folgenden Abschnitt wird beschrieben, wie Sie Auto Scaling-Parameter für die Anwendung konfigurieren.

Starten der Konfiguration

Sie können die Startkonfiguration bearbeiten, um zu steuern, wie Ihre Elastic Beanstalk Beanstalk-Anwendung Amazon EC2 Auto Scaling Scaling-Ressourcen bereitstellt.

In den Feldern Minimum Instance Count (Minimale Instance-Anzahl) and Maximum Instance Count (Maximale Instance-Anzahl) geben Sie die minimale sowie die maximale Größe der Auto Scaling-Gruppe an, die von der Elastic Beanstalk-Anwendung genutzt wird.



> **Note**
>
> Um eine feste Anzahl von EC2 Amazon-Instances beizubehalten, setzen Sie Minimum Instance Count und Maximum Instance Count auf denselben Wert.

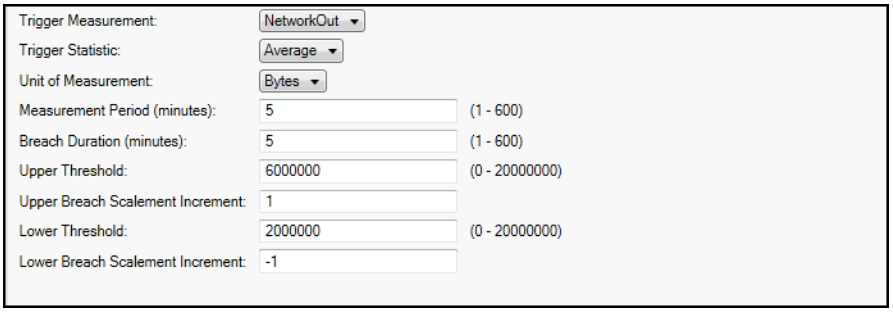
Im Feld Availability Zones können Sie die Anzahl der Availability Zones angeben, in denen sich Ihre EC2 Amazon-Instances befinden sollen. Es ist wichtig, diese Zahl festzulegen, wenn Sie fehlertolerante Anwendungen erstellen möchten. Falls eine Availability Zone ausfällt, werden die Instances in den anderen Availability Zones weiterhin ausgeführt.

> **Note**
>
> Derzeit ist es nicht möglich anzugeben, in welcher Availability Zone sich Ihre Instance befinden wird.

Auslöser

Ein Trigger ist ein Amazon EC2 Auto Scaling Scaling-Mechanismus, den Sie so einrichten, dass er dem System mitteilt, wann Sie die Anzahl der Instances erhöhen (nach oben skalieren) und wann Sie die Anzahl der Instances verringern (skalieren) möchten. Sie können Trigger so konfigurieren, dass sie bei jeder auf Amazon veröffentlichten Metrik ausgelöst werden CloudWatch, z. B. bei der CPU-Auslastung, und feststellen, ob die von Ihnen angegebenen Bedingungen erfüllt sind. Wird der obere oder untere Schwellenwert der Metrikbedingungen für einen bestimmten Zeitraum überschritten, startet der Auslöser den anhaltenden Prozess der Skalierung.

Mit dem AWS Toolkit for Visual Studio können Sie einen Skalierungs-Trigger für Ihre Elastic Beanstalk Beanstalk-Anwendung definieren.



Amazon EC2 Auto Scaling-Trigger funktionieren, indem sie eine bestimmte CloudWatch Amazon-Metrik für eine Instance beobachten. Zu den Auslösern zählen CPU-Auslastung, Netzwerkdatenverkehr und Datenträgeraktivität. Mit der Einstellung Trigger Measurement (Auslösermessung) wählen Sie eine Metrik für den Auslöser aus.

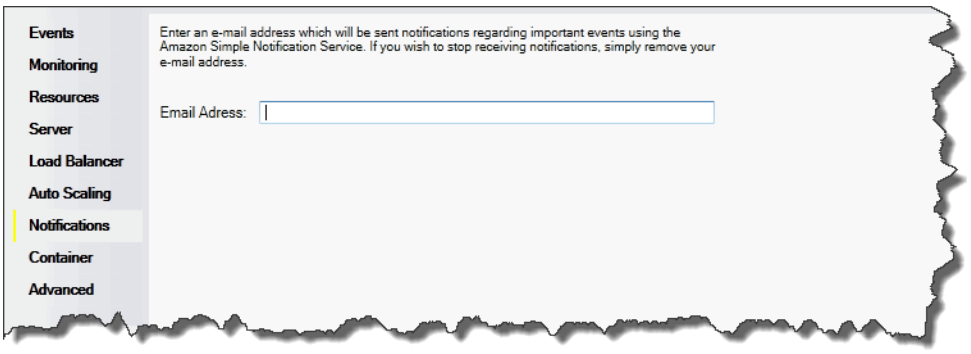
In der folgenden Liste werden die Trigger-Parameter beschrieben, die Sie mit der AWS Management Console konfigurieren können.

- Sie können angeben, welche Statistik der Auslöser verwenden soll. Sie können Minimum, Maximum, Sum (Summe) oder Average (Durchschnitt) für Trigger Statistic (Auslöserstatistik) auswählen.
- Für Unit of Measurement (Maßeinheit) geben Sie die Maßeinheit für den Auslöser an.
- Der Wert im Feld Messzeitraum gibt an, wie oft Amazon die Metriken für Ihren Trigger CloudWatch misst. Mit Breach Duration (Verletzungsdauer) wird der Zeitraum angegeben, für den der Grenzwert einer Metrik überschritten werden kann (wie unter Upper Threshold (Oberer Schwellenwert) und Lower Threshold (Unterer Schwellenwert) definiert), bevor der Auslöser ausgelöst wird.
- Geben Sie für Upper Breach Scale Increment und Lower Breach Scale Increment an, wie viele EC2 Amazon-Instances hinzugefügt oder entfernt werden sollen, wenn eine Skalierungsaktivität ausgeführt wird.

Weitere Informationen zu Amazon EC2 Auto Scaling finden Sie im Abschnitt Amazon EC2 Auto Scaling in der Amazon Elastic Compute Cloud-Dokumentation.

Konfiguration von Benachrichtigungen mit dem AWS Toolkit für Visual Studio

Elastic Beanstalk verwendet den Amazon Simple Notification Service (Amazon SNS) für Benachrichtigungen zu wichtigen Ereignissen in Ihrer Anwendung. Um Amazon SNS-Benachrichtigungen zu aktivieren, geben Sie einfach Ihre E-Mail-Adresse in das Feld Email Address ein. Um diese Benachrichtigungen zu deaktivieren, entfernen Sie Ihre E-Mail-Adresse aus dem Feld.



Konfiguration von.NET-Containern mit dem AWS Toolkit für Visual Studio

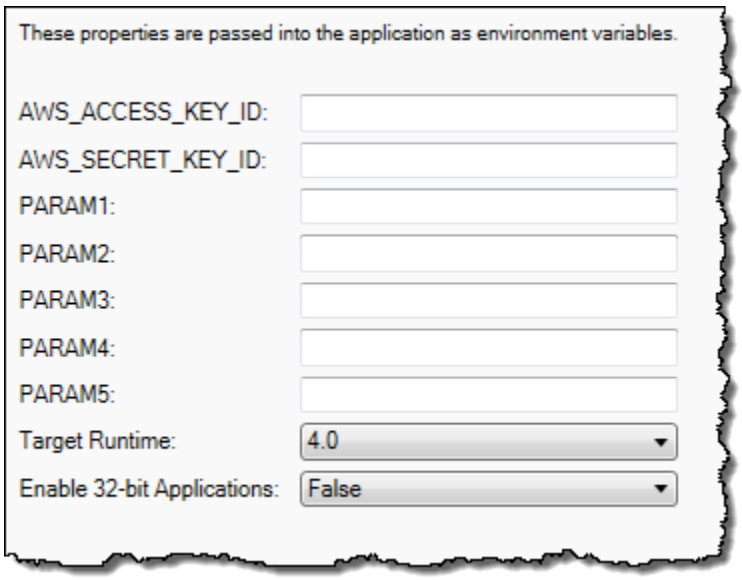
Im Bereich Container/.NET-Optionen können Sie das Verhalten Ihrer EC2 Amazon-Instances optimieren und die Amazon S3-Protokollrotation aktivieren oder deaktivieren. Sie können das AWS Toolkit for Visual Studio verwenden, um Ihre Containerinformationen zu konfigurieren.

> **Note**
>
> Sie können Ihre Konfigurationseinstellungen ändern, ohne eine Ausfallzeit in Kauf nehmen zu müssen, indem Sie den CNAME für Ihre Umgebungen austauschen. Weitere Informationen finden Sie unter Blau/Grün-Bereitstellungen mit Elastic Beanstalk.

Wenn Sie möchten, können Sie die Anzahl der Parameter erhöhen. Weitere Informationen über das Erhöhen der Parameteranzahl finden Sie unter Optionseinstellungen.

So greifen Sie auf den Container/.NET-Optionsbereich für Ihre Elastic Beanstalk-Anwendung zu

1. Erweitern Sie in AWS Toolkit for Visual Studio den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten.
2. Klicken Sie im AWS Explorer doppelt auf Ihre Elastic-Beanstalk-Umgebung.
3. Klicken Sie unten im Bereich Overview (Übersicht) auf die Registerkarte Configuration (Konfiguration).
4. Konfigurieren Sie im Bereich Container die Containeroptionen.

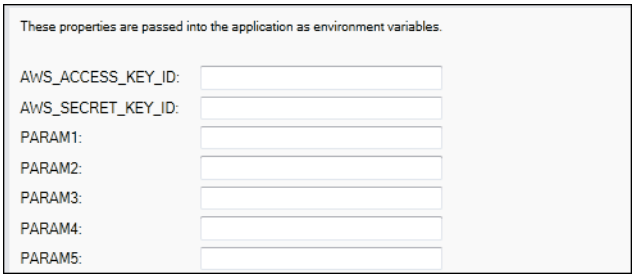


.NET-Container-Optionen

Sie können die Version von .NET-Framework für Ihre Anwendung wählen. Wählen Sie entweder 2.0 oder 4.0 für Target runtime (Ziellaufzeit) aus. Wählen Sie Enable 32-bit Applications (32-Bit-Anwendungen aktivieren) aus, wenn Sie 32-Bit-Anwendungen aktivieren möchten.

Anwendungseinstellungen

Unter Application Settings (Anwendungseinstellungen) geben Sie Umgebungsvariablen an, die aus dem Anwendungscode ausgelesen werden.



Verwalten von Konten

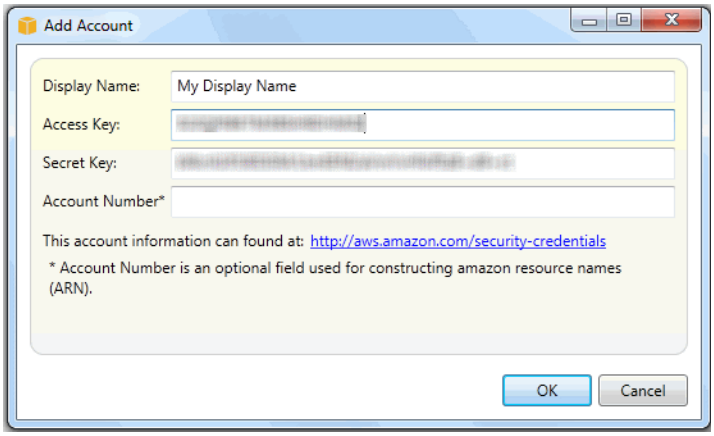
Wenn Sie verschiedene AWS Konten einrichten möchten, um verschiedene Aufgaben wie Testen, Staging und Produktion auszuführen, können Sie Konten mit dem AWS Toolkit for Visual Studio hinzufügen, bearbeiten und löschen.

So verwalten Sie mehrere Konten

1. Klicken Sie in Visual Studio im Menü Anzeigen auf AWS -Explorer.
2. Neben der Liste Account (Konto) klicken Sie auf die Schaltfläche Add Account (Konto hinzufügen).



Das Dialogfeld Add Account (Konto hinzufügen) wird angezeigt.



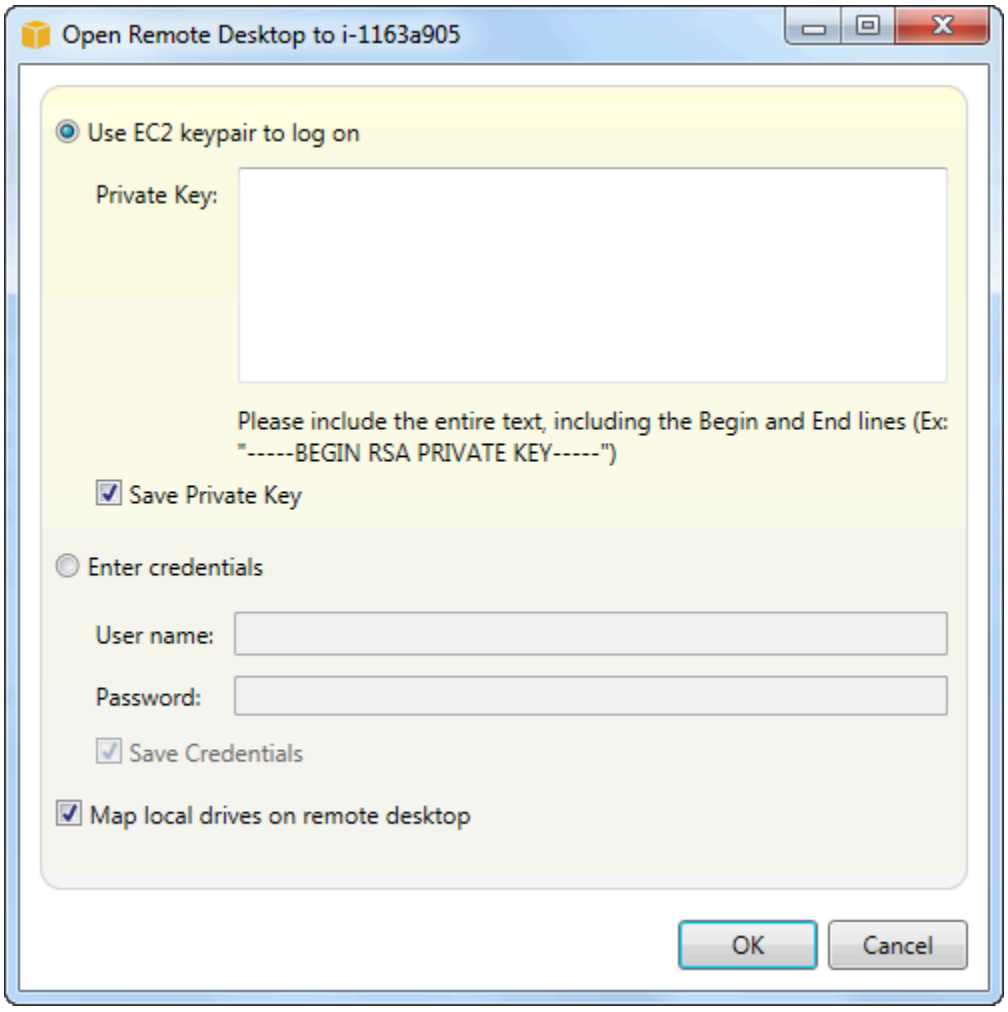
3. Geben Sie die erforderlichen Informationen ein.
4. Ihre Kontoinformationen werden nun auf der Registerkarte AWS -Explorer angezeigt. Wenn Sie eine Veröffentlichung in Elastic Beanstalk durchführen, können Sie auswählen, welches Konto Sie verwenden möchten.

Auflisten von Server-Instances/Verbinden mit Server-Instances

Sie können eine Liste der EC2 Amazon-Instances, auf denen Ihre Elastic Beanstalk Beanstalk-Anwendungsumgebung ausgeführt wird, über das AWS Toolkit for Visual Studio oder über die AWS Management Console anzeigen. Die Verbindung zu diesen Instances können Sie mit Remote Desktop Connection herstellen. Informationen zum Auflisten und Herstellen einer Verbindung zu Ihren Server-Instances mithilfe der AWS Management Console finden Sie unter. Auflisten von Server-Instances/Verbinden mit Server-Instances Im folgenden Abschnitt erfahren Sie, wie Sie Ihre Serverinstanzen mithilfe des AWS Toolkit for Visual Studio anzeigen und eine Verbindung zu ihnen herstellen.

So zeigen Sie EC2 Amazon-Instances für eine Umgebung an und stellen eine Verbindung zu ihnen her

1. Erweitern Sie in Visual Studio im AWS Explorer den EC2Amazon-Knoten und doppelklicken Sie auf Instances.
2. Klicken Sie in der Spalte Instance mit der rechten Maustaste auf die EC2 Instance-ID für die Amazon-Instance, die im Load Balancer Ihrer Anwendung ausgeführt wird, und wählen Sie im Kontextmenü Remote Desktop öffnen aus.



3. Wählen Sie EC2 Schlüsselpaar verwenden, um sich anzumelden, und fügen Sie den Inhalt Ihrer privaten Schlüsseldatei, die Sie zur Bereitstellung Ihrer Anwendung verwendet haben, in das Feld Privater Schlüssel ein. Alternativ geben Sie Ihren Benutzernamen und Ihr Passwort in die Textfelder User name (Benutzername) und Password (Passwort) ein.

> **Note**
>
> Wenn das Schlüsselpaar im Toolkit gespeichert ist, wird das Textfeld nicht angezeigt.

4. Klicken Sie auf OK.

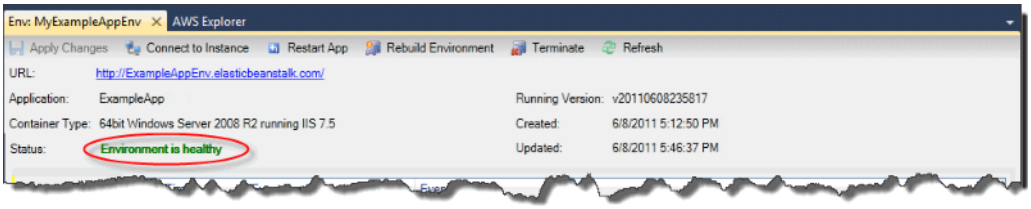
Überwachen des Anwendungsstatus

Bei Ausführung einer Produktionswebsite ist es wichtig zu wissen, dass die Anwendung verfügbar ist und auf Anforderungen reagiert. Für die Überwachung der Reaktionsfähigkeit der Anwendung können Sie die Elastic Beanstalk-Funktionen nutzen, mit denen Sie Anwendungsstatistiken überwachen und Warnungen erstellen können, die bei einer Überschreitung der Schwellenwerte ausgelöst werden.

Weitere Informationen über die Integritätsüberwachung von Elastic Beanstalk finden Sie unter Grundlegende Zustandsberichte.

Sie können auf Betriebsinformationen zu Ihrer Anwendung zugreifen, indem Sie entweder das AWS Toolkit for Visual Studio oder die AWS Management Console verwenden.

Das Toolkit zeigt den Zustand Ihrer Umgebung und den Anwendungsstatus im Feld Status an.



So überwachen Sie den Anwendungsstatus

1. Erweitern Sie im AWS Toolkit for Visual Studio im AWS Explorer den Elastic Beanstalk-Knoten und dann Ihren Anwendungsknoten.
2. Klicken Sie mit der rechten Maustaste auf Ihre Elastic Beanstalk-Umgebung und anschließend auf View Status (Status anzeigen).
3. Klicken Sie auf der Registerkarte der Anwendungsumgebung auf Monitoring (Überwachung).

   Der Bereich Monitoring (Überwachung) enthält eine Reihe von Diagrammen, die die Ressourcennutzung für Ihre bestimmte Anwendungsumgebung zeigen.

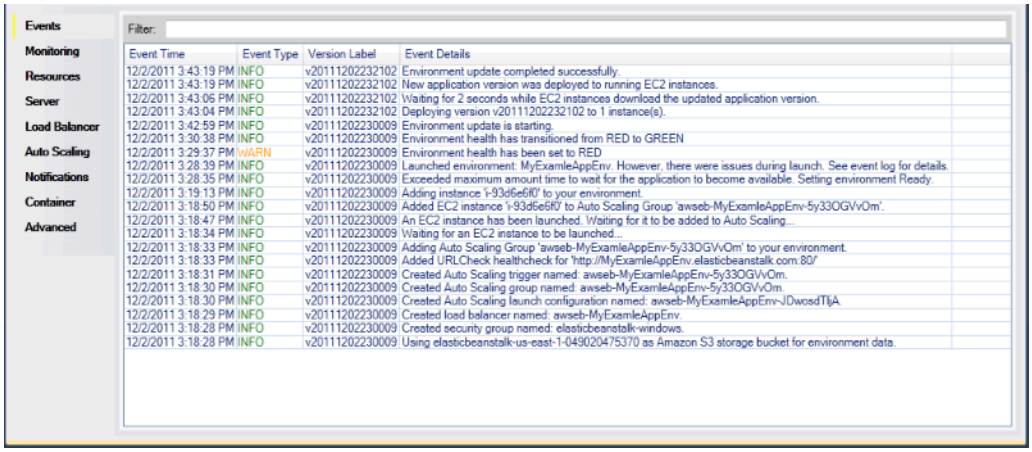


> **Note**
>
> Standardmäßig ist der Zeitraum auf der letzten Stunde gesetzt. Ändern Sie diese Einstellung, indem Sie in der Liste Time Range (Zeitraum) auf einen anderen Zeitraum klicken.

Sie können das AWS Toolkit for Visual Studio oder die AWS Management Console verwenden, um Ereignisse im Zusammenhang mit Ihrer Anwendung anzuzeigen.

So zeigen Sie Anwendungsereignisse an

1. Erweitern Sie im AWS Toolkit for Visual Studio im AWS Explorer den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten.
2. Klicken Sie in AWS -Explorer mit der rechten Maustaste auf Ihre Elastic-Beanstalk-Umgebung und anschließend auf Status anzeigen.
3. Klicken Sie in der Registerkarte der Anwendungsumgebung auf Events (Ereignisse).



Bereitstellung von Elastic Beanstalk Beanstalk-Anwendungen in.NET mithilfe von AWS Bereitstellungstools

AWS bietet zwei Befehlszeilentools für die Bereitstellung von.NET Core- und ASP.NET Core-Anwendungen auf Elastic Beanstalk:

- AWS .NET-Bereitstellungstool — Ein modernes Befehlszeilentool, das die Bereitstellung von.NET-Anwendungen für AWS Dienste, einschließlich Elastic Beanstalk, vereinfacht. Dieses Tool bietet ein interaktives Erlebnis und kann automatisch den besten AWS Service für Ihre Anwendung ermitteln. Weitere Informationen finden Sie AWS unter.NET-Bereitstellungstool.
- AWS Erweiterungen für.NET CLI — Eine Reihe von.NET-CLI-Erweiterungen, die Befehle für die Bereitstellung von.NET Core-Anwendungen für Elastic Beanstalk und andere AWS Dienste bereitstellen. Diese Erweiterungen lassen sich in den standardmäßigen .NET-CLI-Workflow integrieren. Weitere Informationen finden Sie unter AWS Erweiterungen für.NET CLI.

Migrieren Ihrer lokalen .NET-Anwendung zu Elastic Beanstalk

AWS Elastic Beanstalk bietet über die Elastic Beanstalk Command Line Interface (EB CLI) einen optimierten Migrationspfad für Ihre Windows-Anwendungen, die auf Internet Information Services (IIS) ausgeführt werden. Der eb migrate Befehl erkennt automatisch Ihre IIS-Sites, -Anwendungen und virtuellen Verzeichnisse, behält ihre Konfigurationen bei und stellt sie in der Cloud bereit. AWS

Diese integrierte Migrationsfunktion bietet einen einfacheren Ansatz, der die Komplexität und den Zeitaufwand reduziert, der normalerweise mit Cloud-Migrationen verbunden ist. Der Migrationsprozess trägt dazu bei, die Anwendungsfunktionalität und die Konfigurationsintegrität während der Umstellung auf aufrechtzuerhalten. AWS

Vollständige, detaillierte Anweisungen zur Migration Ihrer IIS-Anwendungen auf AWS Elastic Beanstalk finden Sie im IIS-Anwendungen migrieren Kapitel in diesem Handbuch.

Empfehlungen für ausrangierte Windows Server-Komponenten auf Elastic Beanstalk

Dieses Thema enthält Empfehlungen für den Fall, dass Ihre Anwendungen derzeit auf den veralteten Windows Server 2012 R2-Plattformzweigen ausgeführt werden. Es befasst sich auch mit der veralteten Unterstützung für die Protokollversionen TLS 1.0 und 1.1 auf unseren AWS Service-API-Endpunkten und den betroffenen Plattformzweigen.

Plattformzweige von Windows Server 2012 R2 außer Betrieb genommen

Elastic Beanstalk hat die Windows Server 2012 R2-Plattformzweige am 4. Dezember 2023 eingestellt und die mit diesen Plattformen AMIs verknüpften Zweige am 10. April 2024 privat gemacht. Diese Aktion verhindert das Starten von Instances in Ihren Windows Server 2012-Umgebungen, die das standardmäßige Beanstalk-AMI verwenden.

Wenn Sie Umgebungen haben, die auf ausgemusterten Windows-Plattformzweigen ausgeführt werden, empfehlen wir Ihnen, diese auf eine der folgenden Windows Server-Plattformen zu migrieren, die aktuell sind und vollständig unterstützt werden:

- Windows Server 2022 mit IIS 10.0 Version 2.x
- Windows Server 2019 mit IIS 10.0, Version 2.x

Vollständige Überlegungen zur Migration finden Sie unter Migration von früheren Hauptversionen der Windows Server-Plattform.

Weitere Informationen zu Plattform-Veraltung finden Sie unter Support-Richtlinie für die Elastic Beanstalk-Plattform.

> **Note**
>
> Wenn Sie nicht auf diese vollständig unterstützten Plattformen migrieren können, empfehlen wir, ein mit Windows Server 2012 R2 oder Windows Server 2012 R2 Core AMIs erstelltes benutzerdefiniertes Image AMIs als Basisimage zu verwenden, sofern Sie dies noch nicht getan haben. Detaillierte Anweisungen finden Sie unter Behalten des Zugriffs auf ein Amazon Machine Image (AMI) für eine außer Betrieb genommene Plattform. Wenden Sie sich an das AWS Support Center, wenn Sie während der Durchführung eines dieser Migrationsschritte temporären Zugriff auf ein AMI benötigen.

TLS 1.2 Kompatibilität

Am 31. Dezember 2023 wurde mit der vollständigen Durchsetzung von TLS 1.2 auf allen AWS API-Endpunkten AWS begonnen. Durch diese Aktion wurde die Möglichkeit, die TLS-Versionen 1.0 und 1.1 mit allen zu verwenden, aufgehoben. AWS APIs Diese Information wurde ursprünglich am 28. Juni 2022 übermittelt. Um das Risiko einer Beeinträchtigung der Verfügbarkeit zu vermeiden, sollten

Sie alle Umgebungen, in denen die hier genannten Plattformversionen ausgeführt werden, so bald wie möglich auf eine neuere Version aktualisieren, sofern Sie dies noch nicht getan haben.

Mögliche Auswirkungen

Versionen der Elastic Beanstalk-Plattformen, auf denen TLS v1.1 oder früher ausgeführt wird, sind betroffen. Diese Änderung wirkt sich auf Umgebungsaktionen aus, die unter anderem Folgendes umfassen: Konfigurationsbereitstellungen, Anwendungsbereitstellungen, auto Skalierung, Start neuer Umgebungen, Protokollrotation, erweiterte Gesundheitsberichte und Veröffentlichung von Anwendungsprotokollen im Amazon S3 S3-Bucket, der Ihren Anwendungen zugeordnet ist.

Betroffene Windows-Plattformversionen

Kunden mit Elastic-Beanstalk-Umgebungen auf der folgenden Plattformversion wird empfohlen, jede ihrer entsprechenden Umgebungen auf die Windows-Plattformversion 2.8.3 oder höher, veröffentlicht am 18. Februar 2022, zu aktualisieren.

- Windows Server 2019 – Plattformversion 2.8.2 oder frühere Versionen

Kunden mit Elastic-Beanstalk-Umgebungen auf den folgenden Plattformversionen wird empfohlen, jede ihrer entsprechenden Umgebungen auf die Windows-Plattformversion 2.10.7 oder höher, veröffentlicht am 28. Dezember 2022, zu aktualisieren.

- Windows Server 2016 – Plattformversion 2.10.6 oder frühere Versionen
- Windows Server 2012 — alle Plattformversionen; diese Plattform wurde am 4. Dezember 2023 eingestellt
- Windows Server 2008 — alle Plattformversionen; diese Plattform wurde am 28. Oktober 2019 eingestellt

Eine Liste der neuesten und unterstützten Versionen der Windows-Server-Plattform finden Sie unter Unterstützte Plattformen im AWS Elastic Beanstalk -Plattformen-Leitfaden.

Einzelheiten und bewährte Methoden zur Aktualisierung Ihrer Umgebung finden Sie unterAktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung.

Bereitstellen von.NET-Core-Anwendungen (Linux) mit Elastic Beanstalk

> **Schauen Sie sich das.NET on AWS Developer Center an**
>
> Haben Sie schon einmal in unserem .Net Developer Center vorbeigeschaut? Es ist unsere zentrale Anlaufstelle für alles, was mit .NET zu tun hat AWS.
> Weitere Informationen finden Sie unter.NET on AWS Developer Center.

Dieses Kapitel enthält Anweisungen zur Konfiguration und Bereitstellung Ihrer.NET-Core-Webanwendung (Linux) für AWS Elastic Beanstalk. Elastic Beanstalk macht es einfach, Ihren.NET-Core-Webanwendungen (Linux) mithilfe von Amazon Web Services bereitzustellen, zu verwalten und zu skalieren.

Sie können Ihre Anwendung in nur wenigen Minuten mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) oder der Elastic Beanstalk Beanstalk-Konsole bereitstellen. Nachdem Sie Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt haben, können Sie weiterhin die EB-CLI verwenden, um Ihre Anwendung und Umgebung zu verwalten, oder Sie können die Elastic Beanstalk Beanstalk-Konsole oder die verwenden. AWS CLI APIs

Folgen Sie den step-by-step Anweisungen unter, um eine ASP.NET Core Hello World-Webanwendung mit der EB CLI zu erstellen und bereitzustellen. QuickStart für .NET Core unter Linux

Themen

- QuickStart: Bereitstellen einer.NET-Core-on-Linux-Anwendung auf Elastic Beanstalk
- Einrichtung Ihrer Entwicklungsumgebung für.NET Core auf Linux für Elastic Beanstalk
- Verwenden des Elastic Beanstalk Beanstalk-.NET-Kerns auf einer Linux-Plattform
- The AWS Toolkit for Visual Studio - Arbeiten mit.Net Core auf Elastic Beanstalk
- Migration von .NET auf Windows Server-Plattform zur .NET Core auf Linux-Plattform auf Elastic Beanstalk

QuickStart: Bereitstellen einer.NET-Core-on-Linux-Anwendung auf Elastic Beanstalk

Dieses QuickStart Tutorial führt Sie durch den Prozess der Erstellung einer.NET Core-Anwendung unter Linux und deren Bereitstellung in einer AWS Elastic Beanstalk Umgebung.

> ⚠ Nicht für Produktionszwecke
>
> Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine.NET-Core-Anwendung unter Linux
- Schritt 2: Führen Sie Ihre Anwendung lokal aus
- Schritt 3: Stellen Sie Ihre.NET Core on Linux-Anwendung mit der EB CLI bereit
- Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus
- Schritt 5: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center - Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

EB-CLI

Dieses Tutorial verwendet die Befehlszeilenschnittstelle Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

.NET Core auf Linux

Wenn Sie das .NET SDK nicht auf Ihrem lokalen Computer installiert haben, können Sie es installieren, indem Sie auf der Website mit der .NET-Dokumentation auf den Link „.NET herunterladen“ klicken.

Überprüfen Sie Ihre .NET SDK-Installation, indem Sie den folgenden Befehl ausführen.

```
~$ dotnet --info
```

Schritt 1: Erstellen Sie eine.NET-Core-Anwendung unter Linux

Erstellen Sie ein Projektverzeichnis.

```
~$ mkdir eb-dotnetcore
~$ cd eb-dotnetcore
```

Erstellen Sie als Nächstes eine Hello World-Beispielanwendung, indem Sie die folgenden Befehle ausführen.

```
~/eb-dotnetcore$ dotnet new web --name HelloElasticBeanstalk
~/eb-dotnetcore$ cd HelloElasticBeanstalk
```

Schritt 2: Führen Sie Ihre Anwendung lokal aus

Führen Sie den folgenden Befehl aus, um Ihre Anwendung lokal auszuführen.

```
~/eb-dotnetcore/HelloElasticBeasntalk$ dotnet run
```

Die Ausgabe sollte in etwa wie der folgende Text aussehen.

```
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7294
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5052
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
```

> **Note**
>
> Der `dotnet` Befehl wählt zufällig einen Port aus, wenn die Anwendung lokal ausgeführt wird. In diesem Beispiel ist der Port 5052. Wenn Sie die Anwendung in Ihrer Elastic Beanstalk Beanstalk-Umgebung bereitstellen, wird die Anwendung auf Port 5000 ausgeführt.

Geben Sie die URL-Adresse `http://localhost:`*`port`* in Ihren Webbrowser ein. Für dieses spezielle Beispiel lautet der Befehl`http://localhost:5052`. Der Webbrowser sollte „Hello World!“ anzeigen.

Schritt 3: Stellen Sie Ihre.NET Core on Linux-Anwendung mit der EB CLI bereit

Führen Sie die folgenden Befehle aus, um eine Elastic Beanstalk Beanstalk-Umgebung für diese Anwendung zu erstellen.

Um eine Umgebung zu erstellen und Ihre Anwendung für.NET Core auf Linux bereitzustellen

1. Kompilieren und veröffentlichen Sie Ihre Anwendung in einem Ordner für die Bereitstellung in der Elastic Beanstalk Beanstalk-Umgebung, die Sie gerade erstellen möchten.

   ```
   ~$ cd eb-dotnetcore/HelloElasticBeanstalk
   ```

```
~/eb-dotnetcore/HelloElasticBeanstalk$ dotnet publish -o site
```

2. Navigieren Sie zu dem Site-Verzeichnis, in dem Sie Ihre App gerade veröffentlicht haben.

```
~/eb-dotnetcore/HelloElasticBeanstalk$ cd site
```

3. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

   Beachten Sie die folgenden Details zur Version der Plattformverzweigung, die Sie im Befehl angeben:

   - Ersetzen Sie *x.y.z* den folgenden Befehl durch die neueste Version der Plattformverzweigung.NET 6 auf AL2 023.
   - Die neueste Version von Platform Branch finden Sie unter .NET Core on Linux Supported AWS Elastic Beanstalk Platforms im Plattformhandbuch.
   - Ein Beispiel für einen Lösungsstapelnamen, der die Versionsnummer enthält, ist`64bit-amazon-linux-2023-v`**`3.1.1`**`-running-.net-6`. In diesem Beispiel ist die Zweigversion 3.1.1.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb init -p 64bit-amazon-linux-2023-
vx.y.z-running-.net-6 dotnetcore-tutorial --region us-east-2
Application dotnetcore-tutorial has been created.
```

   Dieser Befehl erstellt eine Anwendung mit dem Namen `dotnetcore-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der im Befehl angegebenen Plattformversion .NET Core auf Linux erstellt werden.

4. (Optional) Führen Sie den Vorgang eb init erneut aus, um ein Standardschlüsselpaar zu konfigurieren, sodass Sie SSH verwenden können, um eine Verbindung zu der EC2 Instance herzustellen, auf der Ihre Anwendung ausgeführt wird.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein Schlüsselpaar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

5. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erstellt automatisch eine Zip-Datei für Ihre Anwendung und startet sie auf Port 5000.

   to

   ```
   ~eb-dotnetcore/HelloElasticBeanstalk/site$ eb create dotnet-tutorial
   ```

   Es dauert etwa fünf Minuten, bis Elastic Beanstalk Ihre Umgebung erstellt hat.

Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, öffnen Sie Ihre Website mit. eb open

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb open
```

Herzlichen Glückwunsch! Sie haben eine.NET-Core-Anwendung auf Linux mit Elastic Beanstalk bereitgestellt! Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet.

Schritt 5: Bereinigen

Sie können Ihre Umgebung beenden, wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind. Elastic Beanstalk beendet alle AWS Ressourcen, die mit Ihrer Umgebung verknüpft sind.

Führen Sie den folgenden Befehl aus, um Ihre Elastic Beanstalk Beanstalk-Umgebung mit der EB-CLI zu beenden.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb terminate
```

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt. Es dient als einfache Beispielanwendung mit einer einzigen EC2 Instanz, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 Instanz — Eine EC2 virtuelle Maschine von Amazon, die so konfiguriert ist, dass sie Web-Apps auf der von Ihnen ausgewählten Plattform ausführt.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Die

Bereitstellung einer neuen Anwendungsversion ist sehr schnell, da dafür keine Bereitstellung oder Neustarts von Instances erforderlich sind. EC2 Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

Wenn Sie eine oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, .NET Core lokal auf Linux-Anwendungen zu entwickeln und auszuführen, finden Sie weitere Informationen unterEinrichtung Ihrer Entwicklungsumgebung für.NET Core auf Linux für Elastic Beanstalk.

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

Einrichtung Ihrer Entwicklungsumgebung für.NET Core auf Linux für Elastic Beanstalk

Dieses Thema enthält Anweisungen zum Einrichten einer Entwicklungsumgebung für .NET Core auf Linux, in der Sie Ihre Anwendung lokal testen können, bevor Sie sie bereitstellen AWS Elastic Beanstalk. Es verweist auch auf Websites, die Installationsanweisungen für nützliche Tools bereitstellen.

Sections

- Installation des .NET Core SDK
- Installieren einer IDE
- Installation des AWS Toolkit for Visual Studio

Installation des .NET Core SDK

Sie können das .NET Core SDK für die Entwicklung von Anwendungen verwenden, die auf Linux ausgeführt werden.

Sie können das .NET Core SDK von der Seite für .NET-Downloads zur anschließenden Installation herunterladen.

Installieren einer IDE

Integrierte Entwicklungsumgebungen (IDEs) bieten eine Reihe von Funktionen, die die Anwendungsentwicklung erleichtern. Wenn Sie noch nie eine IDE für die .NET-Entwicklung verwendet haben, können Sie mit Visual Studio Community erste Schritte unternehmen.

Sie können Visual Studio Community von der Seite Visual Studio Community zur anschließenden Installation herunterladen.

Installation des AWS Toolkit for Visual Studio

Das AWS Toolkit for Visual Studioist ein Open-Source-Plug-in für die Visual Studio-IDE, das Entwicklern das Entwickeln, Debuggen und Bereitstellen von.NET-Anwendungen mithilfe von AWS. Installationsanleitungen finden Sie auf der Toolkit for Visual Studio-Homepage.

Verwenden des Elastic Beanstalk Beanstalk-.NET-Kerns auf einer Linux-Plattform

In diesem Thema wird beschrieben, wie Sie Ihre.NET Core auf Linux-Anwendungen auf Elastic Beanstalk konfigurieren, erstellen und ausführen.

AWS Elastic Beanstalk unterstützt eine Reihe von Plattformzweigen für verschiedene Versionen des .NET Core-Frameworks, die auf dem Linux-Betriebssystem ausgeführt werden. Eine vollständige Liste finden Sie unter .NET Core unter Linux unter AWS Elastic Beanstalk Plattformen.

Details zu den verschiedenen Erweiterungsmöglichkeiten einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter the section called “Erweitern von Linux-Plattformen”.

Überlegungen zur Plattform .NET Core auf Linux

Proxy-Server

Die Elastic Beanstalk-.NET Core unter Linux-Plattform enthält einen Reverse-Proxy-Server, der Anforderungen an die Anwendung weiterleitet. Standardmäßig verwendet Elastic Beanstalk NGINX als Proxyserver. Sie können festlegen, dass kein Proxy-Server verwendet wird, und Kestrel als Webserver Ihrer Anwendung konfigurieren. Kestrel ist standardmäßig in ASP.NET Core-Projektvorlagen enthalten.

Struktur der Anwendung

Sie können laufzeitabhängige Anwendungen veröffentlichen, die die von Elastic Beanstalk bereitgestellte .NET Core-Laufzeitumgebung verwenden. Sie können auch eigenständige Anwendungen veröffentlichen, die die .NET Core-Laufzeit und die Abhängigkeiten Ihrer Anwendung im Quellpaket enthalten. Weitere Informationen hierzu finden Sie unter the section called “Anwendungsbündelung”.

Plattformkonfiguration

Schließen Sie eine optionale Procfile-Datei in Ihr Quell-Bundle ein, um die Prozesse zu konfigurieren, die auf den Server-Instances in Ihrer Umgebung ausgeführt werden. Eine `Procfile` ist erforderlich, wenn das Quellpaket mehr als eine Anwendung enthält.

Sie sollten mit Ihrer Anwendung jedoch stets eine `Procfile`-Datei im Quellpaket bereitstellen. So können Sie genau steuern, welche Prozesse von Elastic Beanstalk für Ihre Anwendung ausgeführt werden.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das Ändern der Konfiguration einer ausgeführten Umgebung verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie gespeicherte Konfigurationen verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie Konfigurationsdateien einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe Konfigurationsoptionen.

Konfigurieren Ihrer .NET Core unter Linux-Umgebung

Mithilfe der .NET Core unter Linux-Plattformeinstellungen können Sie das Verhalten Ihrer Amazon EC2-Instances optimieren. Sie können die Konfiguration der Amazon EC2-Instance der Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-Konsole bearbeiten.

Mithilfe der Elastic Beanstalk-Konsole aktivieren Sie die Protokollrotation an Amazon S3 und konfigurieren Variablen, die die Anwendung aus der Umgebung auslesen kann.

So konfigurieren Sie Ihre .NET Core unter Linux-Umgebung mit der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon-EC2-Instances der Anwendung in den der Anwendung zugeordneten Amazon-S3-Bucket kopiert werden.

Umgebungseigenschaften

Im Abschnitt Environment properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon-EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel/Wert-Paare an die Anwendung weitergeleitet.

Innerhalb der .NET Core unter Linux-Umgebung, die in Elastic Beanstalk ausgeführt wird, kann über `Environment.GetEnvironmentVariable("variable-name")` auf Umgebungsvariablen zugegriffen werden. Beispielsweise können Sie mit folgendem Code eine Eigenschaft mit dem Namen `API_ENDPOINT` als Variable lesen.

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

.NET Core unter Linux-Konfigurations-Namespace

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben durchführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Die .NET Core unter Linux-Plattform unterstützt zusätzlich zu den für alle Elastic Beanstalk-Umgebungen unterstützten Optionen auch Optionen im folgenden Namespace:

- `aws:elasticbeanstalk:environment:proxy`— Wählen Sie, ob Sie NGINX oder keinen Proxyserver verwenden möchten. Gültige Werte sind `nginx` oder `none`.

In der folgenden Beispielkonfigurationsdatei wird die Verwendung der für .NET Core unter Linux spezifischen Konfigurationsoptionen gezeigt.

Example.ebextensions/proxy-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: none
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Bündelung von Anwendungen für den.NET Core auf der Linux-Plattform Elastic Beanstalk

Sie können sowohl laufzeitabhängige als auch eigenständige .NET Core-Anwendungen auf AWS Elastic Beanstalk ausführen.

Eine laufzeitabhängige Anwendung verwendet eine .NET Core-Laufzeit, die Elastic Beanstalk für die Ausführung Ihrer Anwendung bereitstellt. Elastic Beanstalk verwendet die Datei `runtimeconfig.json` in Ihrem Quellpaket, um die für Ihre Anwendung zu verwendende Laufzeit

zu bestimmen. Elastic Beanstalk wählt die neueste kompatible Laufzeit, die für Ihre Anwendung verfügbar ist.

Eine eigenständige Anwendung enthält die .NET Core-Laufzeit, Ihre Anwendung und ihre Abhängigkeiten. Um eine Version der .NET Core-Laufzeit zu verwenden, die nicht von Elastic Beanstalk mit den Plattformen bereitgestellt wird, stellen Sie eine eigenständige Anwendung bereit.

Beispiele

Sie können mit dem Befehl `dotnet publish` sowohl eigenständige als auch laufzeitabhängige Anwendungen kompilieren. Weitere Informationen zum Veröffentlichen von .NET Core-Apps finden Sie unter Übersicht über die Veröffentlichung von .NET Core-Anwendungen in der .NET Core-Dokumentation.

Die folgende Beispieldateistruktur definiert eine einzelne Anwendung, die eine von Elastic Beanstalk bereitgestellte .NET Core-Laufzeit verwendet.

```
### appsettings.Development.json
### appsettings.json
### dotnetcoreapp.deps.json
### dotnetcoreapp.dll
### dotnetcoreapp.pdb
### dotnetcoreapp.runtimeconfig.json
### web.config
### Procfile
### .ebextensions
### .platform
```

Sie können in das Quellpaket mehrere Anwendungen einfügen. Im folgenden Beispiel werden zwei Anwendungen definiert, die auf demselben Webserver ausgeführt werden sollen. Um mehrere Anwendungen auszuführen, müssen Sie eine Procfile-Datei in das Quellpaket einfügen. Eine vollständige Beispielanwendung finden Sie unter -apps.zip. dotnet-core-linux-multiple

```
### DotnetMultipleApp1
#   ### Amazon.Extensions.Configuration.SystemsManager.dll
#   ### appsettings.Development.json
#   ### appsettings.json
#   ### AWSSDK.Core.dll
#   ### AWSSDK.Extensions.NETCore.Setup.dll
#   ### AWSSDK.SimpleSystemsManagement.dll
#   ### DotnetMultipleApp1.deps.json
```

```
#   ### DotnetMultipleApp1.dll
#   ### DotnetMultipleApp1.pdb
#   ### DotnetMultipleApp1.runtimeconfig.json
#   ### Microsoft.Extensions.PlatformAbstractions.dll
#   ### Newtonsoft.Json.dll
#   ### web.config
### DotnetMultipleApp2
#   ### Amazon.Extensions.Configuration.SystemsManager.dll
#   ### appsettings.Development.json
#   ### appsettings.json
#   ### AWSSDK.Core.dll
#   ### AWSSDK.Extensions.NETCore.Setup.dll
#   ### AWSSDK.SimpleSystemsManagement.dll
#   ### DotnetMultipleApp2.deps.json
#   ### DotnetMultipleApp2.dll
#   ### DotnetMultipleApp2.pdb
#   ### DotnetMultipleApp2.runtimeconfig.json
#   ### Microsoft.Extensions.PlatformAbstractions.dll
#   ### Newtonsoft.Json.dll
#   ### web.config
### Procfile
### .ebextensions
### .platform
```

Verwenden einer Profildatei zur Konfiguration Ihrer Elastic Beanstalk Beanstalk-Umgebung von.NET Core auf Linux

Um mehrere Anwendungen auf demselben Webserver auszuführen, müssen Sie eine `Procfile`-Datei in das Quellpaket einfügen, die Elastic Beanstalk mitteilt, welche Anwendungen ausgeführt werden sollen.

Sie sollten mit Ihrer Anwendung jedoch stets eine `Procfile`-Datei im Quellpaket bereitstellen. Auf diese Weise können Sie genau steuern, welche Prozesse Elastic Beanstalk für Ihre Anwendung ausführt und welche Argumente diese Prozesse erhalten.

Im folgenden Beispiel wird eine `Procfile`-Datei verwendet, um Elastic Beanstalk zwei Anwendungen anzugeben, die auf demselben Webserver ausgeführt werden sollen.

Example Procfile

```
web: dotnet ./dotnet-core-app1/dotnetcoreapp1.dll
```

```
web2: dotnet ./dotnet-core-app2/dotnetcoreapp2.dll
```

Einzelheiten zum Schreiben und Verwenden von a finden Sie `Procfile` unterBuildfile und Procfile.

Konfigurieren des Proxy-Server

AWS Elastic Beanstalk verwendet NGINX als Reverse-Proxy, um Anfragen an Ihre Anwendung weiterzuleiten. Elastic Beanstalk bietet eine Standard-NGINX-Konfiguration, die Sie mit Ihrer eigenen Konfiguration entweder erweitern oder komplett überschreiben können.

Standardmäßig konfiguriert Elastic Beanstalk den NGINX-Proxy so, dass er Anfragen an Ihre Anwendung auf Port 5000 weiterleitet. Diese Port-Standardeinstellung können Sie überschreiben. Legen Sie dazu die `PORT`-Umgebungseigenschaft auf den Überwachungsport der Hauptanwendung fest.

> **Note**
>
> Der Port, auf dem Ihre Anwendung lauscht, hat keinen Einfluss auf den Port, auf dem der NGINX-Server Anfragen vom Load Balancer empfängt.

Konfigurieren des Proxy-Servers auf Ihrer Plattformversion

Alle AL2023 AL2 /-Plattformen unterstützen eine einheitliche Proxykonfigurationsfunktion. Weitere Informationen zur Konfiguration des Proxyservers auf Ihren Plattformversionen, auf denen AL2023/ ausgeführt wirdAL2, finden Sie unterReverse-Proxy-Konfiguration.

Die folgende Beispielkonfigurationsdatei erweitert die NGINX-Konfiguration Ihrer Umgebung. Die Konfiguration leitet Anforderungen für `/api` an eine zweite Webanwendung weiter, die Port 5200 auf dem Webserver überwacht. Standardmäßig leitet Elastic Beanstalk Anforderungen an eine einzelne Anwendung weiter, die den Port 5000 überwacht.

Example **`01_custom.conf`**

```
location /api {
     proxy_pass          http://127.0.0.1:5200;
     proxy_http_version  1.1;

     proxy_set_header   Upgrade $http_upgrade;
     proxy_set_header   Connection $http_connection;
     proxy_set_header   Host $host;
```

```
    proxy_cache_bypass $http_upgrade;
    proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header   X-Forwarded-Proto $scheme;
}
```

The AWS Toolkit for Visual Studio - Arbeiten mit.Net Core auf Elastic Beanstalk

In diesem Thema wird gezeigt, wie Sie die folgenden Aufgaben mit dem AWS Toolkit for Visual Studio ausführen können:

- Erstellen Sie eine ASP.NET Core-Webanwendung mit einer Visual Studio-Vorlage.
- Erstellen Sie eine Elastic Beanstalk Amazon Linux-Umgebung.
- Stellen Sie die ASP.NET Core-Webanwendung in der neuen Amazon Linux-Umgebung bereit.

In diesem Thema wird auch untersucht, wie Sie die verwenden können AWS Toolkit for Visual Studio , um Ihre Elastic Beanstalk Beanstalk-Anwendungsumgebungen zu verwalten und den Zustand Ihrer Anwendung zu überwachen.

Das AWS Toolkit for Visual Studio ist ein Plugin für die Visual Studio-IDE. Mit dem Toolkit können Sie Anwendungen in Elastic Beanstalk bereitstellen und verwalten, während Sie in Ihrer Visual Studio-Umgebung arbeiten.

Sections

- Voraussetzungen
- Erstellen eines neuen Anwendungsprojekts
- Erstellen einer Elastic Beanstalk-Umgebung und Bereitstellen Ihrer Anwendung
- Beenden einer Umgebung
- Verwalten Ihrer Elastic Beanstalk-Anwendungsumgebungen
- Überwachen des Anwendungsstatus

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie das AWS Toolkit for Visual Studio installieren. Anweisungen finden Sie unter Einrichten des AWS Toolkit for Visual Studio.

Wenn Sie das Toolkit noch nie zuvor verwendet haben, müssen Sie nach der Installation des Toolkits zunächst Ihre AWS Anmeldeinformationen im Toolkit registrieren. Weitere Informationen hierzu finden Sie unter Bereitstellen von AWS -Anmeldeinformationen.

Erstellen eines neuen Anwendungsprojekts

Wenn Sie über kein .NET Core-Anwendungsprojekt in Visual Studio verfügen, können Sie ganz einfach eines mit einer der Visual Studio-Projektvorlagen erstellen.

So erstellen Sie ein neues ASP.NET Core-Webanwendungsprojekt:

1. Wählen Sie im Menü File (Datei) von Visual Studio New (Neu) aus und dann Project (Projekt).
2. Wählen Sie im Dialogfeld Create a new project (Neues Projekt erstellen) die Option C#, klicken Sie dann auf Linux und anschließend auf Cloud.
3. Wählen Sie in der Liste der angezeigten Projektvorlagen ASP.NET Core Web Application (ASP.NET Core-Webanwendung) aus, und klicken Sie dann auf Weiter.

   > **Note**
   >
   > Wenn ASP.NET Core Web Application (ASP.NET Core-Webanwendung) nicht in den Projektvorlagen aufgeführt ist, können Sie sie in Visual Studio installieren.
   >
   > 1. Führen Sie einen Bildlauf zum Ende der Vorlagenliste durch, und wählen Sie den Link Weitere Tools und Funktionen installieren unter der Vorlagenliste aus.
   > 2. Wenn Sie aufgefordert werden, zuzulassen, dass die Visual Studio-Anwendung Änderungen an Ihrem Gerät vornimmt, wählen Sie Yes (Ja) aus.
   > 3. Wählen Sie die Registerkarte Workloads aus, und klicken Sie Sie dann auf ASP.NET and web development (ASP.NET und Webentwicklung).
   > 4. Klicken Sie auf die Schaltfläche Ändern. Über den Visual Studio Installer (Installationsprogramm von Visual-Studio) wird dann die Projektvorlage installiert.
   > 5. Nachdem das Installationsprogramm abgeschlossen ist, verlassen Sie das Fenster, um wieder an die Stelle zurückzukehren, an der Sie in Visual Studio aufgehört haben.

4. Geben Sie im Dialogfeld Configure your new project (Neues Projekt erstellen) einen Projektnamen ein. Als Solution name (Lösungsname) wird standardmäßig der Projektname verwendet. Klicken Sie als Nächstes auf Erstellen.

5. Wählen Sie im Dialogfeld Create a new ASP.NET Core web application (Neue ASP.NET Core-Webanwendung erstellen) die Option .NET Core aus, und klicken Sie dann auf ASP.NET Core 3.1. Wählen Sie in der Liste der angezeigten Anwendungstypen Webanwendung aus, und klicken Sie dann auf die Schaltfläche Erstellen.

In Visual Studio wird während der Anwendungserstellung das Dialogfeld Creating Project (Projekt wird erstellt) angezeigt. Nachdem Visual Studio Ihre Anwendung erstellt hat, wird ein Fenster mit dem Namen Ihrer Anwendung angezeigt.

Erstellen einer Elastic Beanstalk-Umgebung und Bereitstellen Ihrer Anwendung

In diesem Abschnitt wird beschrieben, wie Sie eine Elastic Beanstalk-Umgebung für Ihre Anwendung erstellen und Ihre Anwendung in dieser Umgebung bereitstellen.

Erstellen einer neuen Umgebung und Bereitstellen Ihrer Anwendung

1. Wählen Sie in Visual Studio die Option View (Anzeigen) und dann Solution Explorer (Lösungs-Explorer) aus.
2. Öffnen Sie im Solution Explorer (Lösungs-Explorer) das Kontextmenü (mit Rechtsklick) für die Anwendung, und klicken Sie dann auf In AWS Elastic Beanstalk veröffentlichen.
3. Geben Sie im Assistenten In AWS Elastic Beanstalk veröffentlichen Ihre Kontoinformationen ein.

    a. Wählen Sie für Account profile to use (Zu verwendendes Kontoprofil) Ihr Standard-Konto oder klicken Sie auf das Add another account (Ein anderes Konto hinzufügen)-Symbol, um neue Kontoinformationen einzugeben.

    b. Wählen Sie unter Region die Region aus, in der die Anwendung bereitgestellt werden soll. Informationen zu verfügbaren AWS Regionen finden Sie unter AWS Elastic Beanstalk Endpunkte und Kontingente in der. Allgemeine AWS-Referenz Falls Sie eine nicht von Elastic Beanstalk unterstützte Region auswählen, ist die Option für eine Elastic Beanstalk-Bereitstellung nicht verfügbar.

    c. Wählen Sie Create a new application environment (Eine neue Anwendungsumgebung erstellen) aus, und klicken Sie dann auf Weiter.
4. Geben Sie im Dialogfeld Application Environment (Anwendungsumgebung) die Details für die neue Anwendungsumgebung ein.

5. Legen Sie im nächsten AWSOptionsdialogfeld EC2 Amazon-Optionen und andere AWS verwandte Optionen für Ihre bereitgestellte Anwendung fest.

   a. Wählen Sie als Containertyp 64-Bit-Amazon Linux 2 v, auf ***<n.n.n>*** dem .NET Core ausgeführt wird.

   > **Note**
   >
   > Wir empfehlen Ihnen, die aktuelle Plattformversion von Linux auszuwählen. Diese Version enthält die neuesten Sicherheits- und Fehlerbehebungen, die in unserem neuesten Amazon Machine Image (AMI) enthalten sind.

   b. Wählen Sie für Instance-Typ die Option t2.micro aus. (Durch die Wahl eines Micro-Instance-Typs werden die mit dem Betrieb der Instance verbundenen Kosten minimiert).

   c. Wählen Sie im Feld Key pair (Schlüsselpaar) die Option Create new key pair (Neues Schlüsselpaar erstellen) aus. Geben Sie einen Namen für das neue Schlüsselpaar ein, und wählen Sie dann OK aus. (In diesem Beispiel verwenden wir **`myuseastkeypair`**.) Ein key pair ermöglicht den Remote-Desktop-Zugriff auf Ihre EC2 Amazon-Instances. Weitere Informationen zu EC2 Amazon-Schlüsselpaaren finden Sie unter Using Credentials im Amazon Elastic Compute Cloud-Benutzerhandbuch.

   d. Für eine einfach Anwendung mit geringem Datenverkehr wählen Sie Single instance environment (Single-Instance-Umgebung) aus. Weitere Informationen finden Sie unter Typen von Umgebungen.

   e. Klicken Sie auf Weiter.

   Weitere Informationen zu den AWS Optionen, die in diesem Beispiel nicht verwendet werden, finden Sie auf den folgenden Seiten:

   - Informationen zu Use custom AMI (Verwenden eines benutzerdefinierten AMI) finden Sie unter Verwenden eines benutzerdefinierten Amazon Machine Images (AMI) in Ihrer Elastic Beanstalk Beanstalk-Umgebung.
   - Wenn Sie keine Single instance environment (Single-Instance-Umgebung) verwenden, müssen Sie einen Load balance type (Load-Balance-Typ) auswählen. Weitere Informationen finden Sie unter Load Balancer Ihrer Elastic-Beanstalk-Umgebung.
   - Elastic Beanstalk verwendet die Standardkonfiguration Amazon VPC (Amazon Virtual Private Cloud), sofern Sie nicht die Option Use non-default VPC (Nicht-Standard-VPC verwenden)

ausgewählt haben. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon VPC.

- Wenn Sie die Option Enable Rolling Deployments (Rollende Bereitstellungen aktivieren) auswählen, wird eine Bereitstellung in Stapel aufgeteilt, um mögliche Ausfallzeiten während der Bereitstellungen zu vermeiden. Weitere Informationen finden Sie unter Anwendungen in Elastic Beanstalk-Umgebungen bereitstellen.
- Durch die Auswahl der Option Relational Database Access (Zugriff auf relationale Datenbank) wird die Elastic Beanstalk-Umgebung in einer zuvor erstellten Amazon RDS-Datenbank mit Amazon RDS-DB-Sicherheitsgruppen verknüpft. Weitere Informationen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen im Amazon RDS-Benutzerhandbuch.

6. Klicken Sie im Dialogfeld Berechtigungen auf Weiter.
7. Klicken Sie im Dialogfeld Anwendungsoptionen auf Weiter.
8. Überprüfen Sie die Bereitstellungsoptionen. Nachdem Sie überprüft haben, dass Ihre Einstellungen korrekt sind, wählen Sie Deploy (Bereitstellen) aus.

Ihre ASP.NET Core-Webanwendung wird als Webbereitstellungsdatei exportiert. Diese Datei wird anschließend in Amazon S3 hochgeladen und als neue Anwendungsversion in Elastic Beanstalk registriert. Mit der Bereitstellungsfunktion von Elastic Beanstalk wird die Umgebung überwacht, bis sie mit dem neu bereitgestellten Code verfügbar ist. Auf der Registerkarte „Env:<Umgebungsname>“ wird der Status für Ihre Umgebung angezeigt. Nachdem der Status auf Environment is healthy (Umgebung ist fehlerfrei) aktualisiert wurde, können Sie die URL-Adresse auswählen, um die Webanwendung zu starten.

Beenden einer Umgebung

Um zu vermeiden, dass Gebühren für ungenutzte AWS Ressourcen anfallen, können Sie den verwenden, AWS Toolkit for Visual Studio um eine laufende Umgebung zu beenden.

> **Note**
>
> Sie können zu einem späteren Zeitpunkt problemlos eine neue Umgebung mit derselben Version starten.

So beenden Sie eine Umgebung

1. Erweitern Sie den Elastic Beanstalk-Knoten und den Anwendungsknoten. Öffnen Sie im AWS -Explorer das Kontextmenü (mit Rechtsklick) für Ihre Anwendungsumgebung, und wählen Sie Umgebung beenden.
2. Wenn Sie dazu aufgefordert werden, wählen Sie Ja, um zu bestätigen, dass Sie die Umgebung beenden möchten. Es dauert einige Minuten, bis Elastic Beanstalk die in der Umgebung laufenden AWS Ressourcen beendet.

Der Status für Ihre Umgebung auf der Registerkarte „Env:<Umgebungsname>“ ändert sich in Wird beendet und schließlich in Beendet.

> **Note**
>
> Nachdem die Umgebung beendet ist, steht ihr zugehöriger CNAME wieder zur Verfügung und kann von allen verwendet werden.

Verwalten Ihrer Elastic Beanstalk-Anwendungsumgebungen

In diesem Abschnitt werden die spezifischen Service-Einstellungen beschrieben, die Sie im AWS Toolkit for Visual Studio als Teil Ihrer Anwendungs-Umgebungskonfiguration bearbeiten können. Mit dem AWS Toolkit for Visual Studio und der AWS Management Console können Sie die Bereitstellung und Konfiguration der von Ihren AWS Anwendungsumgebungen verwendeten Ressourcen ändern. Informationen zur Verwaltung Ihrer Anwendungsumgebungen mithilfe der AWS Management Console finden Sie unter. Elastic Beanstalk Beanstalk-Umgebungen verwalten

Ändern der Umgebungskonfigurationseinstellungen

Wenn Sie Ihre Anwendung bereitstellen, konfiguriert Elastic Beanstalk mehrere verbundene AWS Cloud-Computing-Dienste. Sie können steuern, wie diese einzelnen Dienste konfiguriert werden, indem Sie das AWS Toolkit for Visual Studio verwenden.

So bearbeiten Sie die Umgebungseinstellungen einer Anwendung

1. Wählen Sie in Visual Studio im Menü Datei die Option AWS -Explorer.

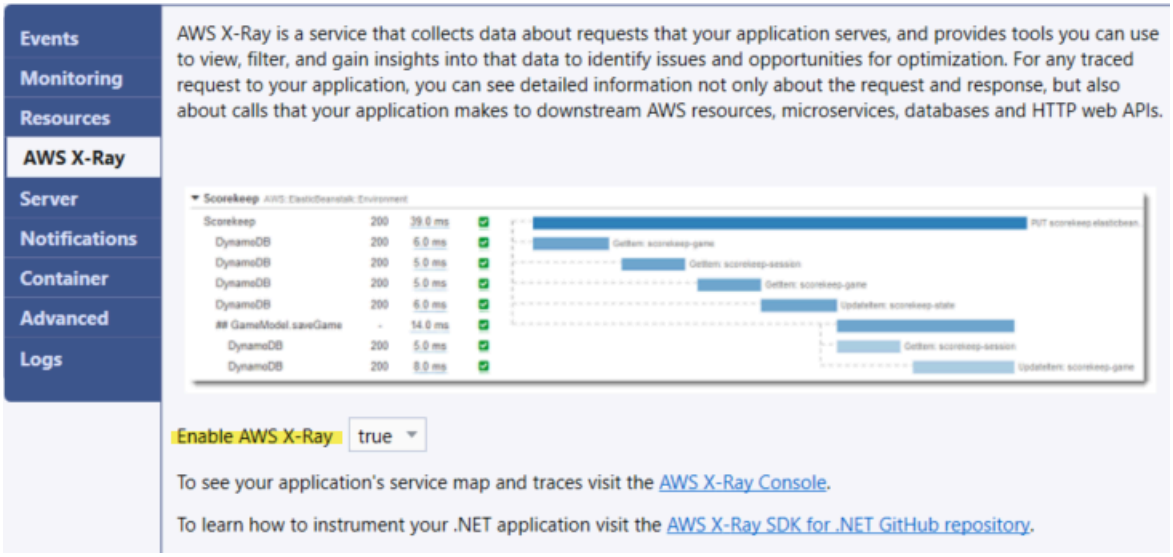
2. Erweitern Sie den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten. Öffnen Sie das Kontextmenü (mit Rechtsklick) für Ihre Anwendungsumgebung, und wählen Sie View Status (Status anzeigen).

   Sie können nun Einstellungen für Folgendes konfigurieren:

   - AWS X-Ray
   - Server
   - Load Balancer (gilt nur für Umgebungen mit mehreren Instances)
   - Auto Scaling (gilt nur für Umgebungen mit mehreren Instances)
   - Benachrichtigungen
   - Behälter
   - Erweiterte Konfigurationsoptionen

Konfiguration von AWS X-Ray mit dem AWS Toolkit für Visual Studio

AWS X-Ray bietet Funktionen zur Nachverfolgung von Anfragen, zur Erfassung von Ausnahmen und zur Profilerstellung. Mit dem AWS X-Ray-Bedienfeld können Sie X-Ray für Ihre Anwendung aktivieren oder deaktivieren. Weitere Informationen zu X-Ray finden Sie im Entwicklerhandbuch für AWS X-Ray.

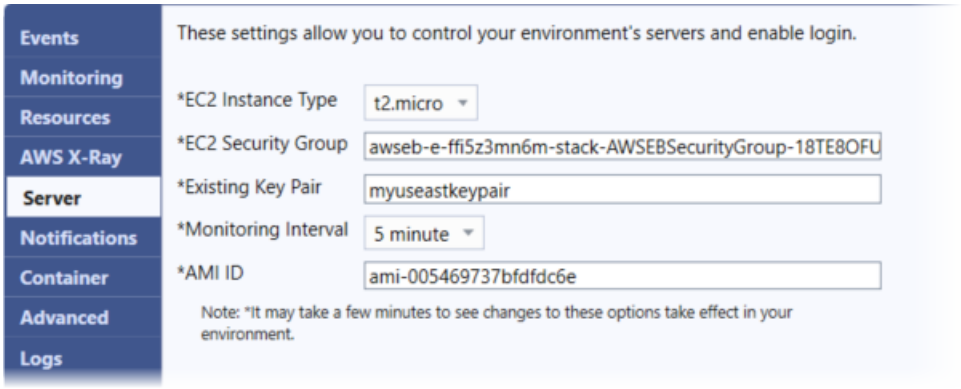


Konfiguration von EC2 Instanzen mit dem AWS Toolkit für Visual Studio

Sie können Amazon Elastic Compute Cloud (Amazon EC2) verwenden, um Server-Instances in den Rechenzentren von Amazon zu starten und zu verwalten. Sie können EC2 Amazon-Server-

Instances jederzeit, so lange Sie benötigen und für jeden legalen Zweck verwenden. Instances sind in verschiedenen Größen und Konfigurationen verfügbar. Weitere Informationen finden Sie auf Amazon EC2.

Sie können Ihre EC2 Amazon-Instance-Konfiguration mit der Registerkarte Server innerhalb Ihrer Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.



EC2 Amazon-Instance-Typen

Instance type (Instance-Typ) zeigt die für Ihre Elastic Beanstalk-Anwendung verfügbaren Instance-Typen an. Ändern Sie den Instance-Typ, um einen Server mit den Merkmalen (einschließlich Speichergröße und CPU-Leistung) auszuwählen, die für Ihre Anwendung am besten geeignet sind. Anwendungen mit aufwändigen und lange dauernden Operationen erfordern mehr CPU oder Speicher.

Weitere Informationen zu den EC2 Amazon-Instance-Typen, die für Ihre Elastic Beanstalk Beanstalk-Anwendung verfügbar sind, finden Sie unter Instance-Typen im Amazon Elastic Compute Cloud-Benutzerhandbuch.

EC2 Amazon-Sicherheitsgruppen

Sie können den Zugriff auf Ihre Elastic Beanstalk Beanstalk-Anwendung mithilfe einer Amazon EC2 Security Group kontrollieren. Eine Sicherheitsgruppe definiert Firewall-Regeln für Ihre Instances. Diese Regeln legen fest, welcher eingehende Netzwerkverkehr an Ihre Instance übertragen werden soll. Der gesamte andere eingehende Datenverkehr wird verworfen. Sie können die Regeln für eine Gruppe jederzeit ändern. Die neuen Regeln werden automatisch für alle laufenden und künftigen Instances durchgesetzt.

Sie können angeben, welche EC2 Amazon-Sicherheitsgruppen den Zugriff auf Ihre Elastic Beanstalk Beanstalk-Anwendung kontrollieren. Geben Sie dazu die Namen bestimmter EC2 Amazon-Sicherheitsgruppen (trennen Sie mehrere Sicherheitsgruppen durch Kommas) in das Textfeld EC2

Sicherheitsgruppen ein. Sie können dies entweder mit der AWS Management Console oder dem AWS Toolkit for Visual Studio tun.

Um eine Sicherheitsgruppe mit dem AWS Toolkit für Visual Studio zu erstellen

1. Erweitern Sie in Visual Studio im AWS Explorer den EC2Amazon-Knoten und wählen Sie dann Sicherheitsgruppen aus.
2. Wählen Sie Sicherheitsgruppe erstellen aus und geben Sie einen Namen und eine Beschreibung für Ihre Sicherheitsgruppe ein.
3. Wählen Sie OK.

Weitere Informationen zu Amazon EC2 Security Groups finden Sie unter Using Security Groups im Amazon Elastic Compute Cloud-Benutzerhandbuch.

EC2 Amazon-Schlüsselpaare

Sie können sich mit einem Amazon-Schlüsselpaar sicher bei den EC2 Amazon-Instances anmelden, die für Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt wurden. EC2

> **Important**
>
> Sie müssen ein EC2 Amazon-Schlüsselpaar erstellen und Ihre von Elastic Beanstalk bereitgestellten EC2 Amazon-Instances konfigurieren, um auf diese Instances zugreifen zu können. Sie können Ihr key pair mit dem Publish to AWS Wizard im AWS Toolkit for Visual Studio erstellen, wenn Sie Ihre Anwendung auf Elastic Beanstalk bereitstellen. Wenn Sie zusätzliche Schlüsselpaare mit dem Toolkit erstellen möchten, führen Sie die hier beschriebenen Schritte aus. Alternativ können Sie Ihre EC2 Amazon-Schlüsselpaare über die AWS Management Console einrichten. Anweisungen zum Erstellen eines key pair für Amazon finden Sie im Amazon EC2 Elastic Compute Cloud Getting Started Guide.

Im Textfeld Bestehendes key pair können Sie den Namen eines EC2 Amazon-Schlüsselpaars angeben, mit dem Sie sich sicher bei den EC2 Amazon-Instances anmelden können, auf denen Ihre Elastic Beanstalk Beanstalk-Anwendung ausgeführt wird.

Um den Namen eines EC2 Amazon-Schlüsselpaars anzugeben

1. Erweitern Sie den EC2Amazon-Knoten und wählen Sie Schlüsselpaare aus.

2. Wählen Sie Schlüsselpaar erstellen aus und geben Sie den Namen des Schlüsselpaares ein.
3. Wählen Sie OK.

Weitere Informationen zu EC2 Amazon-Schlüsselpaaren finden Sie unter Using Amazon EC2 Credentials im Amazon Elastic Compute Cloud-Benutzerhandbuch. Weitere Informationen zur Verbindung mit EC2 Amazon-Instances finden Sie unter

Überwachungsintervall

Standardmäßig sind nur grundlegende CloudWatch Amazon-Metriken aktiviert. Sie geben Daten in 5-Minuten-Intervallen zurück. Sie können detailliertere CloudWatch Ein-Minuten-Metriken aktivieren, indem Sie im Bereich Server auf der Registerkarte Konfiguration für Ihre Umgebung unter 1 Minute für das Überwachungsintervall auswählen. AWS Toolkit for Eclipse

> **Note**
>
> CloudWatch Amazon-Servicegebühren können für Metriken im Abstand von einer Minute anfallen. Weitere Informationen erhalten Sie CloudWatchbei Amazon.

Benutzerdefinierte AMI-ID

Sie können das Standard-AMI, das für Ihre EC2 Amazon-Instances verwendet wird, durch Ihr eigenes benutzerdefiniertes AMI überschreiben, indem Sie die ID Ihres benutzerdefinierten AMI in das Feld Benutzerdefinierte AMI-ID im Abschnitt Server der Registerkarte Konfiguration für Ihre Umgebung in der eingeben AWS Toolkit for Eclipse.

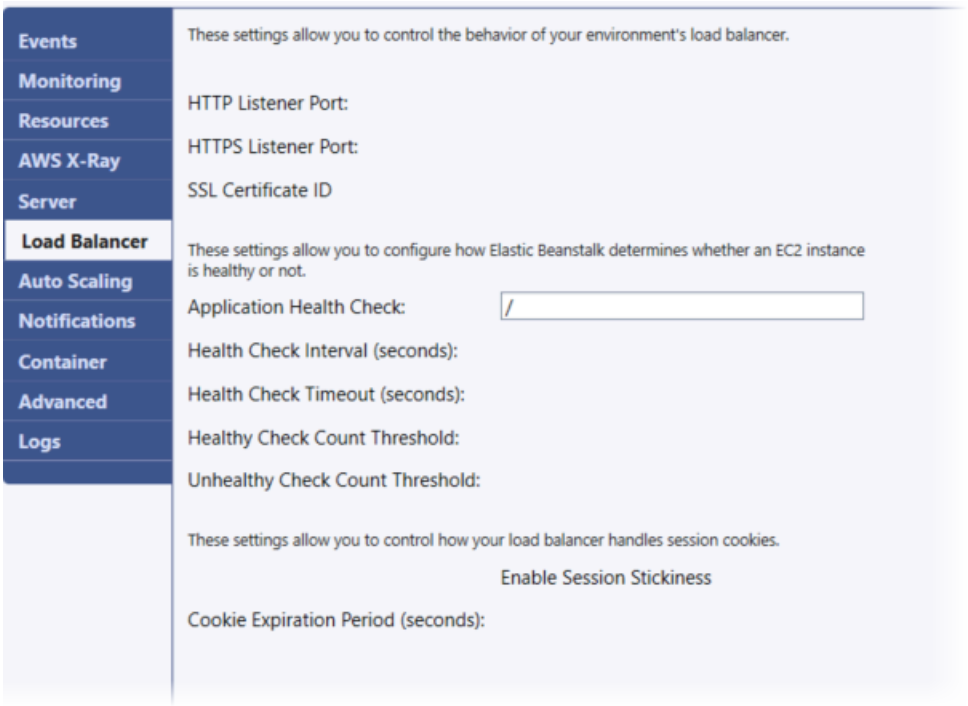
> **Important**
>
> Die Verwendung eines eigenen AMI ist eine Aufgabe für Fortgeschrittene und sollte mit Bedacht ausgeführt werden. Wenn Sie ein benutzerdefiniertes AMI benötigen, empfehlen wir, das standardmäßige Elastic Beanstalk-AMI als Ausgangspunkt zu nehmen und es zu ändern. Um als fehlerfrei angesehen zu werden, geht Elastic Beanstalk davon aus, dass EC2 Amazon-Instances eine Reihe von Anforderungen erfüllen, einschließlich eines laufenden Host-Managers. Werden diese Anforderungen nicht erfüllt, funktioniert Ihre Umgebung möglichweise nicht ordnungsgemäß.

Konfiguration von Elastic Load Balancing mit dem AWS Toolkit für Visual Studio

Elastic Load Balancing ist ein Amazon-Webservice, der Ihnen hilft, die Verfügbarkeit und Skalierbarkeit Ihrer Anwendung zu verbessern. Dieser Service macht es Ihnen leicht, Anwendungslasten auf zwei oder mehr EC2 Amazon-Instances zu verteilen. Elastic Load Balancing verbessert die Verfügbarkeit durch die Bereitstellung zusätzlicher Redundanz und unterstützt wachsenden Datenverkehr für Ihre Anwendung.

Mit Elastic Load Balancing können Sie den eingehenden Anwendungsdatenverkehr automatisch auf alle ausgeführten Instances verteilen und ausgleichen. Sie können auch ganz einfach neue Instances hinzufügen, wenn eine Erhöhung der Kapazität Ihrer Anwendung erforderlich ist.

Elastic Beanstalk stellt Elastic Load Balancing automatisch bereit, wenn Sie eine Anwendung bereitstellen. Sie können die EC2 Amazon-Instance-Konfiguration der Elastic Beanstalk Beanstalk-Umgebung mit der Registerkarte Load Balancer innerhalb Ihrer Anwendungsumgebung in AWS Toolkit for Visual Studio bearbeiten.



In den folgenden Abschnitten werden die Elastic Load Balancing-Parameter beschrieben, die Sie für Ihre Anwendung konfigurieren können.

Ports

Der Load Balancer, der für die Bearbeitung von Anfragen für Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt wurde, sendet Anfragen an die EC2 Amazon-Instances, auf denen Ihre

Anwendung ausgeführt wird. Der bereitgestellte Load Balancer kann Anfragen an HTTP- und HTTPS-Ports abhören und Anfragen an die EC2 Amazon-Instances in Ihrer AWS Elastic Beanstalk Anwendung weiterleiten. Standardmäßig verarbeitet der Load Balancer Anfragen auf dem HTTP-Port. Dazu muss mindestens einer der Ports (HTTP oder HTTPS) aktiviert werden.



> **Important**
>
> Stellen Sie sicher, dass der von Ihnen angegebene Port nicht gesperrt ist. Andernfalls können Sie keine Verbindung zu Ihrer Elastic Beanstalk-Anwendung herstellen.

Steuern des HTTP-Ports

Um den HTTP-Port auszuschalten, wählen Sie OFF (AUS) für HTTP Listener Port aus. Zum Aktivieren des HTTP-Ports wählen Sie einen HTTP-Port (z. B. 80) aus der Liste aus.

> **Note**
>
> Für den Zugriff auf Ihre Umgebung mit einem anderen als dem Standardport 80, z. B. Port 8080, fügen Sie dem vorhandenen Load Balancer einen Listener hinzu und konfigurieren Sie den neuen Listener für die Überwachung des entsprechenden Ports.
> Geben Sie beispielsweise AWS CLI für Classic Load Balancer den folgenden Befehl ein und *LOAD_BALANCER_NAME* ersetzen Sie ihn durch den Namen Ihres Load Balancers für Elastic Beanstalk.
>
> ```
> aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
>  --listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
>  InstancePort=80"
> ```
>
> Geben Sie beispielsweise AWS CLI für Application Load Balancers den folgenden Befehl ein und *LOAD_BALANCER_ARN* ersetzen Sie ihn durch den ARN Ihres Load Balancers für Elastic Beanstalk.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
 --port 8080
```

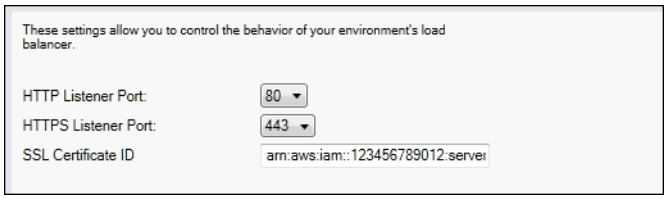
Wenn Elastic Beanstalk Ihre Umgebung überwachen soll, entfernen Sie den Listener für Port 80 nicht.

Steuern des HTTPS-Ports

Elastic Load Balancing unterstützt das HTTPS/TLS Protokoll, um die Verschlüsselung des Datenverkehrs für Client-Verbindungen zum Load Balancer zu ermöglichen. Verbindungen vom Load Balancer zu den EC2 Instances verwenden Klartextverschlüsselung. Standardmäßig ist der HTTPS-Port deaktiviert.

So aktivieren Sie den HTTPS-Port

1. Erstellen Sie ein neues Zertifikat mit AWS Certificate Manager (ACM) oder laden Sie ein Zertifikat und einen Schlüssel zu AWS Identity and Access Management (IAM) hoch. Weitere Informationen zum Anfordern eines ACM-Zertifikats finden Sie unter Zertifikat anfordern im AWS Certificate Manager -Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in ACM finden Sie unter Importieren von Zertifikaten im AWS Certificate Manager -Benutzerhandbuch. Wenn ACM in Ihrer Region nicht verfügbar ist, verwenden Sie AWS Identity and Access Management (IAM), um ein Zertifikat eines Drittanbieters hochzuladen. Die ACM- und IAM-Services speichern das Zertifikat und stellen einen Amazon-Ressourcennamen (ARN) für das SSL-Zertifikat bereit. Weitere Informationen zum Erstellen und Hochladen von Zertifikaten in IAM finden Sie unter Arbeiten mit Serverzertifikaten im IAM-Benutzerhandbuch.
2. Geben Sie den HTTPS-Port an, indem Sie einen Port für HTTPS Listener Port auswählen.



3. Geben Sie für SSL Certificate ID (SSL-Zertifikat-ID) den Amazon-Ressourcennamen (ARN) Ihres SSL-Zertifikats ein. Beispiel: **`arn:aws:iam::123456789012:server-certificate/abc/certs/build`** oder **`arn:aws:acm:us-`**

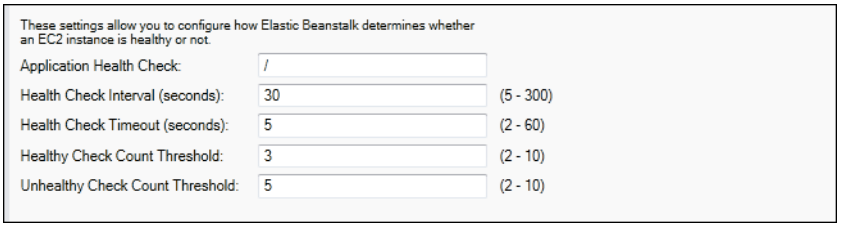
east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678.
Verwenden Sie das SSL-Zertifikat, das Sie in Schritt 1 erstellt oder hochgeladen haben.

Um den HTTPS-Port auszuschalten, wählen Sie OFF (AUS) für HTTPS Listener Port.

Zustandsprüfungen

Die Definition der Zustandsprüfung enthält einen URL, die für den Instance-Zustand abgefragt wird. Standardmäßig nutzt Elastic Beanstalk TCP:80 für Nonlegacy-Container und HTTP:80 für Legacy-Container. Sie können die Standard-URL überschreiben, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/default.aspx`). Dazu geben Sie diese im Feld Application health check URL (Zustandsprüfungs-URL für die Anwendung) ein. Wenn Sie die Standard-URL überschreiben, verwendet Elastic Beanstalk HTTP zum Abfragen der Ressource. Wenn Sie überprüfen möchten, ob Sie einen Legacy-Container verwenden, erhalten Sie Informationen unter the section called “Warum sind einige Plattformversionen als Legacy gekennzeichnet?”

Sie können die Einstellungen für die Integritätsprüfung im Bereich EC2 Instance Health Check im Bereich Load Balancing steuern.



Die Definition der Zustandsprüfung enthält einen URL, die für den Instance-Zustand abgefragt wird. Überschreiben Sie die Standard-URL, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/index.jsp`). Dazu geben Sie diese im Feld Application Health Check URL (Zustandsprüfungs-URL für die Anwendung) ein.

In der folgenden Liste werden die Zustandsprüfungs-Parameter beschrieben, die Sie für Ihre Anwendung festlegen können.

- Geben Sie für Health Check Interval (Sekunden) die Anzahl der Sekunden ein, die Elastic Load Balancing zwischen den Zustandsprüfungen für die EC2 Amazon-Instances Ihrer Anwendung wartet.
- Für Health Check Timeout (seconds) (Zustandsprüfungs-Timeout (Sekunden)) geben Sie die Anzahl der Sekunden ein, die Elastic Load Balancing auf eine Antwort wartet, bevor die Instance als nicht reagierend eingestuft wird.

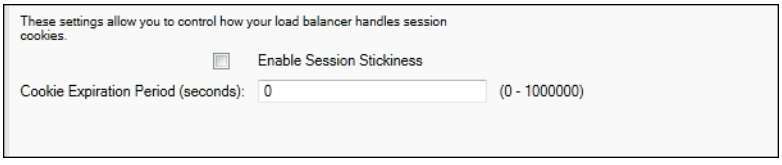
- Für Healthy Check Count Threshold (Schwellenwert Anzahl erfolgreicher Zustandsprüfungen) und Unhealthy Check Count Threshold (Schwellenwert Anzahl fehlgeschlagener Zustandsprüfungen) geben Sie die Anzahl der aufeinanderfolgenden erfolgreichen oder nicht erfolgreichen URL-Proben ein, bevor Elastic Load Balancing den Status des Instance-Zustands ändert. Beispiel: Die Angabe von **5** für Unhealthy Check Count Threshold (Schwellenwert Anzahl fehlgeschlagener Zustandsprüfungen) bedeutet, dass die URL fünfmal hintereinander eine Fehlermeldung oder eine Zeitüberschreitung zurückgeben muss, bevor Elastic Load Balancing die Zustandsprüfung als fehlgeschlagen einstufen würde.

Sitzungen

Standardmäßig leitet ein Load Balancer jede Anforderung getrennt an die Server-Instance mit der kleinsten Last weiter. Im Vergleich dazu bindet eine Sticky Session eine Benutzersitzung an eine bestimmte Server-Instance, sodass alle vom Benutzer während der Sitzung gesendeten Anforderungen an die gleiche Server-Instance übermittelt werden.

Elastic Beanstalk nutzt von Load Balancer generierte HTTP-Cookies, wenn Sticky Sessions für eine Anwendung aktiviert sind. Der Load Balancer verwendet ein vom Load-Balancer generiertes spezielles Cookie, um die Anwendungs-Instance für jede Anforderung nachzuverfolgen. Wenn der Load Balancer eine Anforderung empfängt, prüft er zunächst, ob dieses Cookie in der Anforderung vorhanden ist. Wenn er vorhanden ist, wird die Anforderung an die Anwendungs-Instance gesendet, die im Cookie angegeben ist. Wenn kein Cookie vorhanden ist, wählt der Load Balancer eine Anwendungs-Instance basierend auf dem vorhandenen Load Balancing-Algorithmus aus. Ein Cookie wird in die Antwort eingefügt, um nachfolgende Anforderungen von demselben Benutzer an diese Anwendungs-Instance zu binden. Mit der Konfiguration der Richtlinie wird ein Cookie-Ablauf definiert, der die Dauer der Gültigkeit für jedes Cookie festlegt.

Sie können über den Abschnitt Sessions (Sitzungen) auf der Registerkarte Load Balancer angeben, ob der Load Balancer für Ihre Anwendung "Sticky Sessions" erlaubt.



Weitere Informationen zu Elastic Load Balancing finden Sie im Elastic Load Balancing-Entwicklerhandbuch.

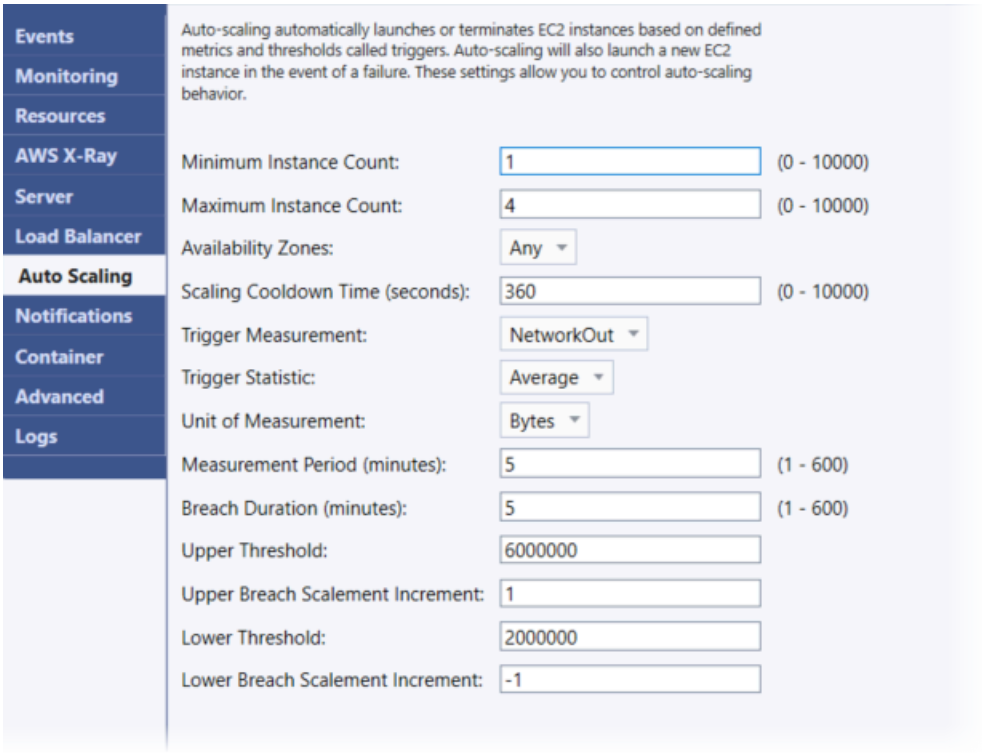
Konfiguration von Auto Scaling mit dem AWS Toolkit für Visual Studio

Amazon EC2 Auto Scaling ist ein Amazon-Webservice, der darauf ausgelegt ist, EC2 Amazon-Instances auf der Grundlage benutzerdefinierter Trigger automatisch zu starten oder zu beenden. Sie können Auto Scaling-Gruppen einrichten und diesen Gruppen Auslöser zuweisen, um Datenverarbeitungsressourcen automatisch basierend auf Metriken, wie Bandbreitennutzung oder CPU-Auslastung, zu skalieren. Amazon EC2 Auto Scaling arbeitet mit Amazon zusammen CloudWatch , um Metriken für die Server-Instances abzurufen, auf denen Ihre Anwendung ausgeführt wird.

Mit Amazon EC2 Auto Scaling können Sie eine Gruppe von EC2 Amazon-Instances nehmen und verschiedene Parameter festlegen, sodass die Anzahl dieser Gruppen automatisch erhöht oder verringert wird. Amazon EC2 Auto Scaling kann EC2 Amazon-Instances zu dieser Gruppe hinzufügen oder daraus entfernen, damit Sie problemlos mit Verkehrsänderungen in Ihrer Anwendung umgehen können.

Amazon EC2 Auto Scaling überwacht auch den Zustand jeder EC2 Amazon-Instance, die es startet. Wenn eine Instance unerwartet beendet wird, erkennt Amazon EC2 Auto Scaling die Kündigung und startet eine Ersatz-Instance. Mit dieser Funktion können Sie automatisch eine feste, gewünschte Anzahl von EC2 Amazon-Instances verwalten.

Elastic Beanstalk stellt Amazon EC2 Auto Scaling für Ihre Anwendung bereit. Sie können die EC2 Amazon-Instance-Konfiguration der Elastic Beanstalk Beanstalk-Umgebung mit der Registerkarte Auto Scaling innerhalb Ihrer Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.

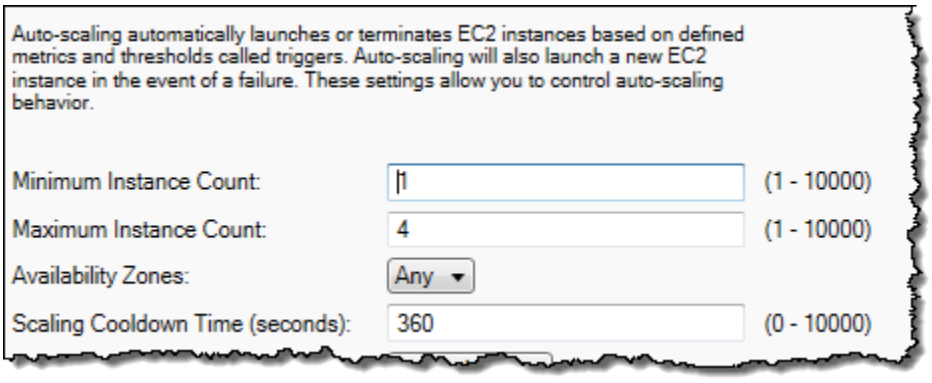


Im folgenden Abschnitt wird beschrieben, wie Sie Auto Scaling-Parameter für die Anwendung konfigurieren.

Starten der Konfiguration

Sie können die Startkonfiguration bearbeiten, um zu steuern, wie Ihre Elastic Beanstalk Beanstalk-Anwendung Amazon EC2 Auto Scaling Scaling-Ressourcen bereitstellt.

In den Feldern Minimum Instance Count (Minimale Instance-Anzahl) and Maximum Instance Count (Maximale Instance-Anzahl) geben Sie die minimale sowie die maximale Größe der Auto Scaling-Gruppe an, die von der Elastic Beanstalk-Anwendung genutzt wird.



> **Note**
>
> Um eine feste Anzahl von EC2 Amazon-Instances beizubehalten, setzen Sie Minimum Instance Count und Maximum Instance Count auf denselben Wert.

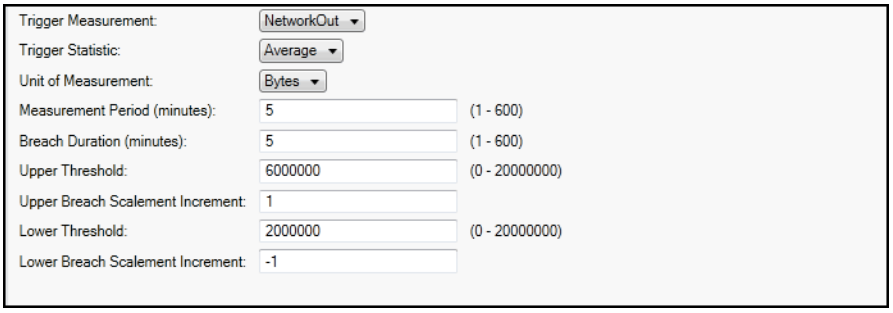
Im Feld Availability Zones können Sie die Anzahl der Availability Zones angeben, in denen sich Ihre EC2 Amazon-Instances befinden sollen. Es ist wichtig, diese Zahl festzulegen, wenn Sie fehlertolerante Anwendungen erstellen möchten. Falls eine Availability Zone ausfällt, werden die Instances in den anderen Availability Zones weiterhin ausgeführt.

> **Note**
>
> Derzeit ist es nicht möglich anzugeben, in welcher Availability Zone sich Ihre Instance befinden wird.

Auslöser

Ein Trigger ist ein Amazon EC2 Auto Scaling Scaling-Mechanismus, den Sie so einrichten, dass er dem System mitteilt, wann Sie die Anzahl der Instances erhöhen (Scale Out) oder verringern (Scale In) möchten. Sie können Trigger so konfigurieren, dass sie bei jeder auf Amazon veröffentlichten Metrik ausgelöst werden CloudWatch (z. B. CPU-Auslastung) und feststellen, ob die von Ihnen angegebenen Bedingungen erfüllt sind. Wird der obere oder untere Schwellenwert der Metrikbedingungen für einen bestimmten Zeitraum überschritten, startet der Auslöser den anhaltenden Prozess der Skalierung.

Mit dem AWS Toolkit for Visual Studio können Sie einen Skalierungs-Trigger für Ihre Elastic Beanstalk Beanstalk-Anwendung definieren.



Amazon EC2 Auto Scaling-Trigger funktionieren, indem sie eine bestimmte CloudWatch Amazon-Metrik einer bestimmten Instance überwachen. Zu den Metriken zählen CPU-Auslastung,

Netzwerkdatenverkehr und Datenträgeraktivität. Mit der Einstellung Trigger Measurement (Auslösermessung) wählen Sie eine Metrik für den Auslöser aus.

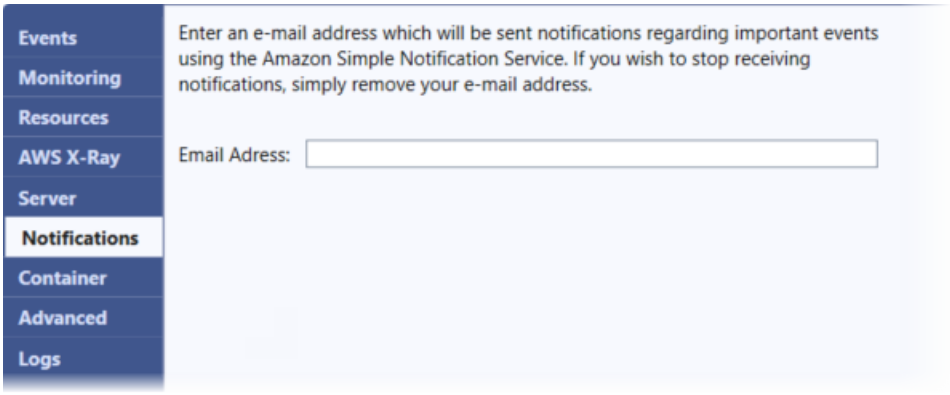
In der folgenden Liste werden die Trigger-Parameter beschrieben, die Sie mit der AWS Management Console konfigurieren können.

- Sie können angeben, welche Statistik der Auslöser verwenden soll. Sie können Minimum, Maximum, Sum (Summe) oder Average (Durchschnitt) für Trigger Statistic (Auslöserstatistik) auswählen.
- Für Unit of Measurement (Maßeinheit) geben Sie die Maßeinheit für den Auslöser an.
- Der Wert im Feld Messzeitraum gibt an, wie oft Amazon die Metriken für Ihren Trigger CloudWatch misst. Mit Breach Duration (Verletzungsdauer) wird der Zeitraum angegeben, für den der Grenzwert einer Metrik überschritten werden kann (wie unter Upper Threshold (Oberer Schwellenwert) und Lower Threshold (Unterer Schwellenwert) definiert), bevor der Auslöser ausgelöst wird.
- Geben Sie für Upper Breach Scale Increment und Lower Breach Scale Increment an, wie viele EC2 Amazon-Instances hinzugefügt oder entfernt werden sollen, wenn eine Skalierungsaktivität ausgeführt wird.

Weitere Informationen zu Amazon EC2 Auto Scaling finden Sie im Abschnitt Amazon EC2 Auto Scaling in der Amazon Elastic Compute Cloud-Dokumentation.

Konfiguration von Benachrichtigungen mit dem AWS Toolkit für Visual Studio

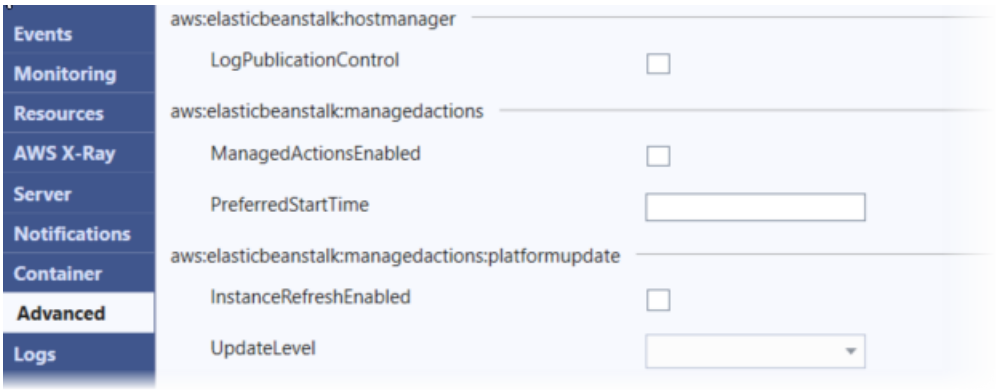
Elastic Beanstalk verwendet den Amazon Simple Notification Service (Amazon SNS) für Benachrichtigungen zu wichtigen Ereignissen in Ihrer Anwendung. Um Amazon SNS-Benachrichtigungen zu aktivieren, geben Sie Ihre E-Mail-Adresse in das Feld Email Address ein. Um diese Benachrichtigungen zu deaktivieren, entfernen Sie Ihre E-Mail-Adresse aus dem Feld.



Konfiguration zusätzlicher Umgebungsoptionen mit dem AWS Toolkit für Visual Studio

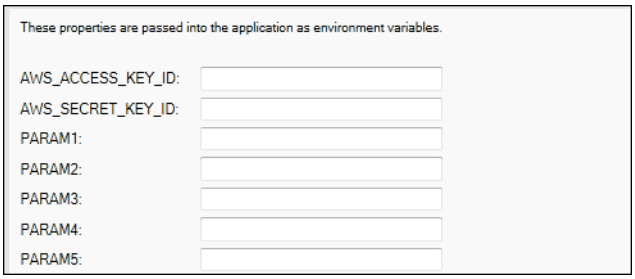
Elastic Beanstalk definiert eine große Anzahl von Konfigurationsoptionen, mithilfe derer Sie das Verhalten Ihrer Umgebung und die darin enthaltenen Ressourcen konfigurieren können. Konfigurationsoptionen sind in Namespaces wie `aws:autoscaling:asg` organisiert. In jedem Namespace werden Optionen für die Auto Scaling-Gruppe einer Umgebung definiert. Im Fenster Erweitert werden die Namespaces der Konfigurationsoptionen in alphabetischer Reihenfolge aufgelistet, die Sie nach der Erstellung der Umgebung aktualisieren können.

Eine vollständige Liste der Namespaces und Optionen einschließlich der Standard- und unterstützten Werte finden Sie unter Allgemeine Optionen für alle Umgebungen und .NET Core auf Linux-Plattformoptionen.



Konfiguration von.NET Core-Containern mit dem AWS Toolkit für Visual Studio

Im Fenster Container können Sie Umgebungsvariablen angeben, die Sie aus Ihrem Anwendungscode lesen können.



Überwachen des Anwendungsstatus

In diesem Thema wird erklärt, wie Sie den Zustand der Website Ihrer Anwendung überwachen können. Es ist wichtig zu wissen, dass Ihre Produktionswebsite verfügbar ist und auf Anfragen reagiert. Elastic Beanstalk bietet Funktionen, mit denen Sie die Reaktionsfähigkeit Ihrer Anwendung

überwachen können. Es werden Statistiken über Ihre Anwendung überwacht und Sie werden gewarnt, wenn Schwellenwerte überschritten werden.

Weitere Informationen über die Integritätsüberwachung von Elastic Beanstalk finden Sie unter Grundlegende Zustandsberichte.

Sie können auf Betriebsinformationen zu Ihrer Anwendung zugreifen, indem Sie entweder das AWS Toolkit for Visual Studio oder die AWS Management Console verwenden.

Das Toolkit zeigt den Zustand Ihrer Umgebung und den Anwendungsstatus im Feld Status an.

So überwachen Sie den Anwendungsstatus

1. Erweitern Sie im AWS Toolkit for Visual Studio im AWS Explorer den Elastic Beanstalk-Knoten und dann Ihren Anwendungsknoten.
2. Öffnen Sie das Kontextmenü (mit Rechtsklick) für Ihre Anwendungsumgebung, und wählen Sie View Status (Status anzeigen).
3. Wählen Sie auf der Registerkarte der Anwendungsumgebung Monitoring (Überwachung) aus.

   Der Bereich Monitoring (Überwachung) enthält eine Reihe von Diagrammen, die die Ressourcennutzung für Ihre bestimmte Anwendungsumgebung zeigen.

   > **Note**
   >
   > Standardmäßig ist der Zeitraum auf der letzten Stunde gesetzt. Ändern Sie diese Einstellung, indem Sie in der Liste Zeitraum einen anderen Zeitraum auswählen.

Sie können das AWS Toolkit for Visual Studio oder die AWS Management Console verwenden, um Ereignisse im Zusammenhang mit Ihrer Anwendung anzuzeigen.

So zeigen Sie Anwendungsereignisse an

1. Erweitern Sie im AWS Toolkit for Visual Studio im AWS Explorer den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten.
2. Öffnen Sie das Kontextmenü (mit Rechtsklick) für Ihre Anwendungsumgebung, und wählen Sie View Status (Status anzeigen).
3. Wählen Sie in der Registerkarte der Anwendungsumgebung Ereignisse aus.

Migration von .NET auf Windows Server-Plattform zur .NET Core auf Linux-Plattform auf Elastic Beanstalk

Sie können Anwendungen, die auf Plattformen auf der Basis von .NET unter Windows Server ausgeführt werden, zu .NET Core unter Linux-Plattformen migrieren. Im Folgenden werden einige Aspekte beschrieben, die bei der Migration von Windows zu Linux-Plattformen berücksichtigt werden sollten.

Aspekte, die bei der Migration zur .NET Core unter Linux-Plattform berücksichtigt werden sollten

Area	Änderungen und Informationen
Konfiguration der Anwendung	Auf Windows-Plattformen verwenden Sie ein Bereitstellungsmanifest zur Angabe der in Ihrer Umgebung ausgeführten Anwendungen. Die .NET Core unter Linux-Plattformen verwenden eine Procfile-Datei zur Angabe der auf den Instances Ihrer Umgebung ausgeführten Anwendungen. Details zur Bündelung von Anwendungen finden Sie unter the section called “Anwendungsbündelung”.
Proxy-Server	Auf Windows-Plattformen verwenden Sie IIS als Proxy-Server Ihrer Anwendung. Die .NET Core unter Linux-Plattformen enthalten standardmäßig nginx als Reverse-Proxy. Sie können festlegen, dass kein Proxy-Server verwendet wird, und Kestrel als Webserver Ihrer Anwendung verwenden. Weitere Informationen hierzu finden Sie unter the section called “Proxy-Server”.
Routing	Auf Windows-Plattformen verwenden Sie IIS in Ihrem Anwendungscode und fügen ein Bereitstellungsmanifest ein, um den IIS-Pfad zu konfigurieren. Für die .NET Core unter Linux-Plattform verwenden Sie ASP .NET Core-Routing im Anwendungscode und aktualisieren die nginx-Konfiguration Ihrer Umgebung. Weitere Informationen hierzu finden Sie unter the section called “Proxy-Server”.
Protokolle	Linux- und Windows-Plattformen streamen verschiedene Protokolle. Details hierzu finden Sie unter the section called “So richtet Elastic Beanstalk Logs ein CloudWatch”.

Bereitstellung von Go-Anwendungen mit Elastic Beanstalk

Dieses Kapitel enthält Anweisungen zur Konfiguration und Bereitstellung Ihrer Go-Webanwendung für. AWS Elastic Beanstalk Elastic Beanstalk macht es einfach, Ihre Go-Webanwendungen mithilfe von Amazon Web Services bereitzustellen, zu verwalten und zu skalieren.

Sie können Ihre Anwendung in nur wenigen Minuten mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) oder der Elastic Beanstalk Beanstalk-Konsole bereitstellen. Nachdem Sie Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt haben, können Sie weiterhin die EB-CLI verwenden, um Ihre Anwendung und Umgebung zu verwalten, oder Sie können die Elastic Beanstalk Beanstalk-Konsole oder die. AWS CLI APIs

Folgen Sie den step-by-step Anweisungen QuickStart für Go zum Erstellen und Bereitstellen einer Hello World Go-Webanwendung mit der EB CLI.

Themen

- QuickStart: Eine Go-Anwendung auf Elastic Beanstalk bereitstellen
- Einrichtung Ihrer Go-Entwicklungsumgebung für Elastic Beanstalk
- Verwenden der Go-Plattform von Elastic Beanstalk

QuickStart: Eine Go-Anwendung auf Elastic Beanstalk bereitstellen

Dieses QuickStart Tutorial führt Sie durch den Prozess der Erstellung einer Go-Anwendung und deren Bereitstellung in einer AWS Elastic Beanstalk Umgebung.

> ⚠ Nicht für den Produktionsgebrauch
>
> Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine Go-Anwendung

- Schritt 2: Stellen Sie Ihre Go-Anwendung mit der EB CLI bereit
- Schritt 3: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus
- Schritt 4: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center -Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

EB-CLI

Dieses Tutorial verwendet die Befehlszeilenschnittstelle Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Schritt 1: Erstellen Sie eine Go-Anwendung

Erstellen Sie ein Projektverzeichnis.

```
~$ mkdir eb-go
~$ cd eb-go
```

Erstellen Sie anschließend eine Anwendung, die Sie mit Elastic Beanstalk bereitstellen. Wir werden einen „Hello World“ RESTful -Webservice erstellen.

In diesem Beispiel wird eine benutzerdefinierte Begrüßung gedruckt, die je nach Pfad, der für den Zugriff auf den Service verwendet wurde, variiert.

Erstellen Sie in diesem Verzeichnis eine Textdatei mit dem Namen `application.go` mit folgendem Inhalt.

Example~/eb-go/application.go

```
package main

import (
 "fmt"
 "net/http"
)

func handler(w http.ResponseWriter, r *http.Request) {
 if r.URL.Path == "/" {
  fmt.Fprintf(w, "Hello World! Append a name to the URL to say hello. For example, use
 %s/Mary to say hello to Mary.", r.Host)
 } else {
  fmt.Fprintf(w, "Hello, %s!", r.URL.Path[1:])
 }
}

func main() {
 http.HandleFunc("/", handler)
 http.ListenAndServe(":5000", nil)
}
```

Schritt 2: Stellen Sie Ihre Go-Anwendung mit der EB CLI bereit

Jetzt erstellen Sie die Anwendungsumgebung und stellen die konfigurierte Anwendung in Elastic Beanstalk bereit.

Erstellen einer Umgebung und Bereitstellen der Go-Anwendung

1. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

```
~/eb-go$ eb init -p go go-tutorial --region us-east-2
Application go-tutorial has been created.
```

Dieser Befehl erstellt eine Anwendung mit dem Namen `go-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der neuesten Go-Plattformversion erstellt werden.

2. (Optional) Führen Sie den Vorgang eb init erneut aus, um ein Standardschlüsselpaar zu konfigurieren, sodass Sie SSH verwenden können, um eine Verbindung zu der EC2 Instance herzustellen, auf der Ihre Anwendung ausgeführt wird.

```
~/eb-go$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein Schlüsselpaar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erstellt automatisch eine Zip-Datei für Ihre Anwendung und startet sie auf Port 5000.

```
~/eb-go$ eb create go-env
```

Es dauert etwa fünf Minuten, bis Elastic Beanstalk Ihre Umgebung erstellt hat.

Schritt 3: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, öffnen Sie Ihre Website mit. eb open

```
~/eb-go$ eb open
```

Herzlichen Glückwunsch! Sie haben eine Go-Anwendung mit Elastic Beanstalk bereitgestellt! Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet.

Schritt 4: Bereinigen

Sie können Ihre Umgebung beenden, wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind. Elastic Beanstalk beendet alle AWS Ressourcen, die mit Ihrer Umgebung verknüpft sind.

Führen Sie den folgenden Befehl aus, um Ihre Elastic Beanstalk Beanstalk-Umgebung mit der EB-CLI zu beenden.

```
~/eb-go$ eb terminate
```

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt. Es dient als einfache Beispielanwendung mit einer einzigen EC2 Instanz, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 Instanz — Eine EC2 virtuelle Maschine von Amazon, die so konfiguriert ist, dass sie Web-Apps auf der von Ihnen ausgewählten Plattform ausführt.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.

- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Die Bereitstellung einer neuen Anwendungsversion ist sehr schnell, da dafür keine Bereitstellung oder Neustarts von Instances erforderlich sind. EC2 Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

Wenn Sie ein oder zwei Beispielanwendungen bereitgestellt haben, können Sie mit der lokalen Entwicklung und Ausführung von Go-Anwendungen beginnen. Informationen dazu finden Sie unter Einrichtung Ihrer Go-Entwicklungsumgebung für Elastic Beanstalk.

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

Einrichtung Ihrer Go-Entwicklungsumgebung für Elastic Beanstalk

Dieses Thema enthält Anweisungen zum Einrichten einer Go-Entwicklungsumgebung, in der Sie Ihre Anwendung lokal testen können, bevor Sie sie bereitstellen. AWS Elastic Beanstalk Es verweist auch auf Websites, die Installationsanweisungen für nützliche Tools bereitstellen.

Installieren von Go

Zum lokalen Ausführen von Go-Anwendungen installieren Sie Go. Wenn Sie keine bestimmte Version benötigen, laden Sie die neueste Version herunter, die Elastic Beanstalk unterstützt. Eine Liste der unterstützten Versionen finden Sie unter Go im Dokument AWS Elastic Beanstalk -Plattformen.

Laden Sie Go unter https://golang herunter. org/doc/install.

Das AWS SDK for Go installieren

Wenn Sie AWS Ressourcen von Ihrer Anwendung aus verwalten müssen, installieren Sie das AWS SDK for Go mit dem folgenden Befehl.

```
$ go get github.com/aws/aws-sdk-go
```

Weitere Informationen finden Sie unter AWS SDK for Go.

Verwenden der Go-Plattform von Elastic Beanstalk

In diesem Thema wird beschrieben, wie Sie Ihre Go-Anwendungen auf Elastic Beanstalk konfigurieren, erstellen und ausführen.

AWS Elastic Beanstalk unterstützt eine Reihe von Plattformzweigen für verschiedene Versionen der Go-Programmiersprache. Eine vollständige Liste finden Sie unter Go im Dokument AWS Elastic Beanstalk Plattformen.

Bei einfachen Go-Anwendungen gibt es zwei Möglichkeiten, Ihre Anwendung bereitzustellen:

- Stellen Sie ein Quell-Bundle mit einer Quelldatei im Stammverzeichnis namens `application.go` bereit, das das Hauptpaket Ihrer Anwendung enthält. Elastic Beanstalk entwickelt den binären Wert mithilfe des folgenden Befehls:

  ```
  go build -o bin/application application.go
  ```

  Nachdem die Anwendung erstellt wurde, startet Elastic Beanstalk sie auf Port 5000.
- Stellen Sie ein Quell-Bundle mit einer Binärdatei namens `application` bereit. Die Binärdatei befindet sich entweder im Stammverzeichnis des Quell-Bundle oder im `bin/`-Verzeichnis des Quell-Bundle. Wenn Sie die `application`-Binärdatei in beiden Speicherorten platzieren, verwendet Elastic Beanstalk die Datei im `bin/`-Verzeichnis.

  Elastic Beanstalk startet diese Anwendung auf Port 5000.

In beiden Fällen können Sie mit unseren unterstützten Go-Plattformbranchen auch die Modulanforderungen in einer Datei mit dem Namen angeben`go.mod`. Weitere Informationen finden Sie unter Migrating to Go Modules (Migration zu Go-Modulen) im Go-Blog.

Bei komplexeren Go-Anwendungen gibt es zwei Möglichkeiten, um Ihre Anwendung bereitzustellen:

- Stellen Sie ein Quell-Bundle bereit, das Ihre Anwendungsquelldateien zusammen mit einer Buildfile-Datei und einer Procfile-Datei enthält. Die Buildfile-Datei enthält einen Befehl zum Erstellen der Anwendung und die Procfile-Datei enthält Anweisungen zum Ausführen der Anwendung.
- Stellen Sie ein Quell-Bundle bereit, das Ihre Binärdateien der Anwendung mit einer Procfile-Datei enthält. Die Procfile-Datei enthält Anweisungen zur Ausführung der Anwendung.

Die Go-Plattform umfasst einen Proxy-Server, der statische Komponenten bereitstellt und den Datenverkehr an Ihre Anwendung weiterleitet. Bei erweiterten Szenarien können Sie die Standard-Proxy-Konfiguration erweitern oder überschreiben.

Details zu den verschiedenen Erweiterungsmöglichkeiten einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter the section called “Erweitern von Linux-Plattformen”.

Konfigurieren Ihrer Go-Umgebung

Mit den Go-Plattformeinstellungen können Sie das Verhalten Ihrer EC2 Amazon-Instances optimieren. Sie können die EC2 Amazon-Instance-Konfiguration der Elastic Beanstalk Beanstalk-Umgebung mithilfe der Elastic Beanstalk Beanstalk-Konsole bearbeiten.

Mithilfe der Elastic Beanstalk-Konsole aktivieren Sie die Protokollrotation an Amazon S3 und konfigurieren Variablen, die die Anwendung aus der Umgebung auslesen kann.

So konfigurieren Sie Ihre Go-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Rotation der Protokolldateien nach Amazon S3 aktivieren — Gibt an, ob Protokolldateien für die EC2 Amazon-Instances Ihrer Anwendung in den Amazon S3-Bucket kopiert werden, der Ihrer Anwendung zugeordnet ist.

Statische Dateien

Um die Leistung zu verbessern, können Sie den Abschnitt Static files (Statische Dateien) benutzen, um den Proxy-Server so zu konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zum Verzeichnis-Mapping fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, anstatt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter the section called “Statische Dateien”.

Umgebungseigenschaften

Im Abschnitt Umgebungseigenschaften können Sie die Umgebungskonfigurationseinstellungen für die EC2 Amazon-Instances angeben, auf denen Ihre Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel/Wert-Paare an die Anwendung weitergeleitet.

Innerhalb der Go-Umgebung, die in Elastic Beanstalk ausgeführt wird, können Sie über die Funktion `os.Getenv` auf Umgebungsvariablen zugreifen. Beispielsweise können Sie mit folgendem Code eine Eigenschaft mit dem Namen `API_ENDPOINT` als Variable lesen:

```
endpoint := os.Getenv("API_ENDPOINT")
```

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Namespace der Go-Konfiguration

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben durchführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Die Go-Plattform definiert keine plattformspezifischen Namespaces. Sie können den Proxy so konfigurieren, dass statische Dateien mittels des `aws:elasticbeanstalk:environment:proxy:staticfiles`-Namespace bereitgestellt werden. Einzelheiten und ein Beispiel finden Sie unter the section called “Statische Dateien”.

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Die Amazon Linux AMI (Vorgängerversion von Amazon Linux 2) Go-Plattform

Wenn Ihre Elastic Beanstalk Go-Umgebung eine Amazon Linux AMI-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/ AL2 Plattformzweige sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
> - Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Namespaces für die Go-Konfiguration — Amazon Linux AMI () AL1

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben durchführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

> **Note**
>
> Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/ AL2 Plattformzweige sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.

Die Amazon Linux AMI Go-Plattform unterstützt zusätzlich zu den Namespaces, die von allen Plattformen unterstützt werden, eine weitere Konfigurationsoption für einen plattformspezifischen Namespace. Mit dem `aws:elasticbeanstalk:container:golang:staticfiles`-Namespace können Sie Optionen definieren, um für die Webanwendung Pfade zu Ordnern im Quell-Bundle der Anwendung zuzuweisen, die statische Inhalte enthalten.

Diese Konfigurationsdatei weist den Proxy-Server beispielsweise an, Dateien im `staticimages`-Ordner im Pfad `/images` bereitzustellen:

Example.ebextensions/go-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:golang:staticfiles:
    /html: statichtml
    /images: staticimages
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Konfiguration benutzerdefinierter Startbefehle mit einer Profildatei auf Elastic Beanstalk

Zur Angabe benutzerdefinierter Befehle zum Starten einer Go-Anwendung, schließen Sie eine Datei mit dem Namen `Procfile` in das Stammverzeichnis Ihres Quell-Bundle ein.

Einzelheiten zum Schreiben und Verwenden von a finden Sie `Procfile` unter. Buildfile und Procfile

Example Procfile

```
web: bin/server
```

```
queue_process: bin/queue_processor
foo: bin/fooapp
```

Sie müssen die Hauptanwendung `web` aufrufen und als ersten Befehl in Ihrem `Procfile` auflisten. Elastic Beanstalk stellt die `web`-Hauptanwendung auf der Stamm-URL der Umgebung bereit, z. B. `http://my-go-env.elasticbeanstalk.com`.

Elastic Beanstalk führt außerdem alle Anwendungen aus, deren Namen kein `web_`-Präfix haben. Diese Anwendungen sind jedoch nicht von außerhalb Ihrer Instance verfügbar.

Elastic Beanstalk erwartet von Prozessen, die aus der `Procfile` ausgeführt werden, dass sie kontinuierlich laufen. Elastic Beanstalk überwacht diese Anwendungen und startet alle Prozesse, die beendet werden, neu. Für kurz laufende Prozesse verwenden Sie einen Buildfile-Befehl.

Verwenden einer Procfile-Datei auf Amazon Linux-AMI (vor Amazon Linux 2)

Wenn Ihre Elastic Beanstalk Go-Umgebung eine Amazon Linux AMI-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/ AL2 Plattformzweige sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
> - Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Port-Weitergabe — Amazon Linux AMI (AL1)

> **Note**
>
> Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/ AL2 Plattformzweige sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.

Elastic Beanstalk konfiguriert den nginx-Proxy für die Weitergabe von Anfragen an Ihre Anwendung, und zwar auf der Portnummer, die in der `PORT` Umgebungseigenschaft für Ihre Anwendung angegeben ist. Ihre Anwendung sollte Port stets abhören. Sie können auf diese Variable im Anwendungscode zugreifen, indem Sie die `os.Getenv("PORT")`-Methode aufrufen.

Elastic Beanstalk nutzt die Portnummer, die in der `PORT`-Umgebungseigenschaft für den Port der ersten Anwendung in `Procfile` angegeben wurde und erhöht dann die Portnummer für jede nachfolgende Anwendung in der `Procfile`-Datei um 100. Wenn die `PORT`-Umgebungseigenschaft nicht festgelegt ist, verwendet für den anfänglichen Port Elastic Beanstalk 5000.

Im Beispiel oben ist die `PORT`-Umgebungseigenschaft der `web`-Anwendung 5000, der `queue_process`-Anwendung 5100 und der `foo`-Anwendung 5200.

Sie können den anfängliche Port angeben, indem Sie die `PORT`-Option mit dem aws:elasticbeanstalk:application:environment-Namespace festlegen, wie im folgenden Beispiel veranschaulicht.

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application:environment
    option_name:  PORT
    value:  <first_port_number>
```

Weitere Informationen zum Festlegen von Umgebungseigenschaften für Ihre Anwendung finden Sie unter Optionseinstellungen.

Benutzerdefinierter Build und Konfiguration mit einer Builddatei auf Elastic Beanstalk

Zur Angabe eines benutzerdefinierten Build- und Konfigurationsbefehls für Ihre Go-Anwendung, schließen Sie eine Datei mit dem Namen `Buildfile` in das Stammverzeichnis Ihres Quell-

Bundle ein. Beachten Sie beim Dateinamen die Groß- und Kleinschreibung. Verwenden Sie für die `Buildfile` folgendes Format:

```
<process_name>: <command>
```

Der Befehl in Ihrer `Buildfile` muss folgendem regulären Ausdruck entsprechen: `^[A-Za-z0-9_]+:\s*.+$`.

Elastic Beanstalk überwacht nicht die Anwendung, die mit einer `Buildfile` ausgeführt wird. Verwenden Sie eine `Buildfile` für Befehle, die für kurze Zeiträume ausgeführt und nach Abschluss ihrer Aufgaben beendet werden. Für lange laufende Anwendungsprozesse, die nicht beendet werden sollen, verwenden Sie stattdessen die Procfile-Datei.

Im folgenden Beispiel einer `Buildfile` ist `build.sh` ein Shell-Skript, das sich im Stammverzeichnis des Quell-Bundle befindet:

```
make: ./build.sh
```

Alle Pfade in der `Buildfile` sind relativ zum Stammverzeichnis des Quell-Bundle. Wenn Sie im Voraus wissen, wo sich die Dateien auf der Instance befinden, können Sie absolute Pfade in die `Buildfile` einschließen.

Konfigurieren des Proxy-Server

Elastic Beanstalk verwendet nginx als Reverse-Proxy, um Ihre Anwendung Ihrem Elastic-Load-Balancing-Load-Balancer auf Port 80 zuzuordnen. Elastic Beanstalk bietet eine nginx-Standardkonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Standardmäßig wird der nginx-Proxy von Elastic Beanstalk so konfiguriert, dass alle Anforderungen an die Anwendung über Port 5000 weitergeleitet werden. Diese Port-Standardeinstellung können Sie überschreiben. Legen Sie dazu die `PORT`-Umgebungseigenschaft auf den Überwachungsport der Hauptanwendung fest.

> **Note**
>
> Der Überwachungsport der Anwendung kollidiert nicht mit dem Überwachungsport des nginx-Servers, von dem dieser die Anforderungen des Load Balancers empfängt.

Konfigurieren des Proxy-Servers auf Ihrer Plattformversion

Alle AL2 023/ AL2 -Plattformen unterstützen eine einheitliche Proxy-Konfigurationsfunktion. Weitere Informationen zur Konfiguration des Proxyservers auf Ihren Plattformversionen, auf denen AL2 AL2 023/ ausgeführt wird, finden Sie unter. Reverse-Proxy-Konfiguration

Konfigurieren des Proxys in Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)

Hinweise

- Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/ AL2 Plattformzweige sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
- Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Wenn Ihre Elastic Beanstalk Go-Umgebung eine Amazon Linux AMI-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie die Informationen in diesem Abschnitt.

Erweiterung und Außerkraftsetzung der Standard-Proxykonfiguration — Amazon Linux AMI () AL1

Elastic Beanstalk nutzt einen nginx-Server als Reverseproxy-Server, um die Anwendung dem Load Balancer auf Port 80 zuzuweisen. Wenn Sie Ihre eigene nginx-Konfiguration bereitstellen möchten, können Sie die Standardkonfiguration von Elastic Beanstalk überschreiben, indem Sie die `.ebextensions/nginx/nginx.conf`-Datei in Ihr Quell-Bundle einschließen. Wenn sie vorhanden ist, nutzt Elastic Beanstalk diese Datei anstelle der nginx-Standardkonfigurationsdatei.

Wenn Sie Richtlinien zusätzlich zu denen im `nginx.conf` `http`-Block hinzufügen möchten, können Sie auch weitere Konfigurationsdateien im `.ebextensions/nginx/conf.d/`-Verzeichnis Ihres Quell-Bundle bereitstellen. Alle Dateien in diesem Verzeichnis müssen über die `.conf`-Erweiterung verfügen.

Wenn Sie die Funktionalität von Elastic Beanstalk nutzen möchten, z. B. Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk, automatische

Anwendungszuordnungen und statische Dateien, müssen Sie die folgende Zeile in den `server`-Block der nginx-Konfigurationsdatei einfügen:

```
include conf.d/elasticbeanstalk/*.conf;
```

Bereitstellung von Java-Anwendungen mit Elastic Beanstalk

Dieses Kapitel enthält Anweisungen zur Konfiguration und Bereitstellung Ihrer Java-Anwendungen auf. AWS Elastic Beanstalk Elastic Beanstalk macht es einfach, Ihre Java-Webanwendungen mithilfe von Amazon Web Services bereitzustellen, zu verwalten und zu skalieren.

Sie können Ihre Anwendung in nur wenigen Minuten mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) oder der Elastic Beanstalk Beanstalk-Konsole bereitstellen. Nachdem Sie Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt haben, können Sie weiterhin die EB-CLI verwenden, um Ihre Anwendung und Umgebung zu verwalten, oder Sie können die Elastic Beanstalk Beanstalk-Konsole verwenden, AWS CLI oder die. APIs

Folgen Sie den step-by-step Anweisungen QuickStart für Java zum Erstellen und Bereitstellen einer Hello World Java-Webanwendung mit der EB CLI. Wenn Sie an step-by-step Anleitungen zur Erstellung einer einfachen Hello World Java-JSP-Anwendung interessiert sind, die Sie mit der EB-CLI auf unserer Tomcat-basierten Plattform bereitstellen können, probieren Sie die. QuickStart für Java auf Tomcat

Die Java-Plattform verzweigt

AWS Elastic Beanstalk unterstützt zwei Plattformen für Java-Anwendungen.

- Tomcat — Eine Plattform, die auf Apache Tomcat basiert, einem Open-Source-Webcontainer für Anwendungen, die Java-Servlets und JavaServer Pages (JSPs) zur Bearbeitung von HTTP-Anfragen verwenden. Tomcat vereinfacht die Entwicklung von Webanwendungen durch Multithreading, eine deklarative Sicherheitskonfiguration und umfassende Anpassungsmöglichkeiten. Elastic Beanstalk hat Plattform-Branchen für jede der aktuellen Hauptversionen von Tomcat. Weitere Informationen finden Sie unter Die Tomcat-Plattform.
- Java SE — Eine Plattform für Anwendungen, die keinen Webcontainer oder einen anderen als Tomcat verwenden, wie Jetty oder. GlassFish Sie können jede Java-Archivbibliothek (JARs), die von Ihrer Anwendung verwendet wird, in das Quellpaket aufnehmen, das Sie auf Elastic Beanstalk bereitstellen. Weitere Informationen finden Sie unter Die Java SE-Plattform.

Neuere Zweige der Tomcat- und Java SE-Plattformen basieren auf Amazon Linux 2 und höher und verwenden Corretto, die AWS Java SE-Distribution. Die Namen dieser Plattformzweige enthalten das Wort Corretto anstelle von Java.

Eine Liste der aktuellen Plattformversionen finden Sie unter Tomcat und Java SE im AWS Elastic Beanstalk Plattform-Handbuch.

AWS Werkzeuge

AWS bietet mehrere Tools für die Arbeit mit Java und Elastic Beanstalk. Unabhängig davon, für welchen Plattformzweig Sie sich entscheiden, können Sie das AWS SDK for Java verwenden, um andere AWS Dienste aus Ihrer Java-Anwendung heraus zu verwenden. Das AWS SDK for Java besteht aus einer Reihe von Bibliotheken, mit denen Sie Code AWS APIs aus Ihrer Anwendung verwenden können, ohne die rohen HTTP-Aufrufe von Grund auf neu schreiben zu müssen.

Wenn Sie Ihre Anwendungen lieber über die Befehlszeile verwalten möchten, installieren Sie das Elastic Beanstalk Command Line Interface (EB CLI) und verwenden Sie es, um Ihre Elastic Beanstalk Beanstalk-Umgebungen zu erstellen, zu überwachen und zu verwalten. Wenn Sie mehrere Umgebungen für Ihre Anwendung ausführen, kann die EB CLI in Git integriert werden, sodass Sie jede Umgebung einer anderen Git-Branche zuordnen können.

Themen

- QuickStart: Eine Java-Anwendung auf Elastic Beanstalk bereitstellen
- QuickStart: Stellen Sie eine Java-JSP-Webanwendung für Tomcat auf Elastic Beanstalk bereit
- Einrichten der Java-Entwicklungsumgebung
- Weitere Elastic Beanstalk Beanstalk-Beispielanwendungen und Tutorials für Java
- Elastic Beanstalk Tomcat-Plattform verwenden
- Elastic Beanstalk Java SE-Plattform verwenden
- Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Java Elastic Beanstalk Beanstalk-Umgebung
- Java-Tools und -Ressourcen

QuickStart: Eine Java-Anwendung auf Elastic Beanstalk bereitstellen

Dieses QuickStart Tutorial führt Sie durch den Prozess der Erstellung einer Java-Anwendung und deren Bereitstellung in einer AWS Elastic Beanstalk Umgebung.

> ⚠ Nicht für den Produktionsgebrauch
>
> Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine Java-Anwendung
- Schritt 2: Führen Sie Ihre Anwendung lokal aus
- Schritt 3: Stellen Sie Ihre Java-Anwendung mit der EB CLI bereit
- Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus
- Schritt 5: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center -Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

EB-CLI

Dieses Tutorial verwendet die Befehlszeilenschnittstelle Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Java und Maven

Wenn Sie Amazon Corretto nicht auf Ihrem lokalen Computer installiert haben, können Sie es installieren, indem Sie den Installationsanweisungen im Amazon Corretto-Benutzerhandbuch folgen.

Überprüfen Sie Ihre Java-Installation, indem Sie den folgenden Befehl ausführen.

```
~$ java -version
```

Dieses Tutorial verwendet Maven. Folgen Sie den Download - und Installationsanweisungen auf der Apache Maven Project-Website. Weitere Informationen zu Maven finden Sie im Maven Users Center auf der Apache Maven Project-Website.

Überprüfen Sie Ihre Maven-Installation, indem Sie den folgenden Befehl ausführen.

```
~$ mvn -v
```

Schritt 1: Erstellen Sie eine Java-Anwendung

Erstellen Sie ein Projektverzeichnis.

```
~$ mkdir eb-java
~$ cd eb-java
```

Erstellen Sie anschließend eine Anwendung, die Sie mit Elastic Beanstalk bereitstellen. Wir werden einen „Hello World“ RESTful -Webservice erstellen.

In diesem Beispiel wird das Spring Boot-Framework verwendet. Diese Anwendung öffnet einen Listener auf Port 5000. Elastic Beanstalk leitet Anfragen standardmäßig an Port 5000 an Ihre Anwendung weiter.

Erstellen Sie die folgenden Dateien:

Diese Datei erstellt eine einfache Spring Boot-Anwendung.

Example~/eb-java/src/main/java/com/example/Application.java

```
package com.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

Diese Datei erstellt ein Mapping, das eine Zeichenfolge zurückgibt, die wir hier definieren.

Example~/eb-java/src/main/java/com/example/Controller.java

```
package com.example;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class Controller {

    @GetMapping("/")
    public String index() {
       return "Hello Elastic Beanstalk!";
    }
}
```

Diese Datei definiert die Maven-Projektkonfiguration.

Example~/eb-java/pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
```

```
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.2.3</version>
  </parent>

  <groupId>com.example</groupId>
  <artifactId>BeanstalkJavaExample</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <java.version>21</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-rest</artifactId>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>

</project>
```

Diese Eigenschaftendatei überschreibt den Standardport 5000. Dies ist der Standardport, an den Elastic Beanstalk Traffic für Java-Anwendungen sendet.

Example~/eb-java/application.properties

```
server.port=5000
```

Schritt 2: Führen Sie Ihre Anwendung lokal aus

Package Sie Ihre Anwendung mit dem folgenden Befehl:

```
~/eb-java$ mvn clean package
```

Führen Sie Ihre Anwendung lokal mit dem folgenden Befehl aus:

```
~/eb-java$ java -jar target/BeanstalkJavaExample-1.0-SNAPSHOT.jar
```

Während die Anwendung ausgeführt wird, navigieren Sie `http://127.0.0.1:5000/` in Ihrem Browser zu. Du solltest den Text „Hello Elastic Beanstalk!“ sehen.

Schritt 3: Stellen Sie Ihre Java-Anwendung mit der EB CLI bereit

Bevor wir Ihre Java-Anwendung auf Elastic Beanstalk bereitstellen, löschen wir die Build-Anwendung aus Ihrem Verzeichnis und erstellen ein Buildfile und ein Procfile, um zu kontrollieren, wie die Anwendung in Ihrer Elastic Beanstalk Beanstalk-Umgebung erstellt und ausgeführt wird.

Um die Anwendungsbereitstellung vorzubereiten und zu konfigurieren

1. Reinigen Sie die erstellte Anwendung.

   ```
   ~/eb-java$ mvn clean
   ```

2. Erstellen Sie Ihr `Buildfile`.

   Example~/eb-java/Buildfile

   ```
   build: mvn clean package
   ```

   Dies `Buildfile` gibt den Befehl an, mit dem Ihre Anwendung erstellt wurde. Wenn Sie a `Buildfile` für eine Java-Anwendung nicht angeben, versucht Elastic Beanstalk nicht, Ihre Anwendung zu erstellen.

3. Erstellen Sie Ihr `Procfile`.

   Example~/eb-java/Procfile

   ```
   web: java -jar target/BeanstalkJavaExample-1.0-SNAPSHOT.jar
   ```

Dies `Procfile` gibt den Befehl an, mit dem Ihre Anwendung ausgeführt wird. Wenn Sie a `Procfile` für eine Java-Anwendung nicht angeben, geht Elastic Beanstalk davon aus, dass sich im Stammverzeichnis Ihres Quellpakets eine JAR-Datei befindet, und versucht, sie mit dem `java -jar` Befehl auszuführen.

Nachdem Sie nun die Konfigurationsdateien zum Erstellen und Starten Ihrer Anwendung eingerichtet haben, können Sie sie bereitstellen.

Um eine Umgebung zu erstellen und Ihre Java-Anwendung bereitzustellen

1. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

   ```
   ~/eb-java eb init -p corretto java-tutorial --region us-east-2

   Application java-tutorial has been created.
   ```

   Dieser Befehl erstellt eine Anwendung mit dem Namen `java-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der neuesten Version der Java-Plattform erstellt werden.
2. (Optional) Führen Sie eb init erneut aus, um ein Standardschlüsselpaar für die SSH-Verbindung zur EC2-Instance (auf der die Anwendung ausgeführt wird) zu konfigurieren.

   ```
   ~/eb-java$ eb init
   Do you want to set up SSH for your instances?
   (y/n): y
   Select a keypair.
   1) my-keypair
   2) [ Create new KeyPair ]
   ```

   Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein Schlüsselpaar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.
3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erstellt automatisch eine Zip-Datei für Ihre Anwendung und startet sie auf Port 5000.

   ```
   ~/eb-java$ eb create java-env
   ```

Es dauert etwa fünf Minuten, bis Elastic Beanstalk Ihre Umgebung erstellt hat.

Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, öffnen Sie Ihre Website mit. eb open

```
~/eb-java eb open
```

Herzlichen Glückwunsch! Sie haben eine Java-Anwendung mit Elastic Beanstalk bereitgestellt! Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet.

Schritt 5: Bereinigen

Sie können Ihre Umgebung beenden, wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind. Elastic Beanstalk beendet alle AWS Ressourcen, die mit Ihrer Umgebung verknüpft sind.

Führen Sie den folgenden Befehl aus, um Ihre Elastic Beanstalk Beanstalk-Umgebung mit der EB-CLI zu beenden.

```
~/eb-java$ eb terminate
```

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt. Es dient als einfache Beispielanwendung mit einer einzigen EC2-Instance, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 instance (EC2-Instance) – Eine virtuelle Amazon EC2-Maschine zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 zugelassen wird. Mit dieser

Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Web-App erreichen. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.

- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen. Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

> **Probieren Sie weitere Tutorials aus**
>
> Wenn Sie andere Tutorials mit anderen Beispielanwendungen ausprobieren möchten, finden Sie weitere Informationen unterBeispielanwendungen und Tutorials.

Wenn Sie eine oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, Java-Anwendungen lokal zu entwickeln und auszuführen, finden Sie weitere Informationen unterEinrichten der Java-Entwicklungsumgebung.

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

QuickStart: Stellen Sie eine Java-JSP-Webanwendung für Tomcat auf Elastic Beanstalk bereit

Dieses Tutorial führt Sie durch den Prozess der Erstellung einer einfachen Java-Webanwendung mithilfe von JavaServer Pages ()JSPs. Wenn Sie mehrere Webanwendungen in Form von WAR-Dateien in einer einzigen Elastic Beanstalk Beanstalk-Umgebung bündeln möchten, finden Sie weitere Informationen unter. Bündeln mehrerer WAR-Dateien für Tomcat-Umgebungen

> ⚠ Nicht für Produktionszwecke
>
> Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine Java-JSP-Anwendung
- Schritt 2: Stellen Sie Ihre Java-JSP-Anwendung mit der EB CLI bereit
- Schritt 3: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus
- Schritt 4: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center - Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

EB-CLI

Dieses Tutorial verwendet die Befehlszeilenschnittstelle Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Schritt 1: Erstellen Sie eine Java-JSP-Anwendung

Erstellen Sie ein Projektverzeichnis.

```
~$ mkdir eb-tomcat
~$ cd eb-tomcat
```

Erstellen Sie anschließend eine Anwendung, die Sie mit Elastic Beanstalk bereitstellen. Wir werden eine „Hello World“ -Webanwendung erstellen.

Erstellen Sie eine einfache JSP-Datei mit dem Namen`index.jsp`.

Example~/eb-tomcat/index.jsp

```
<html>
  <body>
    <%out.println("Hello Elastic Beanstalk!");%>
  </body>
</html>
```

Schritt 2: Stellen Sie Ihre Java-JSP-Anwendung mit der EB CLI bereit

Führen Sie die folgenden Befehle aus, um eine Elastic Beanstalk Beanstalk-Umgebung für diese Anwendung zu erstellen.

Um eine Umgebung zu erstellen und Ihre Java-JSP-Anwendung bereitzustellen

1. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

   ```
   ~/eb-tomcat$ eb init -p tomcat tomcat-tutorial --region us-east-2
   ```

   Dieser Befehl erstellt eine Anwendung mit dem Namen `tomcat-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der neuesten Tomcat-Plattformversion erstellt werden.

2. (Optional) Führen Sie eb init erneut aus, um ein Standardschlüsselpaar für die SSH-Verbindung zur EC2-Instance (auf der die Anwendung ausgeführt wird) zu konfigurieren.

   ```
   ~/eb-go$ eb init
   Do you want to set up SSH for your instances?
   (y/n): y
   Select a keypair.
   1) my-keypair
   2) [ Create new KeyPair ]
   ```

   Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein Schlüsselpaar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erstellt automatisch eine Zip-Datei für Ihre Anwendung und startet sie auf Port 5000.

   ```
   ~/eb-tomcat$ eb create tomcat-env
   ```

   Es dauert etwa fünf Minuten, bis Elastic Beanstalk Ihre Umgebung erstellt hat.

Schritt 3: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, öffnen Sie Ihre Website mit. eb open

```
~/eb-tomcat$ eb open
```

Herzlichen Glückwunsch! Sie haben eine Java-JSP-Anwendung mit Elastic Beanstalk bereitgestellt! Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet.

Schritt 4: Bereinigen

Sie können Ihre Umgebung beenden, wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind. Elastic Beanstalk beendet alle AWS Ressourcen, die mit Ihrer Umgebung verknüpft sind.

Führen Sie den folgenden Befehl aus, um Ihre Elastic Beanstalk Beanstalk-Umgebung mit der EB-CLI zu beenden.

```
~/eb-tomcat$ eb terminate
```

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt. Es dient als einfache Beispielanwendung mit einer einzigen EC2-Instance, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 instance (EC2-Instance) – Eine virtuelle Amazon EC2-Maschine zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 zugelassen wird. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Web-App erreichen. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.

- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen. Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

> **Probieren Sie weitere Tutorials aus**
>
> Wenn Sie andere Tutorials mit anderen Beispielanwendungen ausprobieren möchten, finden Sie weitere Informationen unterBeispielanwendungen und Tutorials.

Wenn Sie eine oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, mit der Entwicklung und Ausführung von Java-Anwendungen in einem lokalen Tomcat-Webcontainer zu beginnen, finden Sie weitere Informationen unterEinrichten der Java-Entwicklungsumgebung.

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

Einrichten der Java-Entwicklungsumgebung

Dieses Thema enthält Anweisungen zum Einrichten einer Java-Entwicklungsumgebung, um Ihre Anwendung lokal zu testen, bevor Sie sie bereitstellen AWS Elastic Beanstalk. Außerdem wird auf Websites verwiesen, die Installationsanweisungen für nützliche Tools bereitstellen.

Sections

- Installieren des Java Development Kits
- Installieren eines Webcontainers
- Herunterladen von Bibliotheken
- Das AWS SDK for Java installieren
- Installieren einer IDE oder eines Texteditors

Installieren des Java Development Kits

Installieren des Java Development Kits (JDK). Falls Sie keine Version bevorzugen, wählen Sie die neueste aus. Laden Sie das JDK unter oracle.com herunter.

Das JDK enthält den Java-Compiler, mit dem Sie Quelldateien in Klassendateien erstellen können, die sich auf einem Elastic Beanstalk-Webserver ausführen lassen.

Installieren eines Webcontainers

Wenn Sie noch keinen anderen Webcontainer oder Framework haben, installieren Sie eine Version von Tomcat, die Elastic Beanstalk für Ihr Amazon Linux-Betriebssystem unterstützt. Eine Liste der aktuellen Versionen von Apache Tomcat, die Elastic Beanstalk unterstützt, finden Sie unter Tomcat im Dokument Platforms.AWS Elastic Beanstalk Laden Sie die Tomcat-Version, die für Ihre Umgebung gilt, von der Apache Tomcat-Website herunter.

Herunterladen von Bibliotheken

Elastic Beanstalk-Plattformen enthalten standardmäßig nur wenige Bibliotheken. Laden Sie die Bibliotheken, die von der Anwendung genutzt werden, herunter und speichern Sie diese im Projektordner, um sie im Quell-Bundle der Anwendung bereitzustellen.

Wenn Sie Tomcat lokal installiert haben, können Sie die Servlet-API und die JavaServer Pages (JSP) -API-Bibliotheken aus dem Installationsordner kopieren. Wenn die Bereitstellung für eine Tomcat-Plattformversion erfolgt, ist es nicht nötig, diese Dateien in das Quell-Bundle einzubinden. Allerdings müssen Sie zur Kompilierung aller Klassen, die sie verwenden, im `classpath` vorhanden sein.

JUnit, Google Guava und Apache Commons bieten mehrere nützliche Bibliotheken. Weitere Informationen finden Sie auf den entsprechenden Homepages:

- Download JUnit
- Google Guava herunterladen
- Apache Commons herunterladen

Das AWS SDK for Java installieren

Wenn Sie AWS Ressourcen von Ihrer Anwendung aus verwalten müssen, installieren Sie das AWS SDK for Java. Mit dem können Sie beispielsweise Amazon DynamoDB (DynamoDB) verwenden AWS SDK für Java, um den Sitzungsstatus von Apache Tomcat-Anwendungen auf mehreren Webservern gemeinsam zu nutzen. Weitere Informationen finden Sie unter Tomcat-Sitzungsstatus mit Amazon DynamoDB verwalten in der Dokumentation zum AWS SDK for Java.

Weitere Informationen und Installationsanleitungen finden Sie auf der AWS Homepage SDK für Java.

Installieren einer IDE oder eines Texteditors

Integrierte Entwicklungsumgebungen (IDEs) bieten eine Vielzahl von Funktionen, die die Anwendungsentwicklung erleichtern. Falls Sie noch keine IDE für die Java-Entwicklung eingesetzt haben, probieren Sie Eclipse und IntelliJ aus. Dann sehen Sie, welche IDE für Sie am besten geeignet ist.

- Eclipse IDE for Java EE Developers installieren
- IntelliJ installieren

Mit einer IDE werden dem Projektordner möglicherweise Dateien hinzugefügt, die Sie unter Umständen nicht an die Quellüberwachung übergeben möchten. Damit diese Dateien nicht an die Quellüberwachung übergeben werden, setzen Sie `.gitignore` oder die Entsprechung Ihres Quellüberwachungstools ein.

Wenn Sie gerade mit dem Programmieren beginnen und nicht alle Funktionen einer IDE benötigen, können Sie installing Sublime Text (Installieren von Sublime Text) auswählen.

> **Note**
>
> Am 31. Mai 2023 hat er das Ende seiner Lebensdauer AWS Toolkit for Eclipseerreicht und wird nicht mehr von unterstützt AWS. Weitere Informationen zum Ende des Lebenszyklus von finden Sie in der AWS Toolkit for Eclipse Datei README.md im Repository. AWS Toolkit for Eclipse GitHub

Weitere Elastic Beanstalk Beanstalk-Beispielanwendungen und Tutorials für Java

Dieser Abschnitt enthält zusätzliche Anwendungen und Tutorials. Die QuickStart für Java auf Tomcat Themen QuickStart für Java und weiter oben in diesem Thema führen Sie durch das Starten einer Java-Beispielanwendung mit der EB CLI.

Für den Einstieg in Java-Anwendungen benötigen Sie lediglich ein Anwendungsquellpaket AWS Elastic Beanstalk, das Sie als erste Anwendungsversion hochladen und in einer Umgebung bereitstellen können. Im Rahmen der Umgebungserstellung werden von Elastic Beanstalk alle erforderlichen AWS Ressourcen für eine skalierbare Webanwendung zugewiesen.

Starten einer Umgebung mit einer Java-Beispielanwendung

Elastic Beanstalk bietet einseitige Beispielanwendungen für jede Plattform sowie komplexere Beispiele, die die Verwendung zusätzlicher AWS Ressourcen wie Amazon RDS und sprach- oder plattformspezifische Funktionen und zeigen. APIs

Die einzelnen Seitenbeispiele haben den gleichen Code, den Sie auch erhalten, wenn Sie eine Umgebung erstellen, ohne Ihren eigenen Quellcode anzugeben. Die komplexeren Beispiele werden auf gehostet GitHub und müssen möglicherweise vor der Bereitstellung in einer Elastic Beanstalk Beanstalk-Umgebung kompiliert oder erstellt werden.

Beispiele

Name	Unterstützte Versionen	Umgebungstyp	Quelle	Beschreibung
Tomc (einz Seite	Alle Tomcat mit Corretto-Branches der Plattform	Webs Worke	tomcat.zip	Tomcat-Webanwendung mit einer Seite (`index.jsp`), die im Stamm der Website angezeigt wird. Für Worker-Umgebungen enthält dieses Beispiel eine `cron.yaml` -Datei, mit der eine geplante Aufgabe konfiguriert wird, die `scheduled.jsp` einmal pro Minute aufruft. Wenn `scheduled.jsp` aufgerufen wird, schreibt sie in eine Protokolldatei unter `/tmp/sample-app.log` . Schließlich ist eine Konfigurationsdatei in `.ebextensions` enthalten, die die Protokolle aus `/tmp/` in die Standorte kopiert, die von Elastic Beanstalk gelesen werden, wenn Sie Umgebungsprotokolle anfordern. Wenn Sie in einer Umgebung, auf der dieses Beispiel ausgeführt wird, X-Ray-Integration aktivieren, zeigt diese Anwendung zusätzlichen Inhalt hinsichtlich X-Ray und bietet eine Option zum Generieren von Debug-Informationen, die Sie in der X-Ray-Konsole anzeigen können.
Corre (einz Seite	Corretto 11 Corretto 8	Webs	corretto.zip	Corretto-Anwendung mit `Buildfile` - und `Procfile`-Konfigurationsdateien. Wenn Sie in einer Umgebung, auf der dieses Beispiel ausgeführt wird, X-Ray-Integration aktivieren, zeigt diese Anwendung zusätzlichen Inhalt hinsichtlich X-Ray und

Name	Unterstützte Versionen	Umgebungstyp	Quelle	Beschreibung
				bietet eine Option zum Generieren von Debug-Informationen, die Sie in der X-Ray-Konsole anzeigen können.

Name	Unterstützte Versionen	Umgebungstyp	Quelle	Beschreibung
Score	Java 8	Webs	Klonen Sie das Repo auf .com GitHub	Scorekeep ist eine RESTful Web-API, die das Spring-Framework verwendet, um eine Schnittstelle für die Erstellung und Verwaltung von Benutzern, Sitzungen und Spielen bereitzustellen. Die API ist mit einer Angular 1.5-Web-App gebündelt, von der die API über HTTP eingesetzt wird. Die Anwendung nutzt Funktionen der Java SE-Plattform, um Abhängigkeiten herunterzuladen und auf Instances zu erstellen, und minimiert so die Größe des Quell-Bundles. Die Anwendung umfasst auch nginx-Konfigurationsdateien, mit denen die Standardkonfiguration überschrieben wird, sodass die Bereitstellung der Frontend-Webanwendung statisch auf Port 80 über den Proxy erfolgt und Anforderungen über Pfade unter `/api` an die auf `localhost:5000` ausgeführte API weitergeleitet werden. Scorekeep enthält auch eine `xray`-Verzweigung, die zeigt, wie eine Java-Anwendung für die Verwendung mit AWS X-Ray instrumentiert wird. Es zeigt die Instrumentierung eingehender HTTP-Anfragen mit einem Servlet-Filter, die automatische und manuelle AWS SDK-Client-Instrumentierung, die Rekorderkonfiguration und die Instrumentierung ausgehender HTTP-Anfragen und SQL-Clients. Anweisungen finden Sie in der Readme-Datei oder verwenden Sie das AWS X-Ray

Name	Unterstützte Versionen	Umgebungstyp	Quelle	Beschreibung
				Tutorial Erste Schritte, um die Anwendung mit X-Ray auszuprobieren.
Does it Have Snak	Tomcat 8 mit Java 8	Webs	Klonen Sie das Repo auf .com GitHub	Hat es Schlangen? ist eine Tomcat-Webanwendung, die die Verwendung von Elastic Beanstalk Beanstalk-Konfigurationsdateien, Amazon RDS, JDBC, PostgreSQL, Servlets, Simple Tag Support, Tag Files JSPs, Log4J, Bootstrap und Jackson zeigt. Der Quellcode für dieses Projekt enthält ein minimales Build-Skript, das die Servlets und Modelle in Klassendateien kompiliert und die erforderlichen Dateien in ein Webarchiv verpackt, das Sie in einer Elastic Beanstalk-Umgebung bereitstellen können. Vollständige Anweisungen dazu finden Sie in der Readme-Datei im Projekt-Repository.
Locus Load Gene	Java 8	Webs	GitHubKlonen Sie das Repo auf .com	Webanwendung, die Sie verwenden können, um eine andere laufende Webanwendung in einer Elastic Beanstalk-Umgebung zu testen. Zeigt die Verwendung von `Buildfile` - und `Procfile`-Dateien, DynamoDB und Locust, ein Open-Source-Tool für Belastungstests.

Laden Sie eine Beispielanwendung herunter und stellen Sie diese in Elastic Beanstalk bereit. Führen Sie dazu die folgenden Schritte aus:

Um eine Umgebung mit einer Anwendung (Konsole) zu starten

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Applications (Anwendungen). Wählen Sie eine bestehende Anwendung in der Liste aus. Sie können sich auch dafür entscheiden, eine zu erstellen, indem Sie den Anweisungen unter folgenVerwalten von -Anwendungen.
3. Wählen Sie auf der Seite mit der Anwendungsübersicht die Option Umgebung erstellen aus.

   Dadurch wird der Assistent Create environment (Umgebung erstellen) gestartet. Der Assistent enthält eine Reihe von Schritten, mit denen Sie eine neue Umgebung erstellen können.
4. Wählen Sie für die Umgebungsebene die Webserver-Umgebung oder die Umgebungsebene Worker-Umgebung aus. Nach der Erstellung lässt sich die Umgebungsebene nicht mehr ändern.

   > **Note**
   >
   > Die .NET-Plattform auf Windows Server unterstützt die Worker-Umgebungsebene nicht.

   Die Standardfelder für Anwendungsinformationen basieren auf der Anwendung, die Sie zuvor ausgewählt haben.

   In der Gruppe Umgebungsinformationen wird der Umgebungsname standardmäßig auf der Grundlage des Anwendungsnamens voreingestellt. Wenn Sie einen anderen Umgebungsnamen bevorzugen, können Sie einen anderen Wert in das Feld eingeben. Sie können optional einen Domainnamen eingeben. Andernfalls generiert Elastic Beanstalk automatisch einen Wert. Sie können optional auch eine Beschreibung der Umgebung eingeben.
5. Wählen Sie in Platform (Plattform) die Plattform und die Plattformbranche aus, die/der mit der Sprache Ihrer Anwendung übereinstimmt.

   > **Note**
   >
   > Elastic Beanstalk unterstützt mehrere Versionen für die meisten Plattformen, die aufgelistet sind. Standardmäßig wählt die Konsole die empfohlene Version für die gewählte Plattform und Plattformvariante. Wenn Ihre Anwendung eine andere Version erfordert, können Sie diese hier auswählen. Informationen zu unterstützten Plattformversionen finden Sie unter the section called “Unterstützte Plattformen”.

6. Beim Anwendungscode haben Sie mehrere Möglichkeiten, fortzufahren.
   - Um die Standard-Beispielanwendung ohne Angabe des Quellcodes zu starten, wählen Sie Beispielanwendung. Diese Aktion wählt die Einzelseitenanwendung aus, die Elastic Beanstalk für die zuvor ausgewählte Plattform bereitstellt.
   - Wenn Sie eine Beispielanwendung aus diesem Handbuch heruntergeladen haben oder Ihren eigenen Quellcode für eine Anwendung haben, gehen Sie wie folgt vor.
     a. Wählen Sie Code hochladen aus.
     b. Wählen Sie als Nächstes Lokale Datei und dann unter Anwendung hochladen die Option Datei auswählen aus.
     c. Das Betriebssystem Ihres Client-Computers bietet Ihnen eine Oberfläche zur Auswahl der lokalen Datei, die Sie heruntergeladen haben. Wählen Sie die Quellpaketdatei aus und fahren Sie fort.
7. Ihre Wahl für Voreinstellungen hängt von Ihrem Verwendungszweck für die Umgebung ab.
   - Wenn Sie eine Beispielumgebung erstellen, um mehr über Elastic Beanstalk oder eine Entwicklungsumgebung zu erfahren, wählen Sie Single Instance (kostenloses Kontingent berechtigt).
   - Wenn Sie eine Produktionsumgebung oder eine Umgebung erstellen, um mehr über Load Balancing zu erfahren, wählen Sie eine der Optionen für hohe Verfügbarkeit.
8. Wählen Sie Weiter aus.

Um den Servicezugriff zu konfigurieren

Als Nächstes benötigen Sie zwei Rollen. Eine Service-Rolle ermöglicht es Elastic Beanstalk, Ihre EC2-Instances zu überwachen und die Plattform Ihrer Umgebung zu aktualisieren. Eine EC2-Instance-Profilrolle ermöglicht Aufgaben wie das Schreiben von Logs und die Interaktion mit anderen Services.

Um die Service-Rolle zu erstellen oder auszuwählen

1. Wenn Sie zuvor eine Servicerolle erstellt haben und eine vorhandene auswählen möchten, wählen Sie den Wert aus der Dropdownliste Servicerolle aus und überspringen Sie die restlichen Schritte, um eine Servicerolle zu erstellen.
2. Wenn keine Werte für die Servicerolle aufgeführt sind oder Sie eine neue erstellen möchten, fahren Sie mit den nächsten Schritten fort.

3. Wählen Sie für Servicerolle die Option Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS -Service aus.
5. Wählen Sie als Anwendungsfall Elastic Beanstalk — Environment.
6. Wählen Sie Weiter aus.
7. Vergewissern Sie sich, dass die Berechtigungsrichtlinien Folgendes beinhalten, und wählen Sie dann Weiter aus:
   - `AWSElasticBeanstalkEnhancedHealth`
   - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`
8. Wählen Sie Rolle erstellen aus.
9. Kehren Sie zur Registerkarte Dienstzugriff konfigurieren zurück, aktualisieren Sie die Liste und wählen Sie dann die neu erstellte Servicerolle aus.

Um ein EC2-Instanzprofil zu erstellen oder auszuwählen

1. Wenn Sie zuvor ein EC2-Instanzprofil erstellt haben und ein vorhandenes auswählen möchten, wählen Sie den Wert aus der Dropdownliste für das EC2-Instanzprofil aus und überspringen Sie die restlichen Schritte, um ein EC2-Instanzprofil zu erstellen.
2. Wenn keine Werte für das EC2-Instanzprofil aufgeführt sind oder Sie ein neues erstellen möchten, fahren Sie mit den nächsten Schritten fort.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS -Service aus.
5. Wählen Sie als Anwendungsfall Elastic Beanstalk — Compute.
6. Wählen Sie Weiter aus.
7. Vergewissern Sie sich, dass die Berechtigungsrichtlinien Folgendes beinhalten, und wählen Sie dann Weiter aus:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
8. Wählen Sie Rolle erstellen aus.
9. Kehren Sie zur Registerkarte Dienstzugriff konfigurieren zurück, aktualisieren Sie die Liste und wählen Sie dann das neu erstellte EC2-Instanzprofil aus.

Um die Konfiguration und Erstellung Ihrer Anwendung abzuschließen

1. (Optional) Wenn Sie zuvor ein EC2-Schlüsselpaar erstellt haben, können Sie es aus der Dropdownliste für das EC2-Schlüsselpaarfeld auswählen. Sie würden es verwenden, um sich sicher bei der Amazon EC2-Instance anzumelden, die Elastic Beanstalk für Ihre Anwendung bereitstellt. Wenn Sie diesen Schritt überspringen, können Sie jederzeit ein EC2-Schlüsselpaar erstellen und zuweisen, nachdem die Umgebung erstellt wurde. Weitere Informationen finden Sie unter EC2 key pair.
2. Wählen Sie auf der Seite Configure service access (Servicezugriff konfigurieren) die Option Skip to Review (Zur Überprüfung springen).
3. Auf der Seite Review (Überprüfung) wird eine Zusammenfassung all Ihrer Auswahlmöglichkeiten angezeigt.

   Um Ihre Umgebung weiter anzupassen, wählen Sie neben dem Schritt, der alle Elemente enthält, die Sie konfigurieren möchten, die Option Edit (Bearbeiten). Sie können die folgenden Optionen nur während des Erstellens der Umgebung festlegen:

   - Environment name
   - Domainname
   - Version der Plattform
   - Prozessor
   - Load balancer type (Load-Balancer-Typ)
   - Stufe

   Sie können die folgenden Einstellungen nach der Umgebungserstellung zwar ändern, jedoch müssen dafür neue Instances oder andere Ressourcen bereitgestellt werden, deren Anwendung viel Zeit in Anspruch nehmen kann:

   - Instance-Typ, Root-Volume, key pair und AWS Identity and Access Management (IAM-) Rolle
   - Interne Amazon RDS-Datenbank
   - VPC

   Weitere Informationen zu allen verfügbaren Einstellungen finden Sie unter Der Assistent zum Erstellen einer neuen Umgebung.

4. Wählen Sie unten auf der Seite Submit (Senden), um die Erstellung Ihrer neuen Umgebung zu initialisieren.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine gänzlich andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen.

Wenn Sie ein oder zwei Beispielanwendungen bereitgestellt haben, können Sie mit der lokalen Entwicklung und Ausführung von Java-Anwendungen beginnen. Die entsprechenden Informationen zum Einrichten einer Java-Entwicklungsumgebung und zu allen benötigten Tools und Bibliotheken finden Sie im nächsten Abschnitt.

Elastic Beanstalk Tomcat-Plattform verwenden

In diesem Thema wird beschrieben, wie Sie Ihre Java-Anwendungen, die auf der Elastic Beanstalk Tomcat-Plattform ausgeführt werden, konfigurieren, erstellen und ausführen.

Die AWS Elastic Beanstalk Tomcat-Plattform besteht aus einer Reihe von Plattformversionen für Java-Webanwendungen, die in einem Tomcat-Webcontainer ausgeführt werden können. Tomcat läuft hinter einem nginx-Proxy-Server. Jeder Plattformzweig entspricht einer Hauptversion von Tomcat.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das Ändern der Konfiguration einer ausgeführten Umgebung verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie gespeicherte Konfigurationen verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie Konfigurationsdateien einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die Elastic Beanstalk Tomcat-Plattform beinhaltet einen Reverse-Proxy-Server, der Anforderungen an die Anwendung weiterleitet. Sie können Konfigurationsoptionen verwenden, um den Proxy-Server zur Verarbeitung statischer Komponenten aus einem Ordner im Quellcode zu konfigurieren und so die Auslastung für die Anwendung zu reduzieren. In erweiterten Szenarien können Sie eigene

`.conf`-Dateien in das Quell-Bundle einbinden, um die Proxy-Konfiguration von Elastic Beanstalk zu erweitern oder zu überschreiben.

> **Note**
>
> Elastic Beanstalk unterstützt nginx (Standard) und Apache HTTP Server als Proxy-Server auf der Tomcat-Plattform. Wenn Ihre Elastic Beanstalk Tomcat-Umgebung einen Branch der Amazon-Linux-AMI-Plattform (vor Amazon Linux 2) verwendet, haben Sie auch die Möglichkeit, Apache HTTP Server Version 2.2 zu verwenden. Apache (neueste) ist der Standardwert für diese älteren Plattformzweige.
> Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Java-Anwendungen müssen in einem Webanwendungsarchiv (WAR-Datei) mit einer bestimmten Struktur platziert werden. Weitere Informationen zur erforderlichen Struktur und deren Bezug zur Projektverzeichnisstruktur finden Sie unter Strukturieren Ihres Projektordners.

Für die Ausführung von mehreren Anwendungen auf demselben Webserver können Sie in einem einzigen Quell-Bundle mehrere WAR-Dateien bündeln. Jede Anwendung in einem Quell-Bundle mit mehreren WAR-Dateien wird je nach dem WAR-Dateinamen entweder mit dem Stammpfad (`ROOT.war` mit *`myapp`*`.elasticbeanstalk.com/`) oder mit einem direkt darunter liegenden Pfad (`app2.war` mit *`myapp`*`.elasticbeanstalk.com/`*`app2/`*) ausgeführt. In einem Quell-Bundle mit einer einzigen WAR-Datei wird die Anwendung immer mit dem Stammpfad ausgeführt.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe Konfigurationsoptionen.

Details zu den verschiedenen Möglichkeiten für das Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter the section called “Erweitern von Linux-Plattformen”.

Themen

- Konfigurieren Ihrer Tomcat-Umgebung
- Tomcat-Konfigurations-Namespaces
- Bündeln mehrerer WAR-Dateien für Tomcat-Umgebungen
- Strukturieren Ihres Projektordners
- Konfigurieren des Proxy-Server

Konfigurieren Ihrer Tomcat-Umgebung

Die Elastic Beanstalk Tomcat-Plattform bietet einige plattformspezifische Optionen zusätzlich zu den Standard-Optionen, die allen Plattformen gemeinsam sind. Mit diesen Optionen können Sie die JVM (Java Virtual Machine) konfigurieren, die auf den Webservern der Umgebung ausgeführt wird, und die Systemeigenschaften definieren, die Zeichenfolgen mit Konfigurationsinformationen für die Anwendung bereitstellen.

Mit der Elastic Beanstalk-Konsole können Sie die Protokollrotation für Amazon S3 aktivieren und Variablen konfigurieren, die die Anwendung in der Umgebung lesen kann.

So konfigurieren Sie Ihre Tomcat-Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

Container-Optionen

Sie können die folgenden plattformspezifischen Optionen angeben:

- Proxy server (Proxy-Server) – Der Proxy-Server, der in Ihren Umgebungs-Instances verwendet werden soll. Standardmäßig wird nginx verwendet.

JVM-Containeroptionen

Über die Heap-Größe in der JVM wird bestimmt, wie viele Objekte die Anwendung im Speicher erstellen kann, bevor eine garbage collection (Speicherbereinigung) erfolgt. Sie können die Ursprüngliche JVM-Heap-Größe (`-Xms option`) und eine Maximale JVM-Heap-Größe (`-Xmx-` Option) ändern. Wenn Sie einen höheren Wert für die anfängliche Heap-Größe angeben, können mehr Objekte erstellt werden, bevor eine Speicherbereinigung ausgeführt wird. Jedoch dauert es länger, bis der Heap im Rahmen der Garbage Collection (Speicherbereinigung) komprimiert ist. Die maximale Heap-Größe gibt die maximale Größe des Arbeitsspeichers an, die bei einer großen Auslastung von der JVM für die Heap-Erweiterung zugewiesen werden kann.

> **Note**
>
> Der verfügbare Speicher hängt vom Amazon EC2-Instance-Typ ab. Weitere Informationen zu den für die Elastic Beanstalk-Umgebung verfügbaren EC2-Instance-Typen finden Sie unter Instance Types (Instance-Typen) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Der JVM-Heap verfügt über einen Bereich für permanente Generierung, in dem Klassendefinitionen und zugehörige Metadaten gespeichert werden. Um die Größe der permanenten Generierung zu ändern, geben Sie die neue Größe in die Option Maximale PermGen JVM-Größe (`-XX:MaxPermSize`) ein. Diese Einstellung gilt nur für Java 7 und frühere Versionen. Diese Option war in JDK 8 veraltet und wurde durch die Option Size () ersetzt. MaxMetaspace `-XX:MaxMetaspaceSize`

> **Important**
>
> JDK 17 hat die Unterstützung der `-XX:MaxPermSize` Java-Option entfernt. Die Verwendung dieser Option in einer Umgebung, die auf einem Elastic-Beanstalk-Plattformzweig mit Corretto 17 läuft, führt zu einem Fehler. Elastic Beanstalk veröffentlichte am 13. Juli 2023 seinen ersten Plattformzweig, auf dem Tomcat mit Corretto 17 ausgeführt wird. Weitere Informationen finden Sie in den folgenden Ressourcen.
>
> - Website zur Oracle-Java-Dokumentation: Java-Optionen entfernt
> - Website zur Oracle-Java-Dokumentation: Abschnitt Klassenmetadaten in Weitere Überlegungen

Weitere Informationen zu Elastic-Beanstalk-Plattformen und ihren Komponenten finden Sie unter Unterstützte Plattformen im AWS Elastic Beanstalk -Plattformhandbuch.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon-EC2-Instances der Anwendung in den der Anwendung zugeordneten Amazon-S3-Bucket kopiert werden.

Statische Dateien

Um die Leistung zu verbessern, können Sie den Abschnitt Static files (Statische Dateien) benutzen, um den Proxy-Server so zu konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zum Verzeichnis-Mapping fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, anstatt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter the section called “Statische Dateien”.

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon-EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet.

Die Tomcat-Plattform definiert für Tomcat-Umgebungen die Platzhaltereigenschaft `JDBC_CONNECTION_STRING`, um eine Verbindungszeichenfolge an eine externe Datenbank zu übergeben.

> **Note**
>
> Wenn Sie eine RDS DB-Instance an Ihre Umgebung anhängen, konstruieren Sie die JDBC-Verbindungszeichenfolge dynamisch aus den Eigenschaften der Amazon Relational

Database Service (Amazon RDS)-Umgebung, die von Elastic Beanstalk bereitgestellt werden. Verwenden Sie JDBC_CONNECTION_STRING nur für Datenbank-Instances, die nicht mithilfe von Elastic Beanstalk bereitgestellt werden.
Weitere Informationen zur Verwendung von Amazon RDS mit Ihrer Java-Anwendung finden Sie unter Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Java Elastic Beanstalk Beanstalk-Umgebung.

Bei Tomcat-Plattformversionen, die vor dem 26. März 2025 veröffentlicht wurden, kann auf Umgebungsvariablen zugegriffen werden über. `System.getProperty()` Sie könnten beispielsweise eine `API_ENDPOINT` aus einer Variablen benannte Eigenschaft mit dem folgenden Code lesen.

```
String endpoint = System.getProperty("API_ENDPOINT");
```

Tomcat-Plattformversionen, die am oder nach dem 26. März 2025 veröffentlicht wurden, können auch für den `System.getenv` Zugriff auf Klartext-Umgebungsvariablen verwendet werden. Sie können es weiterhin für den Zugriff `System.getProperty` auf Klartext-Umgebungsvariablen verwenden. Umgebungsvariablen, die als Geheimnisse gespeichert sind, sind jedoch nur über `System.getenv` verfügbar. Sie könnten beispielsweise eine Umgebungsvariable lesen, die `API_KEY` mit dem folgenden Code benannt ist.

```
String apiKey = System.getenv("API_KEY");
```

Important

Der zusätzliche `System.getenv()` Zugriff auf Umgebungsvariablen in Versionen der Tomcat-Plattform, die am oder nach dem 26. März 2025 veröffentlicht wurden, kann zu unerwartetem Verhalten in Anwendungen führen, die Umgebungsvariablen Vorrang vor Java-Systemeigenschaften geben, oder wenn explizit von `System.getProperty()` zu gewechselt wird. `System.getenv()`
Da Systemeigenschaften (die über die Befehlszeile übergeben werden) Shell-Escaping für Sonderzeichen erfordern, Umgebungsvariablen jedoch nicht, können Werte anders aufgelöst werden, wenn Umgebungsvariablen anstelle von Java-Systemeigenschaften verwendet werden.
Wenn Ihre Anwendung betroffen ist, sollten Sie Folgendes berücksichtigen:

- Entfernen Sie Escape-Zeichen aus den Eigenschaftswerten Ihrer Umgebung, wenn Sie `System.getenv()`
- Konfiguration Ihrer Anwendung für die explizite Verwendung `System.getProperty()`
- Testen Sie Ihre Anwendung beim Upgrade gründlich, um ein konsistentes Verhalten sicherzustellen

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Tomcat-Konfigurations-Namespaces

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Die Tomcat-Plattform unterstützt neben den unterstützten Optionen für alle Elastic Beanstalk-Umgebungen auch Optionen in den folgenden Namespaces:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions` – Ändern von JVM-Einstellungen. Die Optionen in diesem Namespace entsprechen wie folgt den Optionen in der Managementkonsole:
  - `Xms` – JVM-Befehlszeilenoptionen
  - `JVM Options` – JVM-Befehlszeilenoptionen
- `aws:elasticbeanstalk:environment:proxy` – Wählen Sie den Proxy-Server der Umgebung aus.

In der folgenden Beispielkonfigurationsdatei wird die Verwendung von Tomcat-spezifischen Konfigurationsoptionen veranschaulicht.

Example.ebextensions/tomcat-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:tomcat:jvmoptions:
    Xms: 512m
    JVM Options: '-Xmn128m'
  aws:elasticbeanstalk:application:environment:
```

```
    API_ENDPOINT: mywebapi.zkpexsjtmd.us-west-2.elasticbeanstalk.com
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Die Amazon Linux-AMI (vor Amazon Linux 2) Tomcat-Plattform

Falls Ihre Elastic Beanstalk Tomcat-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/AL2 Plattformbranches sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
> - Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Namespaces für die Tomcat-Konfiguration — Amazon Linux AMI () AL1

Die Tomcat Amazon Linux AMI-Plattform unterstützt zusätzliche Optionen in den folgenden Namespaces:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions` – Zusätzlich zu den oben auf dieser Seite für diesen Namespace erwähnten Optionen unterstützen ältere Amazon Linux-AMI-Plattformversionen auch Folgendes:
  - `XX:MaxPermSize` – Maximale Größe für die permanente JVM-Generierung
- `aws:elasticbeanstalk:environment:proxy` – Konfigurieren Sie nicht nur den Proxy-Server, sondern auch die Antwortkomprimierung.

In der folgenden Beispielkonfigurationsdatei wird die Verwendung der Proxy-Namespace-Konfigurationsoptionen veranschaulicht.

Example.ebextensions/tomcat-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    GzipCompression: 'true'
    ProxyServer: nginx
```

Elastic Beanstalk Beanstalk-Konfigurationsdateien einbeziehen — Amazon Linux AMI () AL1

Um `.ebextensions`-Konfigurationsdateien bereitzustellen, schließen Sie sie in Ihre Anwendungsquelle ein. Fügen Sie für eine einzelne Anwendung `.ebextensions` zu einer komprimierten WAR-Datei hinzu, indem Sie den folgenden Befehl ausführen:

Example

```
zip -ur your_application.war .ebextensions
```

Für eine Anwendung, die mehrere WAR-Dateien benötigt, finden Sie weitere Anweisungen unter Bündeln mehrerer WAR-Dateien für Tomcat-Umgebungen.

Bündeln mehrerer WAR-Dateien für Tomcat-Umgebungen

Wenn Ihre Web-App mehrere Web-Anwendungskomponenten umfasst, können Sie die Bereitstellung vereinfachen und die Betriebskosten senken, indem Sie Komponenten in einer einzigen Umgebung ausführen, anstatt eine separate Umgebung für jede Komponente auszuführen. Diese Strategie ist für leichte Anwendungen effektiv, die nicht viele Ressourcen erfordern, und für Entwicklungs- und Testumgebungen.

Zur Bereitstellung von mehreren Webanwendungen in Ihrer Umgebung kombinieren Sie die WAR-Dateien jeder Komponente zu einem einzelnen Quell-Bundle.

Zum Erstellen eines Quell-Bundle der Anwendung mit mehreren WAR-Dateien organisieren Sie die WAR-Dateien mithilfe der folgenden Struktur.

```
MyApplication.zip
### .ebextensions
```

```
### .platform
### foo.war
### bar.war
### ROOT.war
```

Wenn Sie ein Quellpaket mit mehreren WAR-Dateien in einer AWS Elastic Beanstalk Umgebung bereitstellen, ist jede Anwendung über einen anderen Pfad außerhalb des Stammdomänennamens zugänglich. Das vorausgehende Beispiel enthält drei Anwendungen: `foo`, `bar` und `ROOT`. `ROOT.war` ist ein spezieller Dateiname, der Elastic Beanstalk auffordert, diese Anwendung in der Stamm-Domäne auszuführen, damit die drei Anwendungen unter `http://MyApplication.elasticbeanstalk.com/foo`, `http://MyApplication.elasticbeanstalk.com/bar` und `http://MyApplication.elasticbeanstalk.com` verfügbar sind.

Das Quellpaket kann WAR-Dateien, einen optionalen `.ebextensions`-Ordner und einen optionalen `.platform`-Ordner enthalten. Weitere Informationen zu diesen optionalen Konfigurationsordnern finden Sie unter the section called “Erweitern von Linux-Plattformen”.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
2. Wählen Sie für Platform (Plattform) die Plattform und den Plattformzweig aus, die der von Ihrer Anwendung verwendeten Sprache entsprechen, oder die Docker-Plattform für containerbasierte Anwendungen.
3. Wählen Sie für Application code (Anwendungscode) die Option Upload your code (Eigenen Code hochladen).
4. Wählen Sie Local file (Lokale Datei), wählen Sie Choose file (Datei wählen) und öffnen Sie das Quell-Bundle.
5. Klicken Sie auf Review and launch (Überprüfen und starten).
6. Prüfen Sie die verfügbaren Einstellungen und wählen Sie Create app (Anwendung erstellen) aus.

Weitere Informationen zum Erstellen von Quell-Bundles finden Sie unter Ein Elastic Beanstalk Beanstalk-Anwendungsquellpaket erstellen.

Strukturieren Ihres Projektordners

Damit auf einem Tomcat-Server bereitgestellte kompilierte Java EE (Java Platform Enterprise Edition)-Webanwendungsarchive (WAR-Dateien) genutzt werden können, müssen sie nach bestimmten Richtlinien strukturiert sein. Ihr Projektverzeichnis muss nicht denselben Standards entsprechen, dies wäre aber sinnvoll, da Kompilierung und Paketerstellung vereinfacht werden. Wenn Sie den Projektordner genauso strukturieren wie die WAR-Dateiinhalte, können Sie die Beziehung der Dateien untereinander und deren Verhalten auf dem Webserver leichter nachvollziehen.

In der folgenden empfohlenen Hierarchie wird der Quellcode der Webanwendung im Verzeichnis `src` platziert und so vom Build-Skript und der generierten WAR-Datei separiert.

```
~/workspace/my-app/
|-- build.sh            - Build script that compiles classes and creates a WAR
|-- README.MD           - Readme file with information about your project, notes
|-- ROOT.war            - Source bundle artifact created by build.sh
`-- src                 - Source code folder
    |-- WEB-INF         - Folder for private supporting files
    |   |-- classes     - Compiled classes
    |   |-- lib         - JAR libraries
    |   |-- tags        - Tag files
    |   |-- tlds        - Tag Library Descriptor files
    |   `-- web.xml     - Deployment Descriptor
    |-- com             - Uncompiled classes
    |-- css             - Style sheets
    |-- images          - Image files
    |-- js              - JavaScript files
    `-- default.jsp     - JSP (JavaServer Pages) webpage
```

Die Ordnerinhalte von `src` sind identisch mit den Daten, die Sie packen und auf dem Server bereitstellen, mit Ausnahme des Ordners `com`. Der `com`-Ordner enthält die nicht kompilierten Klassen (`.java`-Dateien). Diese müssen kompiliert und im Verzeichnis `WEB-INF/classes` gespeichert werden, damit der Anwendungscode darauf zugreifen kann.

Das Verzeichnis `WEB-INF` enthält Code und Konfigurationen, die nicht öffentlich auf dem Webserver bereitgestellt werden. Die anderen Ordner im Stamm des Quellverzeichnisses (`css`, `images` und `js`) sind unter dem entsprechenden Pfad öffentlich auf dem Webserver verfügbar.

Das folgende Beispiel ist identisch mit dem vorherigen Projektverzeichnis, weist aber mehr Dateien und Unterverzeichnisse auf. Dieses Beispielprojekt umfasst einfache Tags, Modell- und Support-Klassen sowie eine JSP (Java Server Pages)-Datei für eine `record`-Ressource. Es enthält auch

ein Stylesheet und JavaScript für Bootstrap eine Standard-JSP-Datei und eine Fehlerseite für 404-Fehler.

`WEB-INF/lib` enthält eine JAR-Datei (Java Archive) mit dem JDBC-Treiber (Java Database Connectivity) für PostgreSQL. `WEB-INF/classes` ist leer, da die Klassendateien noch nicht kompiliert wurden.

```
~/workspace/my-app/
|-- build.sh
|-- README.MD
|-- ROOT.war
`-- src
    |-- WEB-INF
    |   |-- classes
    |   |-- lib
    |   |   `-- postgresql-9.4-1201.jdbc4.jar
    |   |-- tags
    |   |   `-- header.tag
    |   |-- tlds
    |   |   `-- records.tld
    |   `-- web.xml
    |-- com
    |   `-- myapp
    |       |-- model
    |       |   `-- Record.java
    |       `-- web
    |           `-- ListRecords.java
    |-- css
    |   |-- bootstrap.min.css
    |   `-- myapp.css
    |-- images
    |   `-- myapp.png
    |-- js
    |   `-- bootstrap.min.js
    |-- 404.jsp
    |-- default.jsp
    `-- records.jsp
```

Erstellen einer WAR-Datei mit einem Shell-Skript

`build.sh` ist ein sehr einfaches Shell-Skript, das Java-Klassen kompiliert, eine WAR-Datei erstellt und diese für lokale Tests in das Tomcat-Verzeichnis `webapps` kopiert:

```
cd src
javac -d WEB-INF/classes com/myapp/model/Record.java
javac -classpath WEB-INF/lib/*:WEB-INF/classes -d WEB-INF/classes com/myapp/model/
Record.java
javac -classpath WEB-INF/lib/*:WEB-INF/classes -d WEB-INF/classes com/myapp/web/
ListRecords.java

jar -cvf ROOT.war *.jsp images css js WEB-INF
cp ROOT.war /Library/Tomcat/webapps
mv ROOT.war ../
```

Die Struktur in der WAR-Datei entspricht der Struktur im Verzeichnis `src` aus dem vorigen Beispiel, mit Ausnahme des Ordners `src/com`. Mit dem Befehl `jar` wird die Datei `META-INF/MANIFEST.MF` automatisch erstellt.

```
~/workspace/my-app/ROOT.war
|-- META-INF
|   `-- MANIFEST.MF
|-- WEB-INF
|   |-- classes
|   |   `-- com
|   |       `-- myapp
|   |           |-- model
|   |           |   `-- Records.class
|   |           `-- web
|   |               `-- ListRecords.class
|   |-- lib
|   |   `-- postgresql-9.4-1201.jdbc4.jar
|   |-- tags
|   |   `-- header.tag
|   |-- tlds
|   |   `-- records.tld
|   `-- web.xml
|-- css
|   |-- bootstrap.min.css
|   `-- myapp.css
|-- images
|   `-- myapp.png
|-- js
|   `-- bootstrap.min.js
|-- 404.jsp
|-- default.jsp
```

```
`-- records.jsp
```

benutze `.gitignore`

Damit keine kompilierten Klassendateien und WAR-Dateien an das Git-Repository übergeben oder diesbezügliche Meldungen beim Ausführen von Git-Befehlen angezeigt werden, fügen Sie die entsprechenden Dateitypen zur Datei `.gitignore` im Projektordner hinzu:

~/workspace/myapp/.gitignore

```
*.zip
*.class
```

Konfigurieren des Proxy-Server

Die Tomcat-Plattform verwendet nginx (die Standardeinstellung) oder Apache HTTP Server als Reverse-Proxy, um Anfragen von Port 80 auf der Instance an Ihren Tomcat Web-Container weiterzuleiten, der auf Port 8080 lauscht. Elastic Beanstalk bietet eine Proxy-Standardkonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Konfigurieren des Proxy-Servers auf Ihrer Plattformversion

Alle AL2023 AL2 /-Plattformen unterstützen eine einheitliche Proxykonfigurationsfunktion. Weitere Informationen zur Konfiguration des Proxyservers auf Ihren Plattformversionen, auf denen AL2023/ ausgeführt wirdAL2, finden Sie unterReverse-Proxy-Konfiguration.

Proxy auf der Amazon Linux AMI (vor Amazon Linux 2) Tomcat-Plattform konfigurieren

Falls Ihre Elastic Beanstalk Tomcat-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/AL2 platform Branches sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
> - Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu

einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Auswahl eines Proxyservers für Ihre Tomcat-Umgebung — Amazon Linux AMI () AL1

Tomcat-Plattformversionen, die auf Amazon Linux-AMIs (vor Amazon Linux 2) basieren, verwenden standardmäßig Apache 2.4 für den Proxy. Sie können Apache 2.2 oder nginx verwenden, indem Sie eine Konfigurationsdatei in Ihren Quellcode einbinden. Im folgenden Beispiel wird Elastic Beanstalk für die Verwendung von nginx konfiguriert.

Example.ebextensions/nginx-proxy.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: nginx
```

Migration von Apache 2.2 auf Apache 2.4 — Amazon Linux AMI () AL1

Wenn Ihre Anwendung für Apache 2.2 entwickelt wurde, lesen Sie diesen Abschnitt, um mehr über die Migration zu Apache 2.4 zu erfahren.

Beginnend mit Tomcat-Plattform Version 3.0.0- Konfigurationen, die mit der Java mit Tomcat-Plattformaktualisierung am 24. Mai 2018 veröffentlicht wurden, ist Apache 2.4 ist der standardmäßige Proxy der Tomcat-Plattform. Die Apache 2.4 `.conf`-Dateien sind häufig, aber nicht vollständig abwärtskompatibel zu denjenigen von Apache 2.2. Elastic Beanstalk enthält standardmäßige `.conf`-Dateien, die einwandfrei mit jeder Apache-Version funktionieren. Wenn Ihre Anwendung die Apache-Konfiguration nicht anpasst, wie unter Erweiterung und Außerkraftsetzung der Apache-Standardkonfiguration — Amazon Linux AMI () AL1 beschrieben , sollte es bei der Migration zu Apache 2.4 keine Probleme geben.

Wenn Ihre Anwendung die Konfiguration von Apache erweitert oder überschreibt, müssen Sie möglicherweise einige Änderungen vornehmen, um zu Apache 2.4 zu migrieren. Weitere Informationen finden Sie unter Upgrading to 2.4 from 2.2 (Upgrade von 2.2 auf 2.4) auf der Website von Apache Software Foundation. Bis zur erfolgreichen Migration auf Apache 2.4 können Sie den Apache 2.2 mit Ihrer Anwendung verwenden, indem Sie die folgende Konfigurationsdatei in Ihren Quellcode aufnehmen.

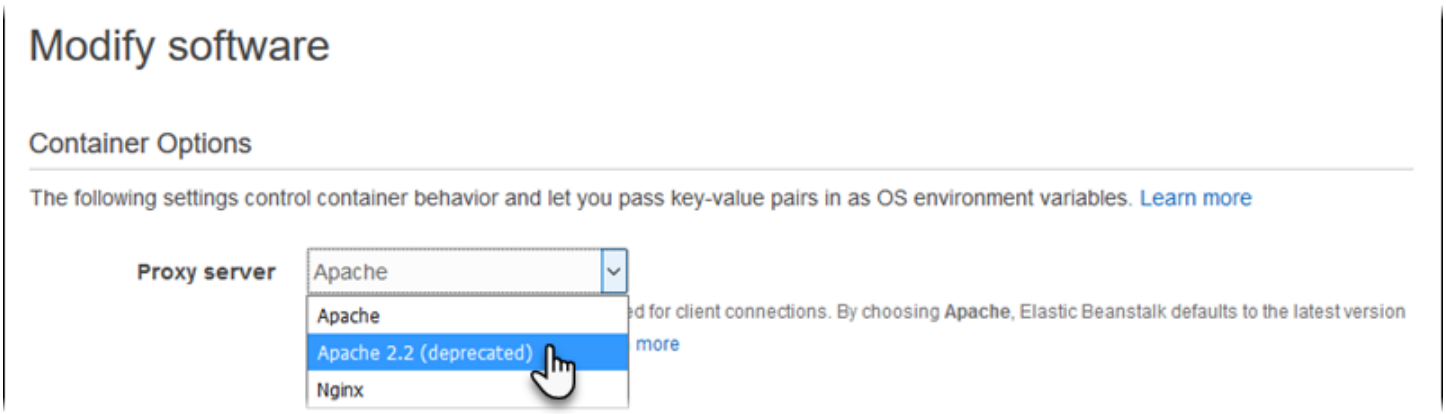
Example.ebeextensions/ .config apache-legacy-proxy

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache/2.2
```

Für eine schnelle Lösung können Sie auch den Proxy-Server in der Elastic Beanstalk-Konsole auswählen.

So wählen Sie den Proxy-Server in Ihrer Tomcat-Umgebung in der Elastic Beanstalk-Konsole aus:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Für Proxy-Server wählen Sie `Apache 2.2 (deprecated)`.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.



Erweiterung und Außerkraftsetzung der Apache-Standardkonfiguration — Amazon Linux AMI () AL1

Sie können die Elastic Beanstalk-Apache-Standardkonfiguration mit Ihren zusätzlichen Konfigurationsdateien erweitern. Alternativ können Sie die Apache-Standardkonfiguration von Elastic Beanstalk vollständig überschreiben.

> **Note**
>
> - Alle Amazon Linux 2-Plattformen unterstützen eine einheitliche Proxykonfigurations-Features. Einzelheiten zur Konfiguration des Proxyservers auf Tomcat-Plattformversionen, auf denen Amazon Linux 2 ausgeführt wird, finden Sie unterReverse-Proxy-Konfiguration.
> - Wenn Sie Ihre Elastic Beanstalk-Anwendung auf eine Amazon Linux 2-Plattform migrieren, lesen Sie auch die Informationen unter the section called “Migrieren Sie zu AL2 023/ AL2”.

Um die Apache-Standardkonfiguration von Elastic Beanstalk zu erweitern, fügen Sie `.conf`-Konfigurationsdateien zum Ordner `.ebextensions/httpd/conf.d` im Quell-Bundle der Anwendung hinzu. Die Apache-Konfiguration von Elastic Beanstalk fügt `.conf`-Dateien automatisch zu diesem Ordner hinzu.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd
|      -- conf.d
|         -- myconf.conf
|         -- ssl.conf
-- index.jsp
```

Beispielsweise wird mit der folgenden Apache 2.4-Konfiguration ein Listener auf Port 5000 hinzugefügt:

Example. ebextensions/httpd/conf.d/port5000.conf

```
listen 5000
<VirtualHost *:5000>
  <Proxy *>
    Require all granted
  </Proxy>
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-error_log
</VirtualHost>
```

Um die Apache-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, fügen Sie bei eine Konfiguration in das Quellpaket ei `.ebextensions/httpd/conf/httpd.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- httpd
|       `-- conf
|           `-- httpd.conf
`-- index.jsp
```

Wenn Sie die Apache-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeilen zu `httpd.conf` hinzu, damit die Konfigurationen von Elastic Beanstalk für Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk, Antwortkomprimierungen und statische Dateien einbezogen werden.

```
IncludeOptional conf.d/*.conf
IncludeOptional conf.d/elasticbeanstalk/*.conf
```

Wenn Ihre Umgebung Apache 2.2 als Proxy verwendet, ersetzen Sie die `IncludeOptional`-Anweisungen durch `Include`. Einzelheiten zum Verhalten dieser beiden Direktiven in den beiden Apache-Versionen finden Sie unter Include in Apache 2.4, IncludeOptional in Apache 2.4 und Include in Apache 2.2.

> **Note**
>
> Um den Standard-Listener auf Port 80 zu überschreiben, binden Sie eine Datei mit dem Namen `00_application.conf` bei `.ebextensions/httpd/conf.d/elasticbeanstalk/` ein, um die Konfiguration von Elastic Beanstalk zu überschreiben.

Ein funktionierendes Beispiel für die Elastic Beanstalk-Standardkonfigurationsdatei finden Sie unter `/etc/httpd/conf/httpd.conf` auf einer Umgebungs-Instance. Alle Dateien im Ordner `.ebextensions/httpd` des Quell-Bundles werden bei Bereitstellungen in `/etc/httpd` kopiert.

Erweiterung der Standard-Nginx-Konfiguration — Amazon Linux AMI () AL1

Um die nginx-Standardkonfiguration von Elastic Beanstalk zu überschreiben, fügen Sie `.conf`-Konfigurationsdateien zum Ordner `.ebextensions/nginx/conf.d/` im Quell-Bundle der

Anwendung hinzu. Die nginx-Konfiguration von Elastic Beanstalk fügt `.conf`-Dateien automatisch zu diesem Ordner hinzu.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- conf.d
|           |-- elasticbeanstalk
|           |   `-- my-server-conf.conf
|           `-- my-http-conf.conf
`-- index.jsp
```

Dateien mit der Erweiterung ".conf" im Ordner `conf.d` sind im `http`-Block der Standardkonfiguration enthalten. Dateien im Ordner `conf.d/elasticbeanstalk` befinden sich im `server`-Block innerhalb des `http`-Blocks.

Um die nginx-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, fügen Sie bei eine Konfiguration in das Quellpaket ei `.ebextensions/nginx/nginx.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- nginx.conf
`-- index.jsp
```

Hinweise

- Wenn Sie die nginx-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeile zum `server`-Block der Konfiguration hinzu, damit die Konfigurationen von Elastic Beanstalk für den Listener auf Port 80, Antwortkomprimierungen und statische Dateien einbezogen werden.

  ```
  include conf.d/elasticbeanstalk/*.conf;
  ```

- Um den Standard-Listener auf Port 80 zu überschreiben, binden Sie eine Datei mit dem Namen `00_application.conf` bei `.ebextensions/nginx/conf.d/elasticbeanstalk/` ein, um die Konfiguration von Elastic Beanstalk zu überschreiben.

- Fügen Sie zudem die folgende Zeile in den `http`Block der Konfiguration ein, damit die Konfigurationen von Elastic Beanstalk für Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk und Protokollierung einbezogen werden.

```
    include       conf.d/*.conf;
```

Ein funktionierendes Beispiel für die Elastic Beanstalk-Standardkonfigurationsdatei finden Sie unter `/etc/nginx/nginx.conf` auf einer Umgebungs-Instance. Alle Dateien im Ordner `.ebextensions/nginx` des Quell-Bundles werden bei Bereitstellungen in `/etc/nginx` kopiert.

Elastic Beanstalk Java SE-Plattform verwenden

In diesem Thema wird beschrieben, wie Sie Ihre Java-Anwendungen, die auf der AWS Elastic Beanstalk Java SE-Plattform ausgeführt werden, konfigurieren, erstellen und ausführen.

Die Elastic Beanstalk Java SE-Plattform besteht aus einer Reihe von Plattformversionen für Java-Webanwendungen, die aus einer kompilierten JAR-Datei eigenständig ausgeführt werden können. Sie können Ihre Anwendung lokale kompilieren oder den Quellcode mit einem Build-Skript hochladen, um die Anwendung auf der Instance zu kompilieren. Die Versionen der Java SE-Plattform sind in Plattformzweige gruppiert, von denen jede einer Hauptversion von Java entspricht.

> **Note**
>
> Elastic Beanstalk analysiert die JAR-Datei Ihrer Anwendung nicht. Speichern Sie Dateien, die Elastic Beanstalk benötigt, außerhalb der JAR-Datei auf. Beziehen Sie beispielsweise die `cron.yaml`-Datei einer Worker-Umgebung im Stammverzeichnis des Quell-Bundles Ihrer Anwendung neben der JAR-Datei ein.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das Ändern der Konfiguration einer ausgeführten Umgebung verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie gespeicherte Konfigurationen verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie Konfigurationsdateien einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung

erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die Java SE-Plattform in Elastic Beanstalk umfasst einen nginx-Server, der als Reverse-Proxy-Server fungiert und zwischengespeicherte statische Inhalte liefert sowie Anforderungen an die Anwendung weiterleitet. Die Plattform stellt auch Konfigurationsoptionen bereit, um den Proxy-Server zur Verarbeitung statischer Komponenten aus einem Ordner im Quellcode zu konfigurieren und so die Auslastung für die Anwendung zu reduzieren. In erweiterten Szenarien können Sie eigene CONF-Dateien in das Quell-Bundle einbinden, um die Proxy-Konfiguration von Elastic Beanstalk zu erweitern oder zu überschreiben.

Wenn Sie nur eine einzelne JAR-Datei für Ihre Anwendungsquelle bereitstellen (allein, nicht innerhalb eines Quellpakets), benennt Elastic Beanstalk die JAR-Datei in `application.jar` um und führt sie dann mit `java -jar application.jar` aus. Schließen Sie eine optionale Procfile-Datei in Ihr Quell-Bundle ein, um die Prozesse zu konfigurieren, die auf den Server-Instances in Ihrer Umgebung ausgeführt werden. Wenn Ihr Quell-Bundle mehrere JAR-Dateien enthält oder wenn Sie den Java-Befehl zum Festlegen von JVM-Optionen anpassen möchten, benötigen Sie eine `Procfile`-Datei.

Wir empfehlen, immer eine `Procfile` im Quellpaket mit Ihrer Anwendung bereitzustellen. Auf diese Weise können Sie genau steuern, welche Prozesse Elastic Beanstalk für Ihre Anwendung ausführt und welche Argumente diese Prozesse erhalten.

Um Java-Klassen zu kompilieren und andere Build-Befehle für die EC2 Instanzen in Ihrer Umgebung während der Bereitstellung auszuführen, fügen Sie Ihrem Anwendungsquellpaket eine Buildfile hinzu. A `Buildfile` ermöglicht es Ihnen, Ihren Quellcode unverändert bereitzustellen und auf dem Server zu bauen, anstatt ihn lokal zu kompilieren. JARs Die Java SE-Plattform bietet gängige Build-Tools für Server-Builds.

Details zu den verschiedenen Möglichkeiten für das Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter the section called “Erweitern von Linux-Plattformen”.

Konfigurieren der Java SE-Umgebung

Mit den Java SE-Plattformeinstellungen können Sie das Verhalten Ihrer EC2 Amazon-Instances optimieren. Sie können die EC2 Amazon-Instance-Konfiguration der Elastic Beanstalk Beanstalk-Umgebung mithilfe der Elastic Beanstalk Beanstalk-Konsole bearbeiten.

Mithilfe der Elastic Beanstalk-Konsole aktivieren Sie die Protokollrotation an Amazon S3 und konfigurieren Variablen, die die Anwendung aus der Umgebung auslesen kann.

So konfigurieren Sie Ihre Java SE-Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Rotation der Protokolldateien nach Amazon S3 aktivieren — Gibt an, ob Protokolldateien für die EC2 Amazon-Instances Ihrer Anwendung in den Amazon S3-Bucket kopiert werden, der Ihrer Anwendung zugeordnet ist.

Statische Dateien

Um die Leistung zu verbessern, können Sie den Abschnitt Static files (Statische Dateien) benutzen, um den Proxy-Server so zu konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zum Verzeichnis-Mapping fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, anstatt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter the section called “Statische Dateien”.

Umgebungseigenschaften

Im Abschnitt Umgebungseigenschaften können Sie die Umgebungskonfigurationseinstellungen für die EC2 Amazon-Instances angeben, auf denen Ihre Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet.

In der Java SE-Umgebung mit Elastic Beanstalk-Ausführung kann über auf die Umgebungsvariablen zugegriffen werde `System.getenv()`. Beispielsweise können Sie mit folgendem Code eine Eigenschaft mit dem Namen `API_ENDPOINT` als Variable lesen:

```
String endpoint = System.getenv("API_ENDPOINT");
```

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Java SE-Konfigurations-Namespace

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Die Java SE-Plattform definiert keine plattformspezifischen Namespaces. Sie können den Proxy so konfigurieren, dass statische Dateien mittels des Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` bereitgestellt werden. Einzelheiten und ein Beispiel finden Sie unter the section called “Statische Dateien”.

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Die Java SE-Plattform im Amazon Linux-AMI (vor Amazon Linux 2)

Falls Ihre Elastic Beanstalk Java SE-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/ AL2 Plattformbranches sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
> - Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden

Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Namespaces für die Java SE-Konfiguration — Amazon Linux AMI () AL1

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben durchführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Die Java SE-Plattform unterstützt zusätzlich zu den von allen Plattformen unterstützten Namespaces eine weitere Konfigurationsoption für einen plattformspezifischen Namespace. Mit dem Namespace `aws:elasticbeanstalk:container:java:staticfiles` können Sie Optionen definieren, um für die Webanwendung Pfade zu Ordnern im Quell-Bundle der Anwendung zuzuweisen, die statische Inhalte enthalten.

Zum Beispiel werden mit diesem option_settings-Snippet zwei Optionen im Namespace der statischen Dateien definiert. Die erste ordnet den Pfad `/public` dem Ordner `public` zu, die zweite ordnet den Pfad `/images` dem Ordner `img` zu:

```
option_settings:
  aws:elasticbeanstalk:container:java:staticfiles:
    /html: statichtml
    /images: staticimages
```

Die über diesen Namespace zugewiesenen Ordner müssen sich tatsächlich im Stamm des Quell-Bundles befinden. Sie können keinen Pfad zu einem Ordner in einer JAR-Datei zuweisen.

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Auf dem JARs Server mit einer Builddatei bauen

Sie können die Klassendateien und JAR-Dateien Ihrer Anwendung auf den EC2 Instanzen in Ihrer Umgebung erstellen, indem Sie einen Build-Befehl aus einer `Buildfile` Datei in Ihrem Quellpaket aufrufen.

Befehle in einer `Buildfile`-Datei werden nur einmal ausgeführt und müssen anschließend beendet werden, wohingegen die Befehle in einer Procfile normalerweise für die Lebensdauer der Anwendung ausgeführt und im Falle einer Beendigung neu gestartet werden. Um den JARs in Ihrer Anwendung auszuführen, verwenden Sie a. `Procfile`

Einzelheiten zur Platzierung und Syntax von a `Buildfile` finden Sie unterBuildfile und Procfile.

In obigem `Buildfile`-Beispiel wird Apache Maven ausgeführt, um eine Webanwendung aus Quellcode zu erstellen. Eine Beispielanwendung mit dieser Funktion mit dieser Funktion finden Sie unter Java-Webanwendungsbeispiele.

Example Buildfile

```
build: mvn assembly:assembly -DdescriptorId=jar-with-dependencies
```

Die Java SE-Plattform enthält die folgenden Build-Tools, die Sie über das Build-Skript aufrufen können:

- `javac` – Java-Compiler
- `ant` – Apache Ant
- `mvn` – Apache Maven
- `gradle` – Gradle

Konfigurieren der Anwendung mit einer Procfile-Datei

Sofern im Stamm des Quell-Bundles der Anwendung mehr als eine JAR-Datei vorhanden ist, müssen Sie eine `Procfile`-Datei einbinden, anhand derer Elastic Beanstalk erkennt, welche JAR-Dateien(en) ausgeführt werden sollen. Bei einer einzelnen JAR-Anwendung können Sie auch eine `Procfile`-Datei einbinden. Damit wird die JVM (Java Virtual Machine) konfiguriert, auf der die Anwendung ausgeführt wird.

Wir empfehlen, immer eine `Procfile` im Quellpaket mit Ihrer Anwendung bereitzustellen. Auf diese Weise können Sie genau steuern, welche Prozesse Elastic Beanstalk für Ihre Anwendung ausführt und welche Argumente diese Prozesse erhalten.

Einzelheiten zum Schreiben und Verwenden eines `Procfile` finden Sie unterBuildfile und Procfile.

Example Procfile

```
web: java -Xms256m -jar server.jar
cache: java -jar mycache.jar
web_foo: java -jar other.jar
```

Der Befehl für die Ausführung der wichtigsten JAR-Datei in der Anwendung muss den Namen `web` tragen und der erste Befehl in der `Procfile`-Datei sein. Der nginx-Server leitet alle HTTP-Anforderungen, die er vom Load Balancer der Umgebung erhält, an diese Anwendung weiter.

Elastic Beanstalk geht davon aus, dass alle Einträge in der Procfile-Datei stets ausgeführt werden sollen. Daher wird automatisch jede Anwendung, die in der Procfile-Datei definiert ist und beendet wird, neu gestartet. Wenn die Anwendung mit einem Befehl beendet, aber nicht wieder neu gestartet werden soll, verwenden Sie eine -Datei `Buildfile`.

Procfile auf Amazon Linux-AMI (vor Amazon Linux 2) verwenden

Falls Ihre Elastic Beanstalk Java SE-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/ AL2 Plattformbranches sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
> - Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Port-Weitergabe — Amazon Linux AMI (AL1)

Standardmäßig wird der nginx-Proxy von Elastic Beanstalk so konfiguriert, dass alle Anforderungen an die Anwendung über Port 5000 weitergeleitet werden. Diese Port-Standardeinstellung können Sie überschreiben. Legen Sie dazu die `PORT`-Umgebungseigenschaft auf den Überwachungsport der Hauptanwendung fest.

Wenn Sie einen `Procfile` verwenden, um mehrere Anwendungen auszuführen, erwartet Elastic Beanstalk auf den AMI-Plattformversionen von Amazon Linux, dass jede zusätzliche Anwendung auf einem Port lauscht, der 100 höher ist als der vorherige. Elastic Beanstalk legt die PORT-Variable, auf die von jeder Anwendung zugegriffen werden kann, auf den Port fest, auf den die Anwendung erwartungsgemäß zugreift. Sie können auf diese Variable im Anwendungscode zugreifen, indem Sie aufrufe `System.getenv("PORT")`.

Im vorherigen Beispiel für `Procfile` wurden für die `web`-Anwendung der Überwachungsport 5000, für `cache` der Überwachungsport 5100 und für `web_foo` der Überwachungsport 5200 konfiguriert. Für `web` wird der Überwachungsport durch Auslesen der `PORT`-Variable und Addieren der Zahl 100 konfiguriert. Damit wird der Überwachungsport von `cache` für die Anforderungsübermittlung bestimmt.

Konfigurieren des Proxy-Server

Elastic Beanstalk verwendet nginx als Reverse-Proxy, um Ihre Anwendung Ihrem Elastic-Load-Balancing-Load-Balancer auf Port 80 zuzuordnen. Elastic Beanstalk bietet eine nginx-Standardkonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Standardmäßig wird der nginx-Proxy von Elastic Beanstalk so konfiguriert, dass alle Anforderungen an die Anwendung über Port 5000 weitergeleitet werden. Diese Port-Standardeinstellung können Sie überschreiben. Legen Sie dazu die `PORT`-Umgebungseigenschaft auf den Überwachungsport der Hauptanwendung fest.

> **Note**
>
> Der Überwachungsport der Anwendung kollidiert nicht mit dem Überwachungsport des nginx-Servers, von dem dieser die Anforderungen des Load Balancers empfängt.

Konfigurieren des Proxy-Servers auf Ihrer Plattformversion

Alle AL2 023/ AL2 -Plattformen unterstützen eine einheitliche Proxykonfigurationsfunktion. Weitere Informationen zur Konfiguration des Proxyservers auf Ihren Plattformversionen, auf denen AL2 AL2 023/ ausgeführt wird, finden Sie unter. Reverse-Proxy-Konfiguration

Konfigurieren des Proxys auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)

Falls Ihre Elastic Beanstalk Java SE-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/ AL2 Plattformbranches sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
> - Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Erweiterung und Außerkraftsetzung der Standard-Proxykonfiguration — Amazon Linux AMI () AL1

Um die nginx-Standardkonfiguration von Elastic Beanstalk zu überschreiben, fügen Sie `.conf`-Konfigurationsdateien zum Ordner `.ebextensions/nginx/conf.d/` im Quell-Bundle der Anwendung hinzu. Die nginx-Konfiguration von Elastic Beanstalk enthält standardmäßig `.conf`-Dateien in diesem Ordner.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- conf.d
|           `-- myconf.conf
`-- web.jar
```

Um die nginx-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, binden Sie bei eine Konfiguration in Ihr Quell-Bundle ei `.ebextensions/nginx/nginx.conf`:

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- nginx.conf
```

```
`-- web.jar
```

Wenn Sie die nginx-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeile zu `nginx.conf` hinzu, damit die Konfigurationen von Elastic Beanstalk für Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk, automatische Anwendungszuordnungen und statische Dateien einbezogen werden.

```
include conf.d/elasticbeanstalk/*.conf;
```

Die folgende Beispielkonfiguration aus der Beispielanwendung Scorekeep überschreibt die Standardkonfiguration von Elastic Beanstalk, um eine statische Webanwendung aus dem `public`-Unterverzeichnis von `/var/app/current` zu bedienen, wohin die Java SE-Plattform den Anwendungsquellcode kopiert. Der `/api`-Standort leitet den Datenverkehr zu Routen unter `/api/` an die Spring-Anwendung weiter, die den Port 5000 liest. Jeder andere Datenverkehr wird von der Web-Anwendung auf dem Root-Pfad bedient.

Example

```
user                    nginx;
error_log               /var/log/nginx/error.log warn;
pid                     /var/run/nginx.pid;
worker_processes        auto;
worker_rlimit_nofile    33282;

events {
    worker_connections  1024;
}

http {
  include       /etc/nginx/mime.types;
  default_type  application/octet-stream;

  log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

  include       conf.d/*.conf;

  map $http_upgrade $connection_upgrade {
      default     "upgrade";
  }
```

```
  server {
      listen        80 default_server;
      root /var/app/current/public;

      location / {
      }git pull


      location /api {
          proxy_pass          http://127.0.0.1:5000;
          proxy_http_version  1.1;

          proxy_set_header    Connection          $connection_upgrade;
          proxy_set_header    Upgrade             $http_upgrade;
          proxy_set_header    Host                $host;
          proxy_set_header    X-Real-IP           $remote_addr;
          proxy_set_header    X-Forwarded-For     $proxy_add_x_forwarded_for;
      }

      access_log    /var/log/nginx/access.log main;

      client_header_timeout 60;
      client_body_timeout   60;
      keepalive_timeout     60;
      gzip                  off;
      gzip_comp_level       4;

      # Include the Elastic Beanstalk generated locations
      include conf.d/elasticbeanstalk/01_static.conf;
      include conf.d/elasticbeanstalk/healthd.conf;
  }
}
```

Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Java Elastic Beanstalk Beanstalk-Umgebung

Dieses Thema enthält Anweisungen zum Erstellen eines Amazon RDS mit der Elastic-Beanstalk-Konsole. Sie können eine Amazon Relational Database Service (Amazon RDS)-DB-Instance verwenden, um Daten zu speichern, die Ihre Anwendung sammelt und modifiziert. Die Datenbank

kann Ihrer Umgebung angefügt und von Elastic Beanstalk verwaltet werden oder sie kann extern erstellt und verwaltet werden.

Bei der ersten Verwendung von Amazon RDS fügen Sie in der Testumgebung mithilfe der Elastic Beanstalk-Konsole eine DB-Instance hinzu und überprüfen, ob die Anwendung eine Verbindung zu dieser herstellen kann.

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftenwert
`RDS_HOSTNAME`	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
`RDS_PORT`	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.

Eigenschaftenname	Beschreibung	Eigenschaftenwert
RDS_DB_NAME	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Weitere Informationen zum Konfigurieren einer internen DB-Instance finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung. Anweisungen zum Konfigurieren einer externen Datenbank für die Verwendung mit Elastic Beanstalk finden Sie unter Verwenden von Elastic Beanstalk mit Amazon RDS.

Für die Verbindungserstellung zu einer Datenbank fügen Sie die geeignete JAR-Treiberdatei zur Anwendung hinzu, laden die Klasse in den Code und erstellen ein Verbindungsobjekt mit den von Elastic Beanstalk bereitgestellten Umgebungseigenschaften.

Abschnitte

- Herunterladen eines JDBC-Treibers
- Verbinden mit einer Datenbank (Java SE-Plattformen)
- Verbinden mit einer Datenbank (Tomcat-Plattformen)
- Fehlerbehebung bei Datenbankverbindungen

Herunterladen eines JDBC-Treibers

Sie benötigen die JAR-Datei des JDBC-Treibers für die DB-Engine, die Sie gewählt haben. Speichern Sie die JAR-Datei in Ihrem Quellcode und fügen Sie sie in Ihren Klassenpfad ein, wenn Sie die Klasse kompilieren, die Verbindungen mit der Datenbank herstellt.

Die neuesten Treiber für Ihre DB-Engine finden Sie an den folgenden Speicherorten:

- MySQL – MySQL Connector/J
- Oracle SE-1 – Oracle JDBC Driver
- Postgres – PostgreSQL JDBC Driver
- SQL Server – Microsoft JDBC Driver

Um den JDBC-Treiber zu verwenden, rufen Sie `Class.forName()` auf, um ihn zu laden, bevor Sie die Verbindung mit `DriverManager.getConnection()` in Ihrem Code herstellen.

JDBC verwendet eine Verbindungszeichenfolge mit folgendem Format:

```
jdbc:driver://hostname:port/dbName?user=userName&password=password
```

Sie können den Hostnamen, Port, Datenbanknamen, Benutzernamen und das Passwort von den Umgebungsvariablen abrufen, die Elastic Beanstalk für Ihre Anwendung bereitstellt. Der Treibername ist spezifisch für Ihren Datenbanktyp und Ihre Treiberversion. Es folgen Beispiele für Treibernamen:

- `mysql` for MySQL
- `postgresql` for PostgreSQL
- `oracle:thin` for Oracle Thin
- `oracle:oci` for Oracle OCI
- `oracle:oci8` for Oracle OCI 8
- `oracle:kprb` for Oracle KPRB
- `sqlserver` for SQL Server

Verbinden mit einer Datenbank (Java SE-Plattformen)

In einer Java SE-Umgebung verwenden Sie `System.getenv()`, um die Verbindungsvariablen aus der Umgebung zu lesen. Der folgende Beispielcode zeigt eine Klasse, die eine Verbindung zu einer PostgreSQL-Datenbank erstellt.

```
private static Connection getRemoteConnection() {
    if (System.getenv("RDS_HOSTNAME") != null) {
      try {
      Class.forName("org.postgresql.Driver");
```

```
      String dbName = System.getenv("RDS_DB_NAME");
      String userName = System.getenv("RDS_USERNAME");
      String password = System.getenv("RDS_PASSWORD");
      String hostname = System.getenv("RDS_HOSTNAME");
      String port = System.getenv("RDS_PORT");
      String jdbcUrl = "jdbc:postgresql://" + hostname + ":" + port + "/" + dbName + "?
user=" + userName + "&password=" + password;
      logger.trace("Getting remote connection with connection string from environment
 variables.");
      Connection con = DriverManager.getConnection(jdbcUrl);
      logger.info("Remote connection successful.");
      return con;
    }
    catch (ClassNotFoundException e) { logger.warn(e.toString());}
    catch (SQLException e) { logger.warn(e.toString());}
    }
    return null;
  }
```

Verbinden mit einer Datenbank (Tomcat-Plattformen)

In einer Tomcat-Umgebung werden Umgebungseigenschaften als Systemeigenschaften bereitgestellt, die mit zugreifbar sin `System.getProperty()`.

Der folgende Beispielcode zeigt eine Klasse, die eine Verbindung zu einer PostgreSQL-Datenbank erstellt.

```
private static Connection getRemoteConnection() {
    if (System.getProperty("RDS_HOSTNAME") != null) {
      try {
      Class.forName("org.postgresql.Driver");
      String dbName = System.getProperty("RDS_DB_NAME");
      String userName = System.getProperty("RDS_USERNAME");
      String password = System.getProperty("RDS_PASSWORD");
      String hostname = System.getProperty("RDS_HOSTNAME");
      String port = System.getProperty("RDS_PORT");
      String jdbcUrl = "jdbc:postgresql://" + hostname + ":" + port + "/" + dbName + "?
user=" + userName + "&password=" + password;
      logger.trace("Getting remote connection with connection string from environment
 variables.");
      Connection con = DriverManager.getConnection(jdbcUrl);
      logger.info("Remote connection successful.");
      return con;
```

```
    }
    catch (ClassNotFoundException e) { logger.warn(e.toString());}
    catch (SQLException e) { logger.warn(e.toString());}
    }
    return null;
  }
```

Wenn Sie Probleme haben, eine Verbindung herzustellen oder SQL-Anweisungen auszuführen, versuchen Sie, den folgenden Code in einer JSP-Datei zu platzieren. Dieser Code stellt eine Verbindung zu einer DB-Instance her, erstellt eine Tabelle und schreibt in sie.

```
<%@ page import="java.sql.*" %>
<%
  // Read RDS connection information from the environment
  String dbName = System.getProperty("RDS_DB_NAME");
  String userName = System.getProperty("RDS_USERNAME");
  String password = System.getProperty("RDS_PASSWORD");
  String hostname = System.getProperty("RDS_HOSTNAME");
  String port = System.getProperty("RDS_PORT");
  String jdbcUrl = "jdbc:mysql://" + hostname + ":" +
    port + "/" + dbName + "?user=" + userName + "&password=" + password;

  // Load the JDBC driver
  try {
    System.out.println("Loading driver...");
    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("Driver loaded!");
  } catch (ClassNotFoundException e) {
    throw new RuntimeException("Cannot find the driver in the classpath!", e);
  }

  Connection conn = null;
  Statement setupStatement = null;
  Statement readStatement = null;
  ResultSet resultSet = null;
  String results = "";
  int numresults = 0;
  String statement = null;

  try {
    // Create connection to RDS DB instance
    conn = DriverManager.getConnection(jdbcUrl);
```

```
    // Create a table and write two rows
    setupStatement = conn.createStatement();
    String createTable = "CREATE TABLE Beanstalk (Resource char(50));";
    String insertRow1 = "INSERT INTO Beanstalk (Resource) VALUES ('EC2 Instance');";
    String insertRow2 = "INSERT INTO Beanstalk (Resource) VALUES ('RDS Instance');";

    setupStatement.addBatch(createTable);
    setupStatement.addBatch(insertRow1);
    setupStatement.addBatch(insertRow2);
    setupStatement.executeBatch();
    setupStatement.close();

  } catch (SQLException ex) {
    // Handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
  } finally {
    System.out.println("Closing the connection.");
    if (conn != null) try { conn.close(); } catch (SQLException ignore) {}
  }

  try {
    conn = DriverManager.getConnection(jdbcUrl);

    readStatement = conn.createStatement();
    resultSet = readStatement.executeQuery("SELECT Resource FROM Beanstalk;");

    resultSet.first();
    results = resultSet.getString("Resource");
    resultSet.next();
    results += ", " + resultSet.getString("Resource");

    resultSet.close();
    readStatement.close();
    conn.close();

  } catch (SQLException ex) {
    // Handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
  } finally {
       System.out.println("Closing the connection.");
```

```
      if (conn != null) try { conn.close(); } catch (SQLException ignore) {}
  }
%>
```

Zum Anzeigen der Ergebnisse platzieren Sie den folgenden Code im Text des HTML-Teils der JSP-Datei.

```
<p>Established connection to RDS. Read first two rows: <%= results %></p>
```

Fehlerbehebung bei Datenbankverbindungen

> **Testen Sie Amazon Q Developer CLI für KI-gestützte Problembehebung**
>
> Amazon Q Developer CLI kann Ihnen helfen, Umgebungsprobleme schnell zu beheben. Die Q CLI bietet Lösungen, indem sie den Umgebungsstatus überprüft, Ereignisse überprüft, Protokolle analysiert und klärende Fragen stellt. Weitere Informationen und detaillierte Anleitungen finden Sie in den Blogs unter Troubleshooting Elastic Beanstalk Environments with Amazon Q Developer CLI. AWS

Wenn Sie auf Probleme mit der Verbindung zu einer Datenbank über Ihre Anwendung stoßen, überprüfen Sie das Web-Container-Protokoll und die Datenbank.

Überprüfen von Protokollen

Sie können alle Protokolle aus Ihrer Elastic Beanstalk-Umgebung in Eclipse anzeigen. Wenn Sie die AWS Explorer-Ansicht nicht geöffnet haben, klicken Sie auf den Pfeil neben dem orangefarbenen AWS Symbol in der Werkzeugleiste und wählen Sie dann Explorer-Ansicht anzeigen. AWS Erweitern Sie AWS Elastic Beanstalk und Ihren Umgebungsnamen und öffnen Sie dann das Kontextmenü (rechte Maustaste) für den Server. Klicken Sie auf Open in WTP Server Editor.

Wählen Sie die Registerkarte Log der Server-Ansicht, um die aggregierten Protokolle aus Ihrer Umgebung zu sehen. Um die neuesten Protokolle zu öffnen, wählen Sie die Schaltfläche Refresh oben rechts auf der Seite.

Scrollen Sie nach unten, um die Tomcat-Protokolle in zu suche `/var/log/tomcat7/catalina.out`. Wenn Sie die Webseite aus unserem oben stehenden Beispiel mehrmals geladen haben, sehen Sie möglicherweise Folgendes:

```
-------------------------------------
/var/log/tomcat7/catalina.out
-------------------------------------
INFO: Server startup in 9285 ms
Loading driver...
Driver loaded!
SQLException: Table 'Beanstalk' already exists
SQLState: 42S01
VendorError: 1050
Closing the connection.
Closing the connection.
```

Alle Informationen, die die Webanwendung an die Standardausgabe sendet, wird im Web-Container-Protokoll angezeigt. Im vorherigen Beispiel versucht die Anwendung, die Tabelle jedes Mal zu erstellen, wenn die Seite geladen wird. Dies führt zur Erfassung einer SQL-Ausnahme auf jeder Seite, die nach der ersten geladen wird.

Beispielsweise ist die vorhergehende akzeptabel. Aber in tatsächlichen Anwendungen behalten Sie Ihre Datenbankdefinitionen in Schemaobjekten, führen Transaktionen in Modellklassen durch und koordinieren Anforderungen mit Controller-Servlets.

Herstellen einer Verbindung mit einer RDS DB-Instance

Sie können eine direkte Verbindung zur RDS DB-Instance in Ihrer Elastic Beanstalk-Umgebung herstellen, indem Sie die MySQL-Client-Anwendung verwenden.

Öffnen Sie zunächst die Sicherheitsgruppe für Ihre RDS DB-Instance, um Datenverkehr von Ihrem Computer zuzulassen.

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie neben Endpoint (Endpunkt) den Link zur Amazon RDS-Konsole aus.
6. Wählen Sie auf der Detailseite der RDS Dashboard-Instance unter Security and Network die Sicherheitsgruppe beginnend mit rds- neben Security Groups.

> **Note**
>
> Die Datenbank kann über mehrere Einträge mit der Bezeichnung Security Groups verfügen. Wenn Sie ein älteres Konto haben, das nicht standardmäßig über Amazon Virtual Private Cloud (Amazon VPC) verfügt, verwenden Sie die erste, die mit awseb beginnt.

7. Rufen Sie in Security group details die Registerkarte Inbound auf und klicken Sie auf Edit.
8. Fügen Sie eine Regel für MySQL (Port 3306) hinzu, die Datenverkehr von Ihrer IP-Adresse erlaubt, angegeben im CIDR-Format.
9. Wählen Sie Save aus. Die Änderungen werden sofort wirksam.

Kehren Sie zu den Elastic Beanstalk-Konfigurationsdetails für Ihre Umgebung zurück und beachten Sie den Endpunkt. Sie verwenden den Domänennamen für die Verbindung mit der RDS-DB-Instance.

Installieren Sie den MySQL-Client und initiieren Sie eine Verbindung mit der Datenbank auf Port 3306. Installieren Sie unter Windows MySQL Workbench von der MySQL-Startseite und befolgen Sie die Anweisungen.

Installieren Sie unter Linux den MySQL-Client mithilfe des Paket-Managers für Ihre Verteilung. Das folgende Beispiel funktioniert auf Ubuntu und anderen Debian-Derivaten.

```
// Install MySQL client
$ sudo apt-get install mysql-client-5.5
...
// Connect to database
$ mysql -h aas839jo2vwhwb.cnubrrfwfka8.us-west-2.rds.amazonaws.com -u username -
ppassword ebdb
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 117
Server version: 5.5.40-log Source distribution
...
```

Nachdem Sie eine Verbindung hergestellt haben, können Sie SQL-Befehle ausfehlen, um den Status der Datenbank, unabhängig davon, ob Ihre Tabellen und Zeilen erstellt wurden, und andere Informationen anzuzeigen.

```
mysql> SELECT Resource from Beanstalk;
+--------------+
| Resource     |
+--------------+
| EC2 Instance |
| RDS Instance |
+--------------+
2 rows in set (0.01 sec)
```

Java-Tools und -Ressourcen

Es gibt zahlreiche Möglichkeiten, um zusätzliche Unterstützung bei der Entwicklung von Java-Anwendungen zu erhalten.

Ressource	Description
Das AWS Java-Entwicklungsforum	Hier können Sie Fragen stellen und Feedback erhalten.
Java Developer Center	Umfangreiches Kit mit Beispiel-Code, Dokumentation, Tools und zusätzlichen Ressourcen.

Bereitstellung von Node.js Anwendungen mit Elastic Beanstalk

Dieses Kapitel enthält Anweisungen zur Konfiguration und Bereitstellung Ihrer Node.js -Webanwendung für AWS Elastic Beanstalk. Es enthält auch exemplarische Vorgehensweisen für allgemeine Aufgaben wie Datenbankintegration und Arbeit mit dem Express-Framework. Elastic Beanstalk macht es einfach, Ihre Node.js Webanwendungen mithilfe von Amazon Web Services bereitzustellen, zu verwalten und zu skalieren.

Sie können Ihre Anwendung in nur wenigen Minuten mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) oder der Elastic Beanstalk Beanstalk-Konsole bereitstellen. Nachdem Sie Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt haben, können Sie weiterhin die EB-CLI verwenden, um Ihre Anwendung und Umgebung zu verwalten, oder Sie können die Elastic Beanstalk Beanstalk-Konsole verwenden, AWS CLI oder die. APIs

Folgen Sie den step-by-step Anweisungen QuickStart für Node.js zum Erstellen und Bereitstellen einer Hello World Node.js Webanwendung mit der EB CLI.

Themen

- QuickStart: Stellen Sie eine Node.js -Anwendung auf Elastic Beanstalk bereit
- Einrichtung Ihrer Node.js Entwicklungsumgebung für Elastic Beanstalk
- Verwenden der Elastic Beanstalk Node.js-Plattform
- Weitere Elastic Beanstalk Beanstalk-Beispielanwendungen und Tutorials für Node.js
- Eine Node.js Express-Anwendung auf Elastic Beanstalk bereitstellen
- Bereitstellung einer Node.js Express-Anwendung mit Clustering auf Elastic Beanstalk
- Bereitstellen einer Node.js-Anwendung mit DynamoDB in Elastic Beanstalk
- Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Node.js Elastic Beanstalk Beanstalk-Umgebung
- Tools und Ressourcen für Node.js

QuickStart: Stellen Sie eine Node.js -Anwendung auf Elastic Beanstalk bereit

Dieses QuickStart Tutorial führt Sie durch den Prozess der Erstellung einer Node.js -Anwendung und deren Bereitstellung in einer AWS Elastic Beanstalk Umgebung.

> ⚠ Nicht für Produktionszwecke
>
> Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine Node.js -Anwendung
- Schritt 2: Führen Sie Ihre Anwendung lokal aus
- Schritt 3: Stellen Sie Ihre Anwendung Node.js mit der EB CLI bereit
- Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus
- Schritt 5: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center - Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

EB-CLI

Dieses Tutorial verwendet die Befehlszeilenschnittstelle Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Node.js

Installieren Sie Node.js auf Ihrem lokalen Computer, indem Sie den Anweisungen zur Installation von Node.js auf der Website Node.js folgen.

Überprüfen Sie Ihre Node.js -Installation, indem Sie den folgenden Befehl ausführen.

```
~$ node -v
```

Schritt 1: Erstellen Sie eine Node.js -Anwendung

Erstellen Sie ein Projektverzeichnis.

```
~$ mkdir eb-nodejs
~$ cd eb-nodejs
```

Erstellen Sie anschließend eine Anwendung, die Sie mit Elastic Beanstalk bereitstellen. Wir werden einen „Hello World“ RESTful -Webservice erstellen.

Example~/eb-nodejs/server.js

```
const http = require('node:http');

const hostname = '127.0.0.1';
const port = 8080;
```

```
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello Elastic Beanstalk!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Diese Anwendung öffnet einen Listener auf Port 8080. Elastic Beanstalk leitet Anfragen für Node.js standardmäßig an Port 8080 an Ihre Anwendung weiter.

Schritt 2: Führen Sie Ihre Anwendung lokal aus

Führen Sie den folgenden Befehl aus, um Ihre Anwendung lokal auszuführen.

```
~/eb-nodejs$ node server.js
```

Sie sollten den folgenden Text sehen.

```
Server running at http://127.0.0.1:8080/
```

Geben Sie die URL-Adresse `http://127.0.0.1:8080/` in Ihren Webbrowser ein. Der Browser sollte „Hello Elastic Beanstalk!“ anzeigen.

Schritt 3: Stellen Sie Ihre Anwendung Node.js mit der EB CLI bereit

Führen Sie die folgenden Befehle aus, um eine Elastic Beanstalk Beanstalk-Umgebung für diese Anwendung zu erstellen.

Um eine Umgebung zu erstellen und Ihre Anwendung Node.js bereitzustellen

1. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

   ```
   ~/eb-nodejs$ eb init -p node.js nodejs-tutorial --region us-east-2
   ```

   Dieser Befehl erstellt eine Anwendung mit dem Namen `nodejs-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der neuesten Plattformversion von Node.js erstellt werden.

2. (Optional) Führen Sie den Vorgang eb init erneut aus, um ein Standardschlüsselpaar zu konfigurieren, sodass Sie SSH verwenden können, um eine Verbindung zu der EC2 Instance herzustellen, auf der Ihre Anwendung ausgeführt wird.

```
~/eb-nodejs$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

   Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein Schlüsselpaar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erstellt automatisch eine Zip-Datei für Ihre Anwendung und stellt sie auf einer EC2 Instanz in der Umgebung bereit. Nach der Bereitstellung Ihrer Anwendung startet Elastic Beanstalk sie auf Port 8080.

```
~/eb-nodejs$ eb create nodejs-env
```

   Es dauert etwa fünf Minuten, bis Elastic Beanstalk Ihre Umgebung erstellt hat.

Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, öffnen Sie Ihre Website mit. eb open

```
~/eb-nodejs$ eb open
```

Herzlichen Glückwunsch! Sie haben eine Node.js Anwendung mit Elastic Beanstalk bereitgestellt! Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet.

Schritt 5: Bereinigen

Sie können Ihre Umgebung beenden, wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind. Elastic Beanstalk beendet alle AWS Ressourcen, die mit Ihrer Umgebung verknüpft sind.

Führen Sie den folgenden Befehl aus, um Ihre Elastic Beanstalk Beanstalk-Umgebung mit der EB-CLI zu beenden.

```
~/eb-nodejs$ eb terminate
```

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt. Es dient als einfache Beispielanwendung mit einer einzigen EC2 Instanz, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 Instanz — Eine EC2 virtuelle Maschine von Amazon, die so konfiguriert ist, dass sie Web-Apps auf der von Ihnen ausgewählten Plattform ausführt.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Die Bereitstellung einer neuen Anwendungsversion ist sehr schnell, da dafür keine Bereitstellung oder Neustarts von Instances erforderlich sind. EC2 Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

> **Probieren Sie weitere Tutorials aus**
>
> Wenn Sie andere Tutorials mit anderen Beispielanwendungen ausprobieren möchten, finden Sie weitere Informationen unterWeitere Elastic Beanstalk Beanstalk-Beispielanwendungen und Tutorials für Node.js.

Wenn Sie eine oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, Node.js -Anwendungen lokal zu entwickeln und auszuführen, finden Sie weitere Informationen unterEinrichtung Ihrer Node.js Entwicklungsumgebung für Elastic Beanstalk.

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

Einrichtung Ihrer Node.js Entwicklungsumgebung für Elastic Beanstalk

Dieses Thema enthält Anweisungen zum Einrichten einer Entwicklungsumgebung vom Typ Node.js, in der Sie Ihre Anwendung lokal testen können, bevor Sie sie bereitstellen. AWS Elastic Beanstalk Außerdem wird auf Websites verwiesen, die Installationsanweisungen für nützliche Tools bereitstellen.

Themen

- Installieren von Node.js
- Bestätigen der npm-Installation

- Installieren Sie das AWS SDK für Node.js
- Installieren des Express-Generators
- Einrichten eines Express-Frameworks und eines Servers

Installieren von Node.js

Installieren Sie Node.js, um Node.js-Anwendungen lokal auszuführen. Falls Sie keine Version bevorzugen, wählen Sie die neueste, von Elastic Beanstalk unterstützte Version aus. Eine Liste der unterstützten Versionen finden Sie unter Node.js im Dokument AWS Elastic Beanstalk -Plattformen.

Laden Sie Node.js unter nodejs.org herunter.

Bestätigen der npm-Installation

Node.js verwendet einen npm-Paketmanager, mit dem Sie Tools und Frameworks für die Verwendung in Ihrer Anwendung installieren können. npm wird mit Node.js bereitgestellt und automatisch installiert, sobald Sie Node.js herunterladen und installieren. Sie können zur Bestätigung, dass npm installiert ist, folgenden Befehl ausführen:

```
$ npm -v
```

Weitere Informationen zu npm finden Sie auf der npmjs-Website.

Installieren Sie das AWS SDK für Node.js

Wenn Sie AWS Ressourcen von Ihrer Anwendung aus verwalten müssen, installieren Sie das AWS SDK für JavaScript in Node.js. So installieren Sie das SDK mit npm:

```
$ npm install aws-sdk
```

Weitere Informationen finden Sie auf der Startseite des AWS SDK JavaScript in Node.js.

Installieren des Express-Generators

Express ist ein Framework für Webanwendungen, das in Node.js ausgeführt wird. Um es zu verwenden, installieren Sie zuerst die Express-Generator-Befehlszeilenanwendung. Sobald der Express-Generator installiert ist, können Sie den express-Befehl ausführen, um eine

Basisprojektstruktur für Ihre Webanwendung zu generieren. Sobald das Basisprojekt, die Dateien und Abhängigkeiten installiert sind, können Sie einen lokalen Express-Server auf Ihrem Entwicklungscomputer starten.

> **Note**
>
> - Diese Schritte führen Sie durch die Installation des Express-Generators auf einem Linux-Betriebssystem.
> - Unter Linux müssen Sie, je nach Berechtigungsstufe für die Systemverzeichnisse, möglicherweise einige dieser Befehle mit dem Präfix `sudo` versehen.

So installieren Sie den Express-Generator in Ihrer Entwicklungsumgebung

1. Erstellen Sie ein Arbeitsverzeichnis für Ihr Express-Framework und Ihren Server.

   ```
   ~$ mkdir node-express
   ~$ cd node-express
   ```

2. Installieren Sie Express global, sodass Sie Zugriff auf den `express`-Befehl haben.

   ```
   ~/node-express$ npm install -g express-generator
   ```

3. Je nach Betriebssystem müssen Sie möglicherweise einen Pfad zur Ausführung des `express`-Befehls festlegen. Die Ausgabe des vorherigen Schritts gibt Auskunft darüber, ob Sie Ihre Pfadvariable setzen müssen. Im Folgenden finden Sie ein Beispiel für Linux.

   ```
   ~/node-express$ export PATH=$PATH:/usr/local/share/npm/bin/express
   ```

   Wenn Sie den Anleitungen in diesem Kapitel folgen, müssen Sie den express-Befehl von verschiedenen Verzeichnissen ausführen. Jedes Tutorial richtet eine Express-Basisprojektstruktur in einem eigenen Verzeichnis ein.

Sie haben jetzt den Express-Befehlszeilengenerator installiert. Sie können damit ein Framework-Verzeichnis für Ihre Webanwendung erstellen, Abhängigkeiten einrichten und den Web-App-Server starten. Als Nächstes führen wir die Schritte durch, um dies in dem von uns erstellten `node-express`-Verzeichnis zu erreichen.

Einrichten eines Express-Frameworks und eines Servers

Gehen Sie wie folgt vor, um die Basisverzeichnisse und -inhalte des Express-Frameworks zu erstellen. Die Tutorials in diesem Kapitel enthalten auch diese Schritte zum Einrichten des Express-Basis-Frameworks in den einzelnen Anwendungsverzeichnissen des Tutorials.

So richten Sie ein Express-Framework und einen Server ein

1. Führen Sie den Befehl `express` aus. Dadurch werden `package.json`, `app.js` und weitere Verzeichnisse generiert.

   ```
   ~/node-express$ express
   ```

   Wenn Sie gefragt werden, ob Sie fortfahren möchten, geben Sie **y** ein.

2. Richten Sie lokale Abhängigkeiten ein.

   ```
   ~/node-express$ npm install
   ```

3. Stellen Sie sicher, dass der Web-App-Server gestartet wird.

   ```
   ~/node-express$ npm start
   ```

   Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

   ```
   > nodejs@0.0.0 start /home/local/user/node-express
   > node ./bin/www
   ```

   Der Server läuft standardmäßig über Port 3000. Führen Sie als Test `curl http://localhost:3000` in einem anderen Terminal aus oder öffnen Sie einen Browser auf dem lokalen Computer und geben Sie die URL `http://localhost:3000` ein.

   Drücken Sie Strg+C, um den Server zu stoppen.

Verwenden der Elastic Beanstalk Node.js-Plattform

In diesem Thema wird beschrieben, wie Sie Ihre Node.js -Anwendungen auf Elastic Beanstalk konfigurieren, erstellen und ausführen.

AWS Elastic Beanstalk unterstützt eine Reihe von Plattformzweigen für verschiedene Versionen der Programmiersprache Node.js. Eine vollständige Liste finden Sie unter Node.js im Dokument AWS Elastic Beanstalk Platforms.

Elastic Beanstalk bietet Konfigurationsoptionen, mit denen sich die auf den EC2-Instances in der Elastic-Beanstalk-Umgebung ausgeführte Software anpassen lässt. Sie können die von Ihrer Anwendung benötigten Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren und die Ordner in der Anwendungsquelle, die statische Dateien enthalten, den vom Proxy-Server verwendeten Pfaden zuordnen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das Ändern der Konfiguration einer ausgeführten Umgebung verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie gespeicherte Konfigurationen verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie Konfigurationsdateien einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Sie können eine `Package.json`-Datei in Ihr Quell-Bundle aufnehmen, um Pakete während der Bereitstellung zu installieren, einen Startbefehl bereitzustellen und die Node.js-Version anzugeben, die Ihre Anwendung verwenden soll. Sie können eine -Datei `npm-shrinkwrap.json` aufnehmen, um Abhängigkeitsversionen zu sperren.

Die Node.js-Plattform umfasst einen Proxy-Server, um statische Komponenten zu verarbeiten, Datenverkehr an Ihre Anwendung weiterzuleiten und Antworten zu komprimieren. Sie können die für erweiterte Szenarien Proxy-Standardkonfiguration erweitern oder überschreiben.

Es gibt verschiedene Möglichkeiten, um Ihre Anwendung zu starten. Sie können Procfile zu Ihrem Quell-Bundle hinzufügen, um den Befehl anzugeben, mit dem Ihre Anwendung gestartet wird. Wenn Sie kein `Procfile` bereitstellen, aber stattdessen eine `package.json`-Datei verfügbar machen führt Elastic Beanstalk `npm start` aus. Wenn Sie auch dies nicht angeben, sucht Elastic Beanstalk die Datei von `app.js` oder `server.js` (in dieser Reihenfolge) und führt das Script aus.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, falls vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe Konfigurationsoptionen.

Details zu den verschiedenen Möglichkeiten für das Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter the section called “Erweitern von Linux-Plattformen”.

Konfiguration der Node.js-Umgebung

Mit den Einstellungen der Node.js-Plattform können Sie das Verhalten Ihrer Amazon-EC2-Instances optimieren. Sie können die Amazon-EC2-Instance-Konfiguration der Elastic-Beanstalk-Umgebung mit der Elastic-Beanstalk-Konsole bearbeiten.

Mithilfe der Elastic Beanstalk-Konsole aktivieren Sie die Protokollrotation an Amazon S3 und konfigurieren Variablen, die die Anwendung aus der Umgebung auslesen kann.

So konfigurieren Sie Ihre Node.js-Umgebung in der Elastic-Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

Container-Optionen

Sie können die folgenden plattformspezifischen Optionen angeben:

- Proxy server (Proxy-Server) – Der Proxy-Server, der in Ihren Umgebungs-Instances verwendet werden soll. Standardmäßig wird NGINX verwendet.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den mit der Anwendung verknüpften Amazon S3-Bucket berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon-EC2-Instances der Anwendung in den der Anwendung zugeordneten Amazon-S3-Bucket kopiert werden.

Statische Dateien

Um die Leistung zu verbessern, können Sie den Abschnitt Static files (Statische Dateien) benutzen, um den Proxy-Server so zu konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zum Verzeichnis-Mapping fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, anstatt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter the section called “Statische Dateien”.

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon-EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Diese Einstellungen werden als Schlüsselwert-Paare an die Anwendung weitergeleitet.

In der Node.js Umgebung, in der ausgeführt wird AWS Elastic Beanstalk, können Sie auf die Umgebungsvariablen zugreifen, indem Sie den Befehl ausführen`process.env.ENV_VARIABLE`.

```
var endpoint = process.env.API_ENDPOINT
```

Die Node.js-Plattform legt die PORT-Umgebungsvariable auf den Port fest, an den der Proxy-Server den Datenverkehr weiterleitet. Weitere Informationen finden Sie unter Konfigurieren des Proxy-Server.

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Konfigurieren einer Amazon Linux AMI (Vorgängerversion von Amazon Linux 2) Node.js-Umgebung

Die folgenden Konfigurationskategorien für Konsolensoftware werden nur in einer Elastic-Beanstalk-Node.js-Umgebung unterstützt, die eine Amazon-Linux-AMI-Plattformversion (vor Amazon Linux 2) verwendet.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/AL2 Plattformbranches sind nicht mit früheren

Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.

- Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Container-Optionen — Amazon Linux AMI (AL1)

Geben Sie auf der Konfigurationsseite Folgendes ein:

- Proxy Server (Proxy-Server) – Gibt den Webserver an, der für Proxy-Verbindungen zu Node.js verwendet werden soll. Standardmäßig wird NGINX verwendet. Wenn Sie none (keine) auswählen, sind statische Dateizuordnungen nicht wirksam und die GZIP-Komprimierung ist deaktiviert.
- Node.js-Version – Gibt die Version von Node.js an. Die Liste der unterstützten Node.js-Versionen finden Sie unter Node.js im Handbuch für AWS Elastic Beanstalk -Plattformen.
- GZIP-Komprimierung – Gibt an, ob die GZIP-Komprimierung aktiviert ist. Die GZIP-Komprimierung ist standardmäßig aktiviert.
- Node Command (Knotenbefehl) – Hier können Sie den Befehl eingeben, der zum Starten der Node.js-Anwendung verwendet wird. Eine leere Zeichenfolge (Standard) bedeutet, dass Elastic Beanstalk erst `app.js`, dann `server.js` und dann `npm start` verwendet.

Node.js-Konfigurations-Namespace

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben durchführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Sie können mittels des Namespace `aws:elasticbeanstalk:environment:proxy` den Proxy auswählen, der für die Instances Ihrer Umgebung verwendet werden soll. Im folgenden Beispiel wird Ihre Umgebung für die Verwendung des Apache HTTPD-Proxy-Server konfiguriert.

Example.ebextensions/nodejs-settings.config

```
option_settings:
```

```
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

Sie können den Proxy so konfigurieren, dass statische Dateien mittels des Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` bereitgestellt werden. Weitere Informationen sowie ein Beispiel finden Sie unter the section called “Statische Dateien”.

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Die Amazon Linux AMI (Vorgängerversion von Amazon Linux 2) Node.js-Plattform

Wenn die Elastic-Beanstalk-Node.js-Umgebung eine Amazon-Linux-AMI-Plattformversion verwendet (vor Amazon Linux 2), beachten Sie die speziellen Konfigurationen und Empfehlungen in diesem Abschnitt.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/AL2 Plattformbranches sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
> - Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Node.jsplattformspezifische Konfigurationsoptionen — Amazon Linux AMI () AL1

Elastic Beanstalk unterstützt einige plattformspezifische Konfigurationsoptionen für Amazon Linux AMI Node.js-Plattformversionen. Sie können auswählen, welchen Proxy-Server Sie vor Ihrer Anwendung ausführen möchten, eine bestimmte Version von Node.js für die Ausführung verwenden und den Befehl wählen, um Ihre Anwendung auszuführen.

Für Proxyserver können Sie einen NGINX- oder Apache-Proxy-Server verwenden. Sie können den Wert none auf die Option `ProxyServer` festlegen. Mit dieser Einstellung führt Elastic Beanstalk Ihre Anwendung als eigenständige Anwendung aus und nicht hinter einem Proxy-Server. Wenn in Ihrer Umgebung eine eigenständige Anwendung ausgeführt wird, aktualisieren Sie den Code so, dass dem Port zugehört wird, an den NGINX den Datenverkehr weiterleitet.

```
var port = process.env.PORT || 8080;

app.listen(port, function() {
  console.log('Server running at http://127.0.0.1:%s', port);
});
```

Node.jsSprachversionen — Amazon Linux AMI (AL1)

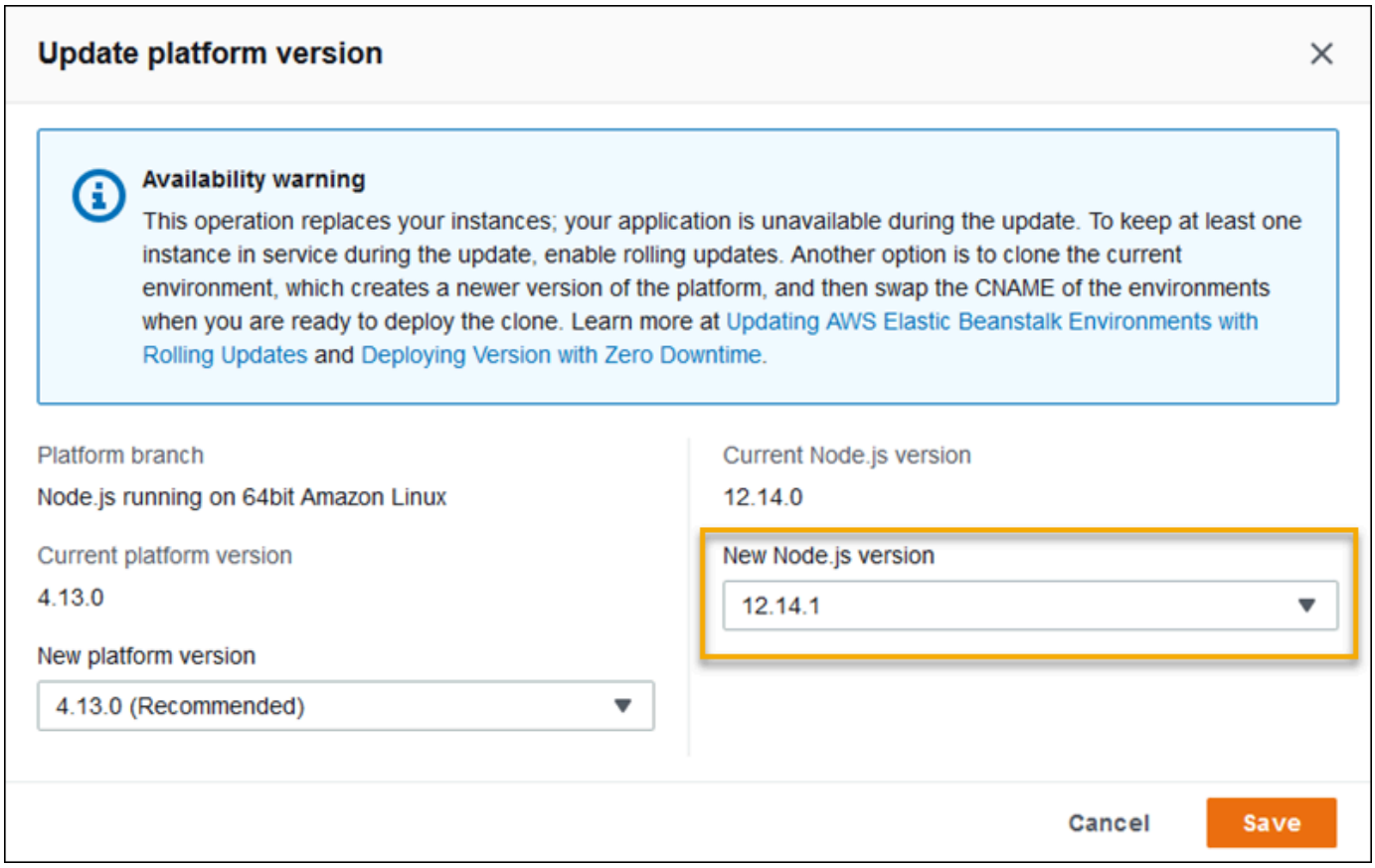
In Bezug auf die unterstützte Sprachversion unterscheidet sich die Amazon-Linux-AMI-Node.js-Plattform von anderen von Elastic Beanstalk verwalteten Plattformen. Das liegt daran, dass jede Node.js-Plattformversion nur wenige Node.js-Sprachversionen unterstützt. Die Liste der unterstützten Node.js-Versionen finden Sie unter Node.js im Handbuch für AWS Elastic Beanstalk -Plattformen.

Sie können eine plattformspezifische Konfigurationsoption verwenden, um die Sprachversion festzulegen. Detaillierte Anweisungen finden Sie unter the section called “Konfiguration der Node.js-Umgebung”. Alternativ können Sie die Elastic-Beanstalk-Konsole verwenden, um die Node.js-Version zu aktualisieren, die Ihre Umgebung als Teil der Aktualisierung Ihrer Plattformversion verwendet.

> **Note**
>
> Wenn die von Ihnen verwendete Version von Node.js von der Plattform nicht mehr unterstützt wird, müssen Sie die Versionseinstellung vor der Ausführung einer Plattformaktualisierung ändern oder löschen. Dieser Fall kann eintreten, wenn eine Schwachstelle für eine oder mehrere Versionen von Node.js identifiziert wird.
> In diesem Fall schlägt der Versuch NodeVersionfehl, auf eine neue Version der Plattform zu aktualisieren, die die konfigurierte Version nicht unterstützt. Um zu vermeiden, dass eine neue Umgebung erstellt werden muss, ändern Sie die NodeVersionKonfigurationsoption in eine Version von Node.js, die sowohl von der alten als auch von der neuen Plattformversion unterstützt wird, oder entfernen Sie die Optionseinstellung und führen Sie dann das Plattformupdate durch.

So konfigurieren Sie die Node.js-Version Ihrer Umgebung in der Elastic-Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite unter Platform (Plattform) die Option Change (Ändern) aus.
4. Wählen Sie im Dialogfeld Plattformversion aktualisieren eine Node.js-Plattformversion aus.



5. Wählen Sie Save aus.

Node.jsKonfigurations-Namespaces — Amazon Linux AMI () AL1

Die Plattform von Node.js Amazon Linux AMI definiert zusätzliche Optionen in den `aws:elasticbeanstalk:container:nodejs:staticfiles`- und `aws:elasticbeanstalk:container:nodejs`-Namespaces.

Die folgende Konfigurationsdatei weist Elastic Beanstalk an, `npm start` zum Ausführen der Anwendung zu verwenden. Sie legt auch den Proxytyp auf Apache fest und aktiviert die Komprimierung. Zuletzt konfiguriert sie den Proxy, um statische Dateien aus zwei Quellverzeichnissen bereitzustellen. Eine Quelle besteht aus HTML -Dateien unter dem Pfad `html` im Stammverzeichnis der Website aus dem Quellverzeichnis von `statichtml`. Die andere Quelle besteht aus Image-Dateien unter dem Pfad `images` im Stammverzeichnis der Website aus dem Quellverzeichnis von `staticimages`.

Example.ebextensions/node-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:nodejs:
    NodeCommand: "npm start"
    ProxyServer: apache
    GzipCompression: true
  aws:elasticbeanstalk:container:nodejs:staticfiles:
    /html: statichtml
    /images: staticimages
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Konfiguration benutzerdefinierter Startbefehle mit einem Procfile auf Elastic Beanstalk

Sie können eine Datei namens `Procfile` in das Stammverzeichnis des Quell-Bundles einfügen, um den Befehl anzugeben, mit dem Ihre Anwendung gestartet wird.

Example Procfile

```
web: node index.js
```

Informationen zur `Procfile` Verwendung finden Sie unter. Buildfile und Procfile

> **Note**
>
> Dieses Feature ersetzt die ältere Option `NodeCommand` im Namespace `aws:elasticbeanstalk:container:nodejs`.

Konfiguration der Abhängigkeiten Ihrer Anwendung von Elastic Beanstalk

Ihre Anwendung verfügt möglicherweise über Abhängigkeiten von einigen Node.js-Modulen, wie z. B. Modulen, die Sie in `require()`-Anweisungen angeben. Diese Module werden in einem `node_modules`-Verzeichnis gespeichert. Wenn Ihre Anwendung ausgeführt wird, lädt Node.js die Module aus diesem Verzeichnis. Weitere Informationen finden Sie in der Node.js-Dokumentation unter Laden aus den Ordnern „node_modules“.

Sie können diese Modul-Abhängigkeiten mithilfe einer `package.json`-Datei angeben. Wenn Elastic Beanstalk diese Datei erkennt und kein `node_modules`-Verzeichnis vorhanden ist, führt Elastic Beanstalk `npm install` als der webapp-Benutzer aus. Der `npm install`-Befehl installiert die Abhängigkeiten in dem `node_modules`-Verzeichnis, das Elastic Beanstalk zuvor erstellt hat. Der `npm install`-Befehl greift auf die in der `package.json`-Datei aufgeführten Pakete aus der öffentlichen npm-Registrierung oder von anderen Orten zu. Weitere Informationen finden Sie auf der Website npm-Dokumente.

Wenn Elastic Beanstalk das `node_modules`-Verzeichnis erkennt, führt Elastic Beanstalk `npm install` nicht aus, auch wenn eine `package.json`-Datei existiert. Elastic Beanstalk geht davon aus, dass die Abhängigkeitspakete im `node_modules`-Verzeichnis verfügbar sind, sodass sie von Node.js abgerufen und geladen werden können.

In den folgenden Abschnitten erhalten Sie weitere Informationen zu den Node.js-Modulabhängigkeiten für Ihre Anwendung.

> **Note**
>
> Wenn bei der Ausführung von `npm install` durch Elastic Beanstalk Probleme bei der Bereitstellung auftreten, sollten Sie einen alternativen Ansatz in Betracht ziehen. Fügen Sie das `node_modules`-Verzeichnis mit den Abhängigkeiten zu Ihrem Anwendungs-Quell-Bundle hinzu. Dadurch können Sie Probleme bei der Installation von Abhängigkeiten aus der öffentlichen npm-Registry umgehen, während Sie das Problem untersuchen. Da die Abhängigkeitsmodule aus einem lokalen Verzeichnis stammen, kann dies auch dabei helfen,

die Bereitstellungszeit zu verkürzen. Weitere Informationen finden Sie unter Einschließen von Node.js-Abhängigkeiten in ein node_modules-Verzeichnis.

Angeben von Node.js-Abhängigkeiten mit einer package.json-Datei

Sie können mithilfe einer `package.json`-Datei im Stammverzeichnis Ihrer Projektquelle Abhängigkeitspakete angeben und einen Startbefehl bereitstellen. Wenn eine `package.json`-Datei vorhanden ist und kein `node_modules`-Verzeichnis im Stammverzeichnis Ihrer Projektquelle vorhanden ist, wird Elastic Beanstalk `npm install` als webapp-Benutzer ausführen, um Abhängigkeiten aus der öffentlichen npm-Registry zu installieren. Außerdem wird Elastic Beanstalk die Anwendung mit dem Befehl `start` starten. Weitere Informationen zur `package.json`-Datei finden Sie unter Angeben von Abhängigkeiten in einer `package.json`-Datei auf der npm-Docs-Website.

Geben Sie mit dem Schlüsselwort `scripts` einen Startbefehl an. Derzeit wird das Schlüsselwort `scripts` anstelle der Legacy-Option `NodeCommand` im Namespace `aws:elasticbeanstalk:container:nodejs` verwendet.

Example package.json – Express

```
{
    "name": "my-app",
    "version": "0.0.1",
    "private": true,
    "dependencies": {
      "ejs": "latest",
      "aws-sdk": "latest",
      "express": "latest",
      "body-parser": "latest"
    },
    "scripts": {
      "start": "node app.js"
    }
  }
```

Produktionsmodus und Entwicklungsabhängigkeiten

Verwenden Sie die Attribute dependencies und devDependencies, um Ihre Abhängigkeiten in der `package.json`-Datei anzugeben. Das Attribut dependencies bezeichnet Pakete, die Ihre

Anwendung in der Produktion benötigt. Das Attribut devDependencies bezeichnet Pakete, die nur für die lokale Entwicklung und das Testen benötigt werden.

Wenn Sie die devDependencies-Pakete installieren müssen, setzen Sie die Umgebungseigenschaft NPM_USE_PRODUCTION auf `false`. Mit dieser Einstellung werden wir die obigen Optionen nicht verwenden, wenn wir npm install ausführen. Dies führt dazu, dass die devDependencies-Pakete installiert werden.

Einschließen von Node.js-Abhängigkeiten in ein node_modules-Verzeichnis

Wenn Sie Abhängigkeitspakete zusammen mit Ihrem Anwendungscode in Umgebungsinstances bereitstellen möchten, fügen Sie sie in das Verzeichnis `node_modules` im Stammverzeichnis der Projektquelle ein. Weitere Informationen finden Sie unter Pakete lokal herunterladen und installieren auf der npm-Docs-Website.

Wenn Sie ein `node_modules` Verzeichnis für eine AL2 Node.js Plattformversion von AL2023/ bereitstellen, geht Elastic Beanstalk davon aus, dass Sie Ihre eigenen Abhängigkeitspakete bereitstellen, und vermeidet die Installation von Abhängigkeiten, die in einer package.json-Datei angegeben sind. Node.jssucht im Verzeichnis nach Abhängigkeiten. `node_modules` Weitere Informationen finden Sie unter Laden aus node_modules-Ordnern in der Node.js-Dokumentation.

> **Note**
>
> Wenn bei der Ausführung von `npm install` durch Elastic Beanstalk Probleme bei der Bereitstellung auftreten, sollten Sie bei der Untersuchung des Problems den in diesem Thema beschriebenen Ansatz als Workaround verwenden.

Überlegungen zu Node.js auf Amazon Linux 2

Lesen Sie diesen Abschnitt, wenn Sie einen Plattformzweig von Node.js verwenden, der auf Amazon Linux 2 ausgeführt wird.

Überlegungen zu Node.js auf Amazon Linux 2

> **Note**
>
> Die Informationen in diesem Thema beziehen sich auf Node.js Plattformbranches, die auf Amazon Linux 2 ausgeführt werden. In diesem Artikel werden AL2 spezifische Funktionen und Verhaltensweisen beschrieben, die sich von AL2023 unterscheiden.

Varianten von Befehlen

Die Befehlsoptionen variieren je nach der npm-Version, die im Amazon Linux 2-Plattformzweig enthalten ist, auf dem Ihre Anwendung ausgeführt wird.

- npm v6 – Elastic Beanstalk installiert die Abhängigkeiten standardmäßig im Produktionsmodus. Es verwendet den Befehl `npm install --production`.
- npm v7 oder höher – Elastic Beanstalk lässt die devDependencies weg. Es verwendet den Befehl `npm install --omit=dev`.

Beide oben aufgeführten Befehle installieren keine Pakete, bei denen es sich um devDependencies handelt.

SSH- und HTTPS-Protokolle für Git-Abhängigkeiten

Ab der Amazon Linux 2-Plattformversion vom 7. März 2023 können Sie die Protokolle SSH und HTTPS verwenden, um Pakete aus einem Git-Repository abzurufen. Plattformzweig Node.js 16 unterstützt sowohl das SSH- als auch das HTTPS-Protokoll. Node.js 14 unterstützt nur das HTTPS-Protokoll.

Example package.json – Node.js 16 unterstützt sowohl HTTPS als auch SSH

```
...
"dependencies": {
  "aws-sdk": "https://github.com/aws/aws-sdk-js.git",
  "aws-chime": "git+ssh://git@github.com:aws/amazon-chime-sdk-js.git"
}
```

Versionen und Versionsbereiche

Verwenden Sie das Schlüsselwort `engines` in der Datei `package.json`, um die Node.js-Version anzugeben, die Ihre Anwendung verwenden soll. Mit der npm-Notation können Sie auch einen Versionsbereich angeben. Weitere Informationen zur Syntax für Versionsbereiche finden Sie unter Semantische Versionsverwaltung mit npm auf der Website Node.js. Das Schlüsselwort `engines` in der Node.js `package.json`-Datei ersetzt die ältere Option `NodeVersion` im `aws:elasticbeanstalk:container:nodejs`-Namespace.

> **Important**
>
> Die Funktion zur Angabe von Versionsbereichen ist für Plattformzweige von Node.js, auf AL2023 denen sie ausgeführt werden, nicht verfügbar. Wir unterstützen nur eine Version von Node.js innerhalb einer bestimmten Node.js -Verzweigung auf AL2023. Wenn Ihre `package.json`-Datei einen Versionsbereich angibt, ignorieren wir diesen und verwenden standardmäßig die Version des Plattformzweigs von Node.js.

Example **`package.json`** – Einzelne Node.js-Version

```
{
    ...
    "engines": { "node" : "14.16.0" }
  }
```

Example **`package.json`** – Node.js Versionsbereich

```
{
    ...
    "engines": { "node" : ">=10 <11" }
  }
```

Wenn ein Versionsbereich angegeben wird, installiert Elastic Beanstalk die neueste Node.js-Version, die der Plattform innerhalb des Bereichs zur Verfügung steht. In diesem Beispiel gibt der Bereich an, dass die Version größer oder gleich als Version 10 sein muss, aber kleiner als Version 11. Infolgedessen installiert Elastic Beanstalk die neueste Version von Node.js 10.x.y, die auf der unterstützten Plattform verfügbar ist.

Beachten Sie, dass Sie jeweils nur eine Node.js-Version angeben können, die Ihrem Plattformzweig entspricht. Wenn Sie beispielsweise den Plattformzweig Node.js 16 verwenden, können Sie nur eine 16.x.y-Version Node.js angeben. Sie können die von npm unterstützten Versionsbereichsoptionen

verwenden, um eine größere Flexibilität zu ermöglichen. Gültige Versionen von Node.js für jeden Plattformzweig finden Sie unter Node.js im Handbuch AWS Elastic Beanstalk -Plattformen.

> **Note**
>
> Wenn die von Ihnen verwendete Version von Node.js von der Plattform nicht mehr unterstützt wird, müssen Sie die Einstellung der Node.js-Version vor der Ausführung einer Plattform-Aktualisierung ändern oder löschen. Dieser Fall kann eintreten, wenn eine Schwachstelle bei einer oder mehreren Versionen von Node.js identifiziert wird.
> In diesem Fall schlägt die Aktualisierung auf eine neue Plattformversion, die die konfigurierte Node.js-Version nicht unterstützt, fehl. Um zu vermeiden, dass eine neue Umgebung erstellt werden muss, ändern Sie die Einstellung der Node.js-Version in `package.json` in eine Node.js-Version, die sowohl von der alten als auch von der neuen Version unterstützt wird. Sie können einen Node.js-Versionsbereich angeben, der eine unterstützte Version enthält, wie oben in diesem Thema beschrieben. Außerdem können Sie die Einstellung entfernen und dann das neue Quellpaket bereitstellen.

Sperren von Abhängigkeiten mit npm shrinkwrap auf Elastic Beanstalk

Die Node.js-Plattform führt `npm install` als webapp-Benutzer bei jeder Bereitstellung aus. Wenn neue Versionen Ihrer Abhängigkeiten verfügbar sind, werden diese installiert, wenn Sie Ihre Anwendung bereitstellen. Dadurch kann die Bereitstellung möglicherweise sehr lange dauern.

Sie können eine Aktualisierung Ihrer Abhängigkeiten vermeiden, indem Sie eine `npm-shrinkwrap.json`-Datei erstellen, die die Abhängigkeiten Ihrer Anwendung von der aktuellen Version sperrt.

```
$ npm install
$ npm shrinkwrap
wrote npm-shrinkwrap.json
```

Fügen Sie diese Datei in Ihr Quell-Bundle sein, um sicherzustellen, dass Abhängigkeiten nur einmal installiert werden.

Konfigurieren des Proxy-Server

Elastic Beanstalk kann NGINX oder Apache-HTTPD als Reverse-Proxy verwenden, um die Anwendung dem Load Balancer von Elastic Load Balancing auf Port 80 zuzuweisen. Der

Standardwert ist NGINX. Elastic Beanstalk bietet eine Proxy-Standard-Proxykonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Standardmäßig wird der Proxy von Elastic Beanstalk für die Weiterleitung von Anforderungen an die Anwendung auf Port 5000 konfiguriert. Sie können den Standardport überschreiben, indem Sie für die `PORT`-Umgebungseigenschaft den Überwachungsport Ihrer Hauptanwendung festlegen.

> **Note**
>
> Der Port, den Ihre Anwendung überwacht, hat keinen Einfluss auf den Port, den der NGINX-Server überwacht, um Anfragen vom Load Balancer zu empfangen.

Konfigurieren des Proxy-Servers auf Ihrer Plattformversion

Alle AL2023 AL2 /-Plattformen unterstützen eine einheitliche Proxy-Konfigurationsfunktion. Weitere Informationen zur Konfiguration des Proxyservers auf Ihren Plattformversionen, auf denen AL2023/ ausgeführt wirdAL2, finden Sie unterReverse-Proxy-Konfiguration.

Konfigurieren des Proxys in Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)

Wenn Ihre Elastic Beanstalk Node.js-Umgebung eine Amazon-Linux-AMI-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie die Informationen in diesem Abschnitt.

> **Hinweise**
>
> - Die Informationen in diesem Thema gelten nur für Plattformbranches, die auf Amazon Linux AMI (AL1) basieren. AL2023/AL2 Plattformbranches sind nicht mit früheren Amazon Linux AMI (AL1) -Plattformversionen kompatibel und erfordern andere Konfigurationseinstellungen.
> - Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Erweiterung und Außerkraftsetzung der Standard-Proxykonfiguration — Amazon Linux AMI () AL1

Die Node.js-Plattform verwendet einen Reverse-Proxy-Server, um Anforderungen von Port 80 auf der Instance an die Anwendung weiterzuleiten, welche Port 8081 überwacht. Elastic Beanstalk bietet eine Proxy-Standard-Proxykonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Um die Standardkonfiguration zu erweitern, fügen Sie `.conf`-Dateien zu `/etc/nginx/conf.d` mit einer Konfigurationsdatei hinzu. Ein spezifisches Beispiel finden Sie unter Beenden von HTTPS auf EC2-Instances, auf denen Node.js ausgeführt wird.

Die Node.js-Plattform legt die PORT-Umgebungsvariable auf den Port fest, an den der Proxy-Server den Datenverkehr weiterleitet. Lesen Sie diese Variable in Ihrem Code, um den Port Ihrer Anwendung zu konfigurieren.

```
    var port = process.env.PORT || 3000;

    var server = app.listen(port, function () {
        console.log('Server running at http://127.0.0.1:' + port + '/');
    });
```

Die standardmäßige NGINX-Konfiguration leitet Datenverkehr an einen Upstream-Server mit dem Namen `nodejs` auf `127.0.0.1:8081` weiter. Sie können die Standardkonfiguration entfernen und Ihre eigene in einer Konfigurationsdatei bereitstellen.

Example.ebextensions/proxy.config

Im folgenden Beispiel wird die Standard-Konfiguration entfernt und eine benutzerdefinierte Konfiguration hinzugefügt, die Datenverkehr an Port 5000 anstatt an 8081 weiterleitet.

```
files:
  /etc/nginx/conf.d/proxy.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      upstream nodejs {
        server 127.0.0.1:5000;
        keepalive 256;
      }

      server {
```

```
        listen 8080;

        if ($time_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})T(\d{2})") {
            set $year $1;
            set $month $2;
            set $day $3;
            set $hour $4;
        }
        access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
 healthd;
        access_log  /var/log/nginx/access.log  main;

        location / {
            proxy_pass  http://nodejs;
            proxy_set_header   Connection "";
            proxy_http_version 1.1;
            proxy_set_header        Host            $host;
            proxy_set_header        X-Real-IP       $remote_addr;
            proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
        }

        gzip on;
        gzip_comp_level 4;
        gzip_types text/html text/plain text/css application/json application/x-
javascript text/xml application/xml application/xml+rss text/javascript;

        location /static {
            alias /var/app/current/static;
        }

      }

  /opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh:
    mode: "000755"
    owner: root
    group: root
    content: |
      #!/bin/bash -xe
      rm -f /etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf
      service nginx stop
      service nginx start

container_commands:
  removeconfig:
```

```
    command: "rm -f /tmp/deployment/config/
#etc#nginx#conf.d#00_elastic_beanstalk_proxy.conf /etc/nginx/
conf.d/00_elastic_beanstalk_proxy.conf"
```

Die Beispielkonfiguration (`/etc/nginx/conf.d/proxy.conf`) verwendet die Standardkonfiguration unter `/etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf` als Basis, um den Standard-Serverblock mit Komprimierungs- und Protokolleinstellungen und eine statische Dateizuordnung einzuschließen.

Der `removeconfig`-Befehl entfernt die Standardkonfiguration für den Container, um sicherzustellen, dass der Proxy-Server die benutzerdefinierte Konfiguration verwendet. Elastic Beanstalk erstellt die Standardkonfiguration während jeder Konfigurationsbereitstellung neu. Um dies zu berücksichtigen, wird im folgenden Beispiel ein post-configuration-deployment Hook (`/opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh`) hinzugefügt. Dadurch wird die Standardkonfiguration entfernt und der Proxy-Server neu gestartet.

> **Note**
>
> Die Standardkonfiguration kann sich in späteren Versionen der Node.js-Plattform ändern. Verwenden Sie die neueste Version der Konfiguration als Grundlage für Ihre Anpassungen, um Kompatibilität sicherzustellen.

Wenn Sie die Standardkonfiguration überschreiben, müssen Sie alle statischen Dateizuordnungen und die GZIP-Komprimierung definieren. Das liegt daran, dass die Plattform die Standardeinstellungen nicht anwenden kann.

Weitere Elastic Beanstalk Beanstalk-Beispielanwendungen und Tutorials für Node.js

Dieser Abschnitt enthält zusätzliche Anwendungen und Tutorials. Das zuvor in diesem QuickStart für Node.js Thema befindliche Thema führt Sie durch den Start der Beispielanwendung Node.js mit der EB CLI.

Um mit den Anwendungen Node.js loszulegen AWS Elastic Beanstalk, benötigen Sie lediglich ein Anwendungsquellpaket, das Sie als erste Anwendungsversion hochladen und in einer Umgebung bereitstellen können.

Starten einer Umgebung mit einer Beispielanwendung von Node.js

Elastic Beanstalk bietet einseitige Beispielanwendungen für jede Plattform sowie komplexere Beispiele, die die Verwendung zusätzlicher AWS Ressourcen wie Amazon RDS und sprach- oder plattformspezifische Funktionen und zeigen. APIs

> **Note**
>
> Folgen Sie den Schritten in der `README.md`-Quellpaketdatei, um sie bereitzustellen.

Beispiele

Umgebungstyp	Quell-Bundle	Description
Webserver	nodejs.zip	Anwendung mit einer Seite. Informationen zum Starten der Beispielanwendung mit der EB-CLI finden Sie unterQuickStart für Node.js. Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.
Webserver mit Amazon RDS	nodejs-example-express-rds.zip	Hiking-Protokollanwendung, die das Express-Framework und eine Amazon Relational Database Service (RDS)-Datenbank verwendet. Praktische Anleitung
Webserver mit Amazon ElastiCache	nodejs-example-express-elasticache.zip	Express-Webanwendung, die Amazon ElastiCache für Clustering verwendet. Mit Clustering lassen sich hohe Verfügbarkeit, Leistung und Sicherheit Ihrer Webanwendung verbessern. Praktische Anleitung

Umgebungstyp	Quell-Bundle	Description
Webserver mit DynamoDB, Amazon SNS und Amazon SQS	nodejs-example-dynamo.zip	Express-Website, die Kontaktinformationen der Benutzer für eine neue Marketingkampagne des Unternehmens sammelt. Verwendet das AWS SDK für JavaScript in Node.js, um Einträge in eine DynamoDB-Tabelle zu schreiben, und Elastic Beanstalk Beanstalk-Konfigurationsdateien, um Ressourcen in DynamoDB, Amazon SNS und Amazon SQS zu erstellen. Praktische Anleitung

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine gänzlich andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen. Weitere Informationen zur Anwendungsbereitstellung finden Sie unter Bereitstellen einer neuen Version Ihrer Anwendung.

Nachdem Sie ein oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, Node.js -Anwendungen lokal zu entwickeln und auszuführen, lesen Sie, wie Sie Einrichtung Ihrer Node.js Entwicklungsumgebung für Elastic Beanstalk eine Node.js -Entwicklungsumgebung mit allen Tools einrichten, die Sie benötigen.

Eine Node.js Express-Anwendung auf Elastic Beanstalk bereitstellen

In diesem Abschnitt erfahren Sie, wie Sie mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) eine Beispielanwendung auf Elastic Beanstalk bereitstellen und anschließend die Anwendung für die Verwendung des Express-Frameworks aktualisieren.

Voraussetzungen

Für dieses Tutorial sind die folgenden Voraussetzungen erforderlich:

- Die Node.js-Laufzeiten
- Die Standard-Paketmanager-Software von Node.js, npm
- Der Express-Befehlszeilengenerator
- Die Elastic Beanstalk Command Line Interface (EB CLI)

Weitere Informationen zur Installation dieser ersten drei Komponenten und zur Einrichtung Ihrer lokalen Entwicklungsumgebung finden Sie unter Einrichtung Ihrer Node.js Entwicklungsumgebung für Elastic Beanstalk. Für dieses Tutorial müssen Sie das AWS SDK für Node.js nicht installieren, das auch im referenzierten Thema erwähnt wird.

Detaillierte Anweisungen zum Installieren und Konfigurieren der EB-CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Erstellen einer Elastic Beanstalk-Umgebung

Ihr Anwendungsverzeichnis

In diesem Tutorial wird ein Verzeichnis verwendet, das als `nodejs-example-express-rds` für das Anwendungsquellpaket bezeichnet wird. Erstellen Sie das `nodejs-example-express-rds`-Verzeichnis für dieses Tutorial.

```
~$ mkdir nodejs-example-express-rds
```

> **Note**
>
> Jedes Tutorial in diesem Kapitel verwendet ein eigenes Verzeichnis für das Anwendungsquellpaket. Der Verzeichnisname entspricht dem Namen der Beispielanwendung, die im Tutorial verwendet wird.

Ändern Sie Ihr aktuelles Arbeitsverzeichnis zu `nodejs-example-express-rds`.

```
~$ cd nodejs-example-express-rds
```

Jetzt können wir eine Elastic-Beanstalk-Umgebung mit der Node.js und der Beispielanwendung einrichten. Wir werden die Elastic-Beanstalk-Befehlszeilenschnittstelle (EB-CLI) verwenden.

So konfigurieren Sie ein EB-CLI-Repository für Ihre Anwendung und erstellen eine Elastic-Beanstalk-Umgebung mit der Node.js-Plattform

1. Erstellen Sie ein Repository mit dem Befehl eb init.

   ```
   ~/nodejs-example-express-rds$ eb init --platform node.js --region <region>
   ```

   Mit diesem Befehl wird eine Konfigurationsdatei in einem Ordner mit dem Namen `.elasticbeanstalk` generiert. Sie gibt die Einstellungen für die Erstellung von Umgebungen für Ihre Anwendung an und erstellt eine Elastic Beanstalk-Anwendung mit dem Namen des aktuellen Ordners.

2. Erstellen Sie mit dem Befehl eb create eine Umgebung mit einer laufenden Beispielanwendung.

   ```
   ~/nodejs-example-express-rds$ eb create --sample nodejs-example-express-rds
   ```

   Mit diesem Befehl wird eine Umgebung mit Lastausgleich, den Standardeinstellungen für die Node.js-Plattform und den folgenden Ressourcen erstellt:

   - EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

     Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.
   - Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
   - Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
   - Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass

HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.

- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation Stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

3. Wenn die Umgebungserstellung abgeschlossen ist, verwenden Sie den Befehl eb open, um die URL der Umgebung im Standardbrowser zu öffnen.

```
~/nodejs-example-express-rds$ eb open
```

Sie haben jetzt eine Node.js Elastic-Beanstalk-Umgebung mit einer Beispielanwendung erstellt. Sie können sie mit Ihrer eigenen Anwendung aktualisieren. Als nächstes aktualisieren wir die Beispielanwendung, um das Express-Framework zu verwenden.

Aktualisieren der Anwendung zur Nutzung von Express

Nachdem Sie eine Umgebung mit einer Beispielanwendung erstellt haben, können Sie diese mit Ihrer eigenen Anwendung aktualisieren. In diesem Verfahren führen wir zunächst die express- und npm install-Befehle aus, um das Express-Framework in Ihrem Anwendungsverzeichnis einzurichten. Dann verwenden wir die EB CLI, um Ihre Elastic-Beanstalk-Umgebung mit der aktualisierten Anwendung zu aktualisieren.

So aktualisieren Sie die Anwendung zur Nutzung von Express

1. Führen Sie den Befehl `express` aus. Dadurch werden `package.json`, `app.js` und weitere Verzeichnisse generiert.

   ```
   ~/nodejs-example-express-rds$ express
   ```

   Wenn Sie gefragt werden, ob Sie fortfahren möchten, geben Sie **y** ein.

   > **Note**
   >
   > Wenn der express-Befehl nicht funktioniert, haben Sie den Express-Befehlszeilengenerator möglicherweise nicht installiert, wie im vorherigen Abschnitt mit den Voraussetzungen beschrieben. Oder die Verzeichnispfadeinstellung für Ihren lokalen Computer muss möglicherweise eingerichtet werden, um den express-Befehl auszuführen. Im Abschnitt Voraussetzungen finden Sie detaillierte Schritte zum Einrichten Ihrer Entwicklungsumgebung, sodass Sie mit diesem Tutorial fortfahren können.

2. Richten Sie lokale Abhängigkeiten ein.

   ```
   ~/nodejs-example-express-rds$ npm install
   ```

3. (Optional) Stellen Sie sicher, dass der Web-Anwendungs-Server startet.

   ```
   ~/nodejs-example-express-rds$ npm start
   ```

   Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

   ```
   > nodejs@0.0.0 start /home/local/user/node-express
   ```

```
> node ./bin/www
```

Der Server läuft standardmäßig über Port 3000. Führen Sie als Test `curl http://localhost:3000` in einem anderen Terminal aus oder öffnen Sie einen Browser auf dem lokalen Computer und geben Sie die URL `http://localhost:3000` ein.

Drücken Sie Strg+C, um den Server zu stoppen.

4. Stellen Sie die Änderungen in Ihrer Elastic-Beanstalk-Umgebung mit dem Befehl eb deploy bereit.

```
~/nodejs-example-express-rds$ eb deploy
```

5. Sobald die Umgebung grün und einsatzbereit ist, aktualisieren Sie die URL, um die ordnungsgemäße Funktionsweise zu prüfen. Sie sollten eine Webseite mit dem Text Welcome to Express sehen.

Aktualisieren wir jetzt die Express-Anwendung für die Bereitstellung statischer Dateien und das Hinzufügen einer neuen Seite.

So konfigurieren Sie statische Dateien und fügen eine neue Seite zu Ihrer Express-Anwendung hinzu

1. Fügen Sie eine zweite Konfigurationsdatei in den `.ebextensions`-Ordner mit dem folgenden Inhalt hinzu:

`nodejs-example-express-rds/.ebextensions/staticfiles.config`

```
option_settings:
    aws:elasticbeanstalk:environment:proxy:staticfiles:
        /stylesheets: public/stylesheets
```

Mit dieser Einstellung wird konfiguriert, dass der Proxy-Server Dateien im `public`-Ordner im Pfad `/public` der Anwendung bereitstellt. Das statische Bereitstellen von Dateien vom Proxy-Server reduziert die Last Ihrer Anwendung. Weitere Informationen finden Sie unter Statische Dateien weiter oben in diesem Kapitel.

2. (Optional) Um zu überprüfen, ob statische Mappings korrekt konfiguriert sind, kommentieren Sie die statische Mapping-Konfiguration unter `nodejs-example-express-rds/app.js` aus. Dadurch wird das Mapping aus der Knotenanwendung entfernt.

```
//  app.use(express.static(path.join(__dirname, 'public')));
```

Die statischen Dateizuordnungen in der `staticfiles.config`-Datei aus dem vorherigen Schritt sollten das Stylesheet immer noch erfolgreich laden, auch wenn Sie diese Zeile auskommentiert haben. Um zu überprüfen, ob die statischen Dateizuordnungen über die statische Proxydateikonfiguration und nicht über die Express-Anwendung geladen werden, entfernen Sie die auf `option_settings:` folgenden Werte. Nachdem es sowohl aus der statischen Dateikonfiguration als auch aus der Knotenanwendung entfernt wurde, kann das Stylesheet nicht geladen werden.

Denken Sie daran, den Inhalt `nodejs-example-express-rds/app.js` und `staticfiles.config` nach dem Testen zurückzusetzen.

3. Fügen Sie `nodejs-example-express-rds/routes/hike.js` hinzu. Geben Sie Folgendes ein:

```
exports.index = function(req, res) {
 res.render('hike', {title: 'My Hiking Log'});
};

exports.add_hike = function(req, res) {
};
```

4. Aktualisieren Sie `nodejs-example-express-rds/app.js`, um drei neue Zeilen hinzuzufügen.

   Fügen Sie zuerst die folgende Zeile hinzu, um einen `require` für diese Route hinzuzufügen:

```
var hike = require('./routes/hike');
```

   Die Datei sollte wie im folgenden Ausschnitt aussehen:

```
var express = require('express');
var path = require('path');
var hike = require('./routes/hike');
```

   Fügen Sie anschließend die folgenden beiden Zeilen zu `nodejs-example-express-rds/app.js` nach `var app = express();` hinzu.

```
app.get('/hikes', hike.index);
app.post('/add_hike', hike.add_hike);
```

Die Datei sollte wie im folgenden Ausschnitt aussehen:

```
var app = express();
app.get('/hikes', hike.index);
app.post('/add_hike', hike.add_hike);
```

5. Kopieren Sie `nodejs-example-express-rds/views/index.jade` in `nodejs-example-express-rds/views/hike.jade`.

```
~/nodejs-example-express-rds$ cp views/index.jade views/hike.jade
```

6. Stellen Sie die Änderungen mit dem eb deploy-Befehl bereit.

```
~/nodejs-example-express-rds$ eb deploy
```

7. Die Umgebung wird nach einigen Minuten aktualisiert. Nachdem Ihre Umgebung grün und bereit ist, überprüfen Sie das Funktionieren durch Aktualisieren Ihres Browsers und Anfügen von **`hikes`** am Ende der URL (z. B. `http://node-express-env-syypntcz2q.elasticbeanstalk.com/hikes`).

   Sie sollten eine Webseite mit dem Titel My Hiking Log (Mein Hiking-Protokoll) sehen.

Sie haben jetzt eine Webanwendung erstellt, die das Express-Framework nutzt. Im nächsten Abschnitt ändern wir die Anwendung so, dass sie einen Amazon Relational Database Service (RDS) zum Speichern eines Wanderprotokolls verwendet.

Aktualisieren der Anwendung, um Amazon RDS zu nutzen

In diesem nächsten Schritt aktualisieren wir die Anwendung, um Amazon RDS für MySQL zu verwenden.

So aktualisieren Sie die Anwendung zur Nutzung von RDS für MySQL

1. Um eine RDS-für-MySQL-Datenbank zu erstellen, die an Ihre Elastic-Beanstalk-Umgebung gekoppelt ist, folgen Sie den Anweisungen im Thema Hinzufügen einer Datenbank weiter unten in diesem Kapitel. Das Hinzufügen einer Datenbank-Instance dauert ca. 10 Minuten.

2. Aktualisieren Sie den Abschnitt Abhängigkeiten in der `package.json` mit dem folgenden Inhalt:

```
"dependencies": {
    "async": "^3.2.4",
    "express": "4.18.2",
    "jade": "1.11.0",
    "mysql": "2.18.1",
    "node-uuid": "^1.4.8",
    "body-parser": "^1.20.1",
    "method-override": "^3.0.0",
    "morgan": "^1.10.0",
    "errorhandler": "^1.5.1"
  }
```

3. Führen Sie npm install.

```
~/nodejs-example-express-rds$ npm install
```

4. Aktualisieren Sie `app.js`, um eine Verbindung zur Datenbank herzustellen, eine Tabelle zu erstellen und ein einzelnes Standard-Wanderprotokoll einzufügen. Jedes Mal, wenn diese Anwendung bereitgestellt wird, wird die Tabelle der vorherigen Wanderungen gelöscht und neu erstellt.

```
/**
 * Module dependencies.
 */

 const express = require('express')
 , routes = require('./routes')
 , hike = require('./routes/hike')
 , http = require('http')
 , path = require('path')
 , mysql = require('mysql')
 , async = require('async')
 , bodyParser = require('body-parser')
 , methodOverride = require('method-override')
 , morgan = require('morgan')
 , errorhandler = require('errorhandler');

const { connect } = require('http2');

const app = express()
```

```
app.set('views', __dirname + '/views')
app.set('view engine', 'jade')
app.use(methodOverride())
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({ extended: true }))
app.use(express.static(path.join(__dirname, 'public')))


app.set('connection', mysql.createConnection({
host: process.env.RDS_HOSTNAME,
user: process.env.RDS_USERNAME,
password: process.env.RDS_PASSWORD,
port: process.env.RDS_PORT}));

function init() {
 app.get('/', routes.index);
 app.get('/hikes', hike.index);
 app.post('/add_hike', hike.add_hike);
}

const client = app.get('connection');
async.series([
 function connect(callback) {
   client.connect(callback);
   console.log('Connected!');
 },
 function clear(callback) {
   client.query('DROP DATABASE IF EXISTS mynode_db', callback);
 },
 function create_db(callback) {
   client.query('CREATE DATABASE mynode_db', callback);
 },
 function use_db(callback) {
   client.query('USE mynode_db', callback);
 },
 function create_table(callback) {
    client.query('CREATE TABLE HIKES (' +
                        'ID VARCHAR(40), ' +
                        'HIKE_DATE DATE, ' +
                        'NAME VARCHAR(40), ' +
                        'DISTANCE VARCHAR(40), ' +
                        'LOCATION VARCHAR(40), ' +
                        'WEATHER VARCHAR(40), ' +
```

```
                        'PRIMARY KEY(ID))', callback);
 },
 function insert_default(callback) {
   const hike = {HIKE_DATE: new Date(), NAME: 'Hazard Stevens',
         LOCATION: 'Mt Rainier', DISTANCE: '4,027m vertical', WEATHER:'Bad', ID:
 '12345'};
   client.query('INSERT INTO HIKES set ?', hike, callback);
 }
], function (err, results) {
 if (err) {
   console.log('Exception initializing database.');
   throw err;
 } else {
   console.log('Database initialization complete.');
   init();
 }
});

module.exports = app
```

5. Fügen Sie `routes/hike.js` den folgenden Inhalt hinzu. Dadurch können die Routen neue Wanderprotokolle in die HIKES-Datenbank einfügen.

```
const uuid = require('node-uuid');
exports.index = function(req, res) {
  res.app.get('connection').query( 'SELECT * FROM HIKES', function(err,
rows) {
    if (err) {
      res.send(err);
    } else {
      console.log(JSON.stringify(rows));
      res.render('hike', {title: 'My Hiking Log', hikes: rows});
  }});
};
exports.add_hike = function(req, res){
  const input = req.body.hike;
  const hike = { HIKE_DATE: new Date(), ID: uuid.v4(), NAME: input.NAME,
  LOCATION: input.LOCATION, DISTANCE: input.DISTANCE, WEATHER: input.WEATHER};
  console.log('Request to log hike:' + JSON.stringify(hike));
  req.app.get('connection').query('INSERT INTO HIKES set ?', hike, function(err) {
      if (err) {
        res.send(err);
      } else {
```

```
        res.redirect('/hikes');
      }
   });
};
```

6. Ersetzen Sie den Inhalt von `routes/index.js` durch Folgendes:

```
/*
 * GET home page.
 */

exports.index = function(req, res){
  res.render('index', { title: 'Express' });
};
```

7. Fügen Sie die folgende Jade-Vorlage zu `views/hike.jade` hinzu, um die Benutzeroberfläche für das Hinzufügen von Wanderprotokollen bereitzustellen.

```
extends layout

block content
  h1= title
  p Welcome to #{title}

  form(action="/add_hike", method="post")
    table(border="1")
      tr
        td Your Name
        td
          input(name="hike[NAME]", type="textbox")
      tr
        td Location
        td
          input(name="hike[LOCATION]", type="textbox")
      tr
        td Distance
        td
          input(name="hike[DISTANCE]", type="textbox")
      tr
        td Weather
        td
          input(name="hike[WEATHER]", type="radio", value="Good")
          | Good
```

```
          input(name="hike[WEATHER]", type="radio", value="Bad")
          | Bad
          input(name="hike[WEATHER]", type="radio", value="Seattle", checked)
          | Seattle
      tr
        td(colspan="2")
          input(type="submit", value="Record Hike")

  div
    h3 Hikes
    table(border="1")
      tr
        td Date
        td Name
        td Location
        td Distance
        td Weather
      each hike in hikes
        tr
          td #{hike.HIKE_DATE.toDateString()}
          td #{hike.NAME}
          td #{hike.LOCATION}
          td #{hike.DISTANCE}
          td #{hike.WEATHER}
```

8. Stellen Sie die Änderungen mit dem eb deploy-Befehl bereit.

```
~/nodejs-example-express-rds$ eb deploy
```

Bereinigen

Falls Sie nicht länger mit Elastic Beanstalk arbeiten möchten, können Sie Ihre Umgebung beenden.

Mit dem eb terminate-Befehl können Sie Umgebung und alle darin enthaltenen Ressourcen beenden.

```
~/nodejs-example-express-rds$ eb terminate
The environment "nodejs-example-express-rds-env" and all associated instances will be
 terminated.
To confirm, type the environment name: nodejs-example-express-rds-env
INFO: terminateEnvironment is starting.
...
```

Bereitstellung einer Node.js Express-Anwendung mit Clustering auf Elastic Beanstalk

Dieses Tutorial führt Sie durch die Bereitstellung einer Beispielanwendung in Elastic Beanstalk mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) und die anschließende Aktualisierung der Anwendung für die Verwendung des Express-Frameworks, Amazon ElastiCache und Clustering. Mit Clustering lassen sich hohe Verfügbarkeit, Leistung und Sicherheit Ihrer Webanwendung verbessern. Weitere Informationen zu Amazon ElastiCache finden Sie unter Was ist Amazon ElastiCache (Memcached)? im Amazon ElastiCache (Memcached) -Benutzerhandbuch.

> **Note**
>
> In diesem Beispiel werden AWS Ressourcen erstellt, für die Ihnen möglicherweise Gebühren berechnet werden. Weitere Informationen zur AWS Preisgestaltung finden Sie unterhttps://aws.amazon.com/pricing/. Einige Dienste sind Teil des AWS kostenlosen Nutzungskontingents. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter https://aws.amazon.com/free/.

Voraussetzungen

Für dieses Tutorial sind die folgenden Voraussetzungen erforderlich:

- Die Node.js-Laufzeiten
- Die Standard-Paketmanager-Software von Node.js, npm
- Der Express-Befehlszeilengenerator
- Die Elastic Beanstalk Command Line Interface (EB CLI)

Weitere Informationen zur Installation dieser ersten drei Komponenten und zur Einrichtung Ihrer lokalen Entwicklungsumgebung finden Sie unter Einrichtung Ihrer Node.js Entwicklungsumgebung für Elastic Beanstalk. Für dieses Tutorial müssen Sie das AWS SDK für Node.js nicht installieren, das auch im referenzierten Thema erwähnt wird.

Detaillierte Anweisungen zum Installieren und Konfigurieren der EB-CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Erstellen einer Elastic Beanstalk-Umgebung

Ihr Anwendungsverzeichnis

In diesem Tutorial wird ein Verzeichnis verwendet, das als `nodejs-example-express-elasticache` für das Anwendungsquellpaket bezeichnet wird. Erstellen Sie das `nodejs-example-express-elasticache`-Verzeichnis für dieses Tutorial.

```
~$ mkdir nodejs-example-express-elasticache
```

> **Note**
>
> Jedes Tutorial in diesem Kapitel verwendet ein eigenes Verzeichnis für das Anwendungsquellpaket. Der Verzeichnisname entspricht dem Namen der Beispielanwendung, die im Tutorial verwendet wird.

Ändern Sie Ihr aktuelles Arbeitsverzeichnis zu `nodejs-example-express-elasticache`.

```
~$ cd nodejs-example-express-elasticache
```

Jetzt können wir eine Elastic-Beanstalk-Umgebung mit der Node.js und der Beispielanwendung einrichten. Wir werden die Elastic-Beanstalk-Befehlszeilenschnittstelle (EB-CLI) verwenden.

So konfigurieren Sie ein EB-CLI-Repository für Ihre Anwendung und erstellen eine Elastic-Beanstalk-Umgebung mit der Node.js-Plattform

1. Erstellen Sie ein Repository mit dem Befehl eb init.

   ```
   ~/nodejs-example-express-elasticache$ eb init --platform node.js --region <region>
   ```

   Mit diesem Befehl wird eine Konfigurationsdatei in einem Ordner mit dem Namen `.elasticbeanstalk` generiert. Sie gibt die Einstellungen für die Erstellung von Umgebungen für Ihre Anwendung an und erstellt eine Elastic Beanstalk-Anwendung mit dem Namen des aktuellen Ordners.
2. Erstellen Sie mit dem Befehl eb create eine Umgebung mit einer laufenden Beispielanwendung.

```
~/nodejs-example-express-elasticache$ eb create --sample nodejs-example-express-elasticache
```

Mit diesem Befehl wird eine Umgebung mit Lastausgleich, den Standardeinstellungen für die Node.js-Plattform und den folgenden Ressourcen erstellt:

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.
- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.

- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

  > **Domain-Sicherheit**
  >
  > Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
  > Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

3. Wenn die Umgebungserstellung abgeschlossen ist, verwenden Sie den Befehl eb open, um die URL der Umgebung im Standardbrowser zu öffnen.

```
~/nodejs-example-express-elasticache$ eb open
```

Sie haben jetzt eine Node.js Elastic-Beanstalk-Umgebung mit einer Beispielanwendung erstellt. Sie können sie mit Ihrer eigenen Anwendung aktualisieren. Als nächstes aktualisieren wir die Beispielanwendung, um das Express-Framework zu verwenden.

Aktualisieren der Anwendung zur Nutzung von Express

Aktualisieren Sie die Beispielanwendung in der Elastic Beanstalk-Umgebung zur Verwendung des Express-Frameworks.

Sie können den endgültigen Quellcode von .zip herunterladen. nodejs-example-express-elasticache

So aktualisieren Sie die Anwendung zur Nutzung von Express

Nachdem Sie eine Umgebung mit einer Beispielanwendung erstellt haben, können Sie diese mit Ihrer eigenen Anwendung aktualisieren. In diesem Verfahren führen wir zunächst die express- und npm install-Befehle aus, um das Express-Framework in Ihrem Anwendungsverzeichnis einzurichten.

1. Führen Sie den Befehl `express` aus. Dadurch werden `package.json`, `app.js` und weitere Verzeichnisse generiert.

   ```
   ~/nodejs-example-express-elasticache$ express
   ```

   Wenn Sie gefragt werden, ob Sie fortfahren möchten, geben Sie **y** ein.

   > **Note**
   >
   > Wenn der express-Befehl nicht funktioniert, haben Sie den Express-Befehlszeilengenerator möglicherweise nicht installiert, wie im vorherigen Abschnitt mit den Voraussetzungen beschrieben. Oder die Verzeichnispfadeinstellung für Ihren lokalen Computer muss möglicherweise eingerichtet werden, um den express-Befehl auszuführen. Im Abschnitt Voraussetzungen finden Sie detaillierte Schritte zum Einrichten Ihrer Entwicklungsumgebung, sodass Sie mit diesem Tutorial fortfahren können.

2. Richten Sie lokale Abhängigkeiten ein.

   ```
   ~/nodejs-example-express-elasticache$ npm install
   ```

3. (Optional) Stellen Sie sicher, dass der Web-Anwendungs-Server startet.

   ```
   ~/nodejs-example-express-elasticache$ npm start
   ```

   Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

   ```
   > nodejs@0.0.0 start /home/local/user/node-express
   > node ./bin/www
   ```

   Der Server läuft standardmäßig über Port 3000. Führen Sie als Test `curl http://localhost:3000` in einem anderen Terminal aus oder öffnen Sie einen Browser auf dem lokalen Computer und geben Sie die URL `http://localhost:3000` ein.

   Drücken Sie Strg+C, um den Server zu stoppen.

4. Benennen Sie `nodejs-example-express-elasticache/app.js` um in `nodejs-example-express-elasticache/express-app.js`.

```
~/nodejs-example-express-elasticache$ mv app.js express-app.js
```

5. Aktualisieren Sie die Zeile `var app = express();` in `nodejs-example-express-elasticache/express-app.js` folgendermaßen:

```
var app = module.exports = express();
```

6. Erstellen Sie auf dem lokalen Computer eine Datei namens `nodejs-example-express-elasticache/app.js` mit dem folgenden Code.

```
/**
 * Module dependencies.
 */

 const express = require('express'),
 session = require('express-session'),
 bodyParser = require('body-parser'),
 methodOverride = require('method-override'),
 cookieParser = require('cookie-parser'),
 fs = require('fs'),
 filename = '/var/nodelist',
 app = express();

let MemcachedStore = require('connect-memcached')(session);

function setup(cacheNodes) {
 app.use(bodyParser.raw());
 app.use(methodOverride());
 if (cacheNodes.length > 0) {
   app.use(cookieParser());

   console.log('Using memcached store nodes:');
   console.log(cacheNodes);

   app.use(session({
     secret: 'your secret here',
     resave: false,
     saveUninitialized: false,
     store: new MemcachedStore({ 'hosts': cacheNodes })
   }));
 } else {
   console.log('Not using memcached store.');
```

```
    app.use(session({
      resave: false,
      saveUninitialized: false, secret: 'your secret here'
    }));
  }

  app.get('/', function (req, resp) {
    if (req.session.views) {
      req.session.views++
      resp.setHeader('Content-Type', 'text/html')
      resp.send(`You are session: ${req.session.id}. Views: ${req.session.views}`)
    } else {
      req.session.views = 1
      resp.send(`You are session: ${req.session.id}. No views yet, refresh the page!
`)
    }
  });

  if (!module.parent) {
    console.log('Running express without cluster. Listening on port %d',
  process.env.PORT || 5000)
    app.listen(process.env.PORT || 5000)
  }
}

console.log("Reading elastic cache configuration")
// Load elasticache configuration.
fs.readFile(filename, 'UTF8', function (err, data) {
  if (err) throw err;

  let cacheNodes = []
  if (data) {
    let lines = data.split('\n');
    for (let i = 0; i < lines.length; i++) {
      if (lines[i].length > 0) {
        cacheNodes.push(lines[i])
      }
    }
  }

  setup(cacheNodes)
});
```

```
module.exports = app;
```

7. Ersetzen Sie den Inhalt der Datei `nodejs-example-express-elasticache/bin/www` durch Folgendes:

```
#!/usr/bin/env node

/**
 * Module dependencies.
 */

const app = require('../app');
const cluster = require('cluster');
const debug = require('debug')('nodejs-example-express-elasticache:server');
const http = require('http');
const workers = {},
  count = require('os').cpus().length;

function spawn() {
  const worker = cluster.fork();
  workers[worker.pid] = worker;
  return worker;
}


/**
 * Get port from environment and store in Express.
 */

const port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

if (cluster.isMaster) {
  for (let i = 0; i < count; i++) {
    spawn();
  }

  // If a worker dies, log it to the console and start another worker.
  cluster.on('exit', function (worker, code, signal) {
    console.log('Worker ' + worker.process.pid + ' died.');
    cluster.fork();
  });
```

```
  // Log when a worker starts listening
  cluster.on('listening', function (worker, address) {
    console.log('Worker started with PID ' + worker.process.pid + '.');
  });

} else {
  /**
   * Create HTTP server.
   */

  let server = http.createServer(app);

  /**
   * Event listener for HTTP server "error" event.
   */

  function onError(error) {
    if (error.syscall !== 'listen') {
      throw error;
    }

    const bind = typeof port === 'string'
      ? 'Pipe ' + port
      : 'Port ' + port;

    // handle specific listen errors with friendly messages
    switch (error.code) {
      case 'EACCES':
        console.error(bind + ' requires elevated privileges');
        process.exit(1);
        break;
      case 'EADDRINUSE':
        console.error(bind + ' is already in use');
        process.exit(1);
        break;
      default:
        throw error;
    }
  }

  /**
   * Event listener for HTTP server "listening" event.
   */
```

```
  function onListening() {
    const addr = server.address();
    const bind = typeof addr === 'string'
      ? 'pipe ' + addr
      : 'port ' + addr.port;
    debug('Listening on ' + bind);
  }

  /**
   * Listen on provided port, on all network interfaces.
   */

  server.listen(port);
  server.on('error', onError);
  server.on('listening', onListening);
}

/**
 * Normalize a port into a number, string, or false.
 */

function normalizePort(val) {
  const port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }

  if (port >= 0) {
    // port number
    return port;
  }

  return false;
}
```

8. Stellen Sie die Änderungen in Ihrer Elastic-Beanstalk-Umgebung mit dem Befehl eb deploy bereit.

```
~/nodejs-example-express-elasticache$ eb deploy
```

9. Die Umgebung wird nach einigen Minuten aktualisiert. Sobald die Umgebung grün und einsatzbereit ist, aktualisieren Sie die URL, um die ordnungsgemäße Funktionsweise zu prüfen. Sie sollten eine Webseite mit dem Text „Welcome to Express“ (Willkommen bei Express) sehen.

Sie können auf die Protokolle Ihrer EC2 Instances zugreifen, auf denen Ihre Anwendung ausgeführt wird. Anleitung zum Zugreifen auf Ihre Protokolle finden Sie unter Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen.

Als Nächstes aktualisieren wir die Express-Anwendung für die Verwendung von Amazon ElastiCache.

Um Ihre Express-Anwendung für die Nutzung von Amazon zu aktualisieren ElastiCache

1. Erstellen Sie auf dem lokalen Computer ein `.ebextensions`-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles. In diesem Beispiel verwenden wir `nodejs-example-express-elasticache/.ebextensions`.
2. Erstellen Sie eine Konfigurationsdatei `nodejs-example-express-elasticache/.ebextensions/elasticache-iam-with-script.config` mit dem folgenden Snippet. Weitere Informationen zur Konfigurationsdatei finden Sie unter Node.js-Konfigurations-Namespace. Damit wird ein IAM-Benutzer mit den benötigten Berechtigungen zum Erkennen der Elasticache-Knoten erstellt. Zudem wird jede Veränderung im Cache in einer Datei erfasst. Sie können die Datei auch aus einer nodejs-example-express-elasticacheZIP-Datei kopieren. Weitere Informationen zu den ElastiCache Eigenschaften finden Sie unterBeispiel: ElastiCache.

   > **Note**
   >
   > Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

```
Resources:
  MyCacheSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: "Lock cache down to webserver access only"
```

```
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort:
          Fn::GetOptionSetting:
            OptionName: CachePort
            DefaultValue: 11211
        ToPort:
          Fn::GetOptionSetting:
            OptionName: CachePort
            DefaultValue: 11211
        SourceSecurityGroupName:
          Ref: AWSEBSecurityGroup
MyElastiCache:
  Type: 'AWS::ElastiCache::CacheCluster'
  Properties:
    CacheNodeType:
      Fn::GetOptionSetting:
        OptionName: CacheNodeType
        DefaultValue: cache.t2.micro
    NumCacheNodes:
      Fn::GetOptionSetting:
        OptionName: NumCacheNodes
        DefaultValue: 1
    Engine:
      Fn::GetOptionSetting:
        OptionName: Engine
        DefaultValue: redis
    VpcSecurityGroupIds:
      -
        Fn::GetAtt:
          - MyCacheSecurityGroup
          - GroupId
AWSEBAutoScalingGroup :
  Metadata :
    ElastiCacheConfig :
      CacheName :
        Ref : MyElastiCache
      CacheSize :
         Fn::GetOptionSetting:
           OptionName : NumCacheNodes
           DefaultValue: 1
WebServerUser :
  Type : AWS::IAM::User
  Properties :
```

```
      Path : "/"
      Policies:
        -
          PolicyName: root
          PolicyDocument :
            Statement :
              -
                Effect : Allow
                Action :
                  - cloudformation:DescribeStackResource
                  - cloudformation:ListStackResources
                  - elasticache:DescribeCacheClusters
                Resource : "*"
  WebServerKeys :
    Type : AWS::IAM::AccessKey
    Properties :
      UserName :
        Ref: WebServerUser

Outputs:
  WebsiteURL:
    Description: sample output only here to show inline string function parsing
    Value: |
      http://`{ "Fn::GetAtt" : [ "AWSEBLoadBalancer", "DNSName" ] }`
  MyElastiCacheName:
    Description: Name of the elasticache
    Value:
      Ref : MyElastiCache
  NumCacheNodes:
    Description: Number of cache nodes in MyElastiCache
    Value:
      Fn::GetOptionSetting:
        OptionName : NumCacheNodes
        DefaultValue: 1

files:
  "/etc/cfn/cfn-credentials" :
    content : |
      AWSAccessKeyId=`{ "Ref" : "WebServerKeys" }`
      AWSSecretKey=`{ "Fn::GetAtt" : ["WebServerKeys", "SecretAccessKey"] }`
    mode : "000400"
    owner : root
    group : root
```

```
  "/etc/cfn/get-cache-nodes" :
    content : |
      # Define environment variables for command line tools
      export AWS_ELASTICACHE_HOME="/home/ec2-user/elasticache/$(ls /home/ec2-user/
elasticache/)"
      export AWS_CLOUDFORMATION_HOME=/opt/aws/apitools/cfn
      export PATH=$AWS_CLOUDFORMATION_HOME/bin:$AWS_ELASTICACHE_HOME/bin:$PATH
      export AWS_CREDENTIAL_FILE=/etc/cfn/cfn-credentials
      export JAVA_HOME=/usr/lib/jvm/jre

      # Grab the Cache node names and configure the PHP page
      aws cloudformation list-stack-resources --stack `{ "Ref" :
 "AWS::StackName" }` --region `{ "Ref" : "AWS::Region" }` --output text | grep
 MyElastiCache | awk '{print $4}' | xargs -I {} aws elasticache describe-cache-
clusters --cache-cluster-id {} --region `{ "Ref" : "AWS::Region" }` --show-
cache-node-info --output text | grep '^ENDPOINT' | awk '{print $2 ":" $3}' >
 `{ "Fn::GetOptionSetting" : { "OptionName" : "NodeListPath", "DefaultValue" : "/
var/www/html/nodelist" } }`
    mode : "000500"
    owner : root
    group : root

  "/etc/cfn/hooks.d/cfn-cache-change.conf" :
    "content": |
      [cfn-cache-size-change]
      triggers=post.update
      path=Resources.AWSEBAutoScalingGroup.Metadata.ElastiCacheConfig
      action=/etc/cfn/get-cache-nodes
      runas=root

sources :
  "/home/ec2-user/elasticache" : "https://elasticache-downloads.s3.amazonaws.com/
AmazonElastiCacheCli-latest.zip"

commands:
  make-elasticache-executable:
    command: chmod -R ugo+x /home/ec2-user/elasticache/*/bin/*

packages :
  "yum" :
    "aws-apitools-cfn"  : []

container_commands:
  initial_cache_nodes:
```

```
    command: /etc/cfn/get-cache-nodes
```

3. Erstellen Sie auf Ihrem lokalen Computer eine Konfigurationsdatei `nodejs-example-express-elasticache/.ebextensions/elasticache_settings.config` mit dem folgenden Codeausschnitt zur Konfiguration. ElastiCache

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     CacheNodeType: cache.t2.micro
     NumCacheNodes: 1
     Engine: memcached
     NodeListPath: /var/nodelist
```

4. Ersetzen Sie auf Ihrem lokalen Computer `nodejs-example-express-elasticache/express-app.js` durch den folgenden Snippet. Diese Datei liest die Knoten-Liste vom Datenträger (`/var/nodelist`) und konfiguriert Express so, dass `memcached` als Sitzungsspeicher verwendet wird, wenn Knoten vorhanden sind. Ihre Datei sollte wie folgt aussehen.

```
/**
 * Module dependencies.
 */

var express = require('express'),
    session = require('express-session'),
    bodyParser = require('body-parser'),
    methodOverride = require('method-override'),
    cookieParser = require('cookie-parser'),
    fs = require('fs'),
    filename = '/var/nodelist',
    app = module.exports = express();

var MemcachedStore = require('connect-memcached')(session);

function setup(cacheNodes) {
  app.use(bodyParser.raw());
  app.use(methodOverride());
  if (cacheNodes) {
      app.use(cookieParser());

      console.log('Using memcached store nodes:');
      console.log(cacheNodes);
```

```
      app.use(session({
          secret: 'your secret here',
          resave: false,
          saveUninitialized: false,
          store: new MemcachedStore({'hosts': cacheNodes})
      }));
  } else {
    console.log('Not using memcached store.');
    app.use(cookieParser('your secret here'));
    app.use(session());
  }

  app.get('/', function(req, resp){
  if (req.session.views) {
      req.session.views++
      resp.setHeader('Content-Type', 'text/html')
      resp.write('Views: ' + req.session.views)
      resp.end()
   } else {
      req.session.views = 1
      resp.end('Refresh the page!')
    }
  });

  if (!module.parent) {
      console.log('Running express without cluster.');
      app.listen(process.env.PORT || 5000);
  }
}

// Load elasticache configuration.
fs.readFile(filename, 'UTF8', function(err, data) {
    if (err) throw err;

    var cacheNodes = [];
    if (data) {
        var lines = data.split('\n');
        for (var i = 0 ; i < lines.length ; i++) {
            if (lines[i].length > 0) {
                cacheNodes.push(lines[i]);
            }
        }
    }
```

```
    setup(cacheNodes);
});
```

5. Aktualisieren Sie `package.json` auf dem lokalen Computer mit den folgenden Inhalten:

```
  "dependencies": {
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "express": "~4.16.1",
    "http-errors": "~1.6.3",
    "jade": "~1.11.0",
    "morgan": "~1.9.1",
    "connect-memcached": "*",
    "express-session": "*",
    "body-parser": "*",
    "method-override": "*"
  }
```

6. Führen Sie npm install.

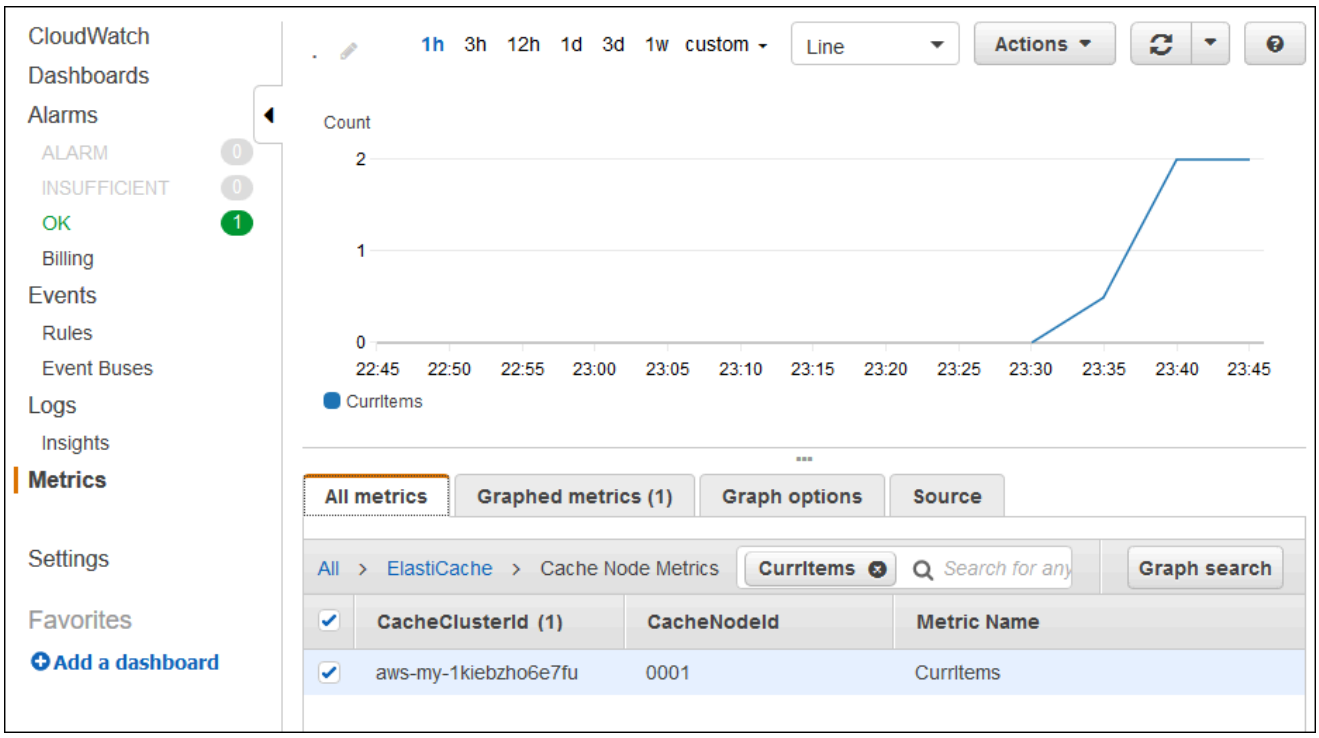
```
~/nodejs-example-express-elasticache$ npm install
```

7. Stellen Sie die aktualisierte Anwendung bereit.

```
~/nodejs-example-express-elasticache$ eb deploy
```

8. Die Umgebung wird nach einigen Minuten aktualisiert. Wenn die Umgebung grün und einsatzbereit ist, überprüfen Sie, ob der Code funktioniert hat.

   a. Sehen Sie in der CloudWatch Amazon-Konsole nach, um Ihre ElastiCache Messwerte einzusehen. Um Ihre ElastiCache Metriken anzuzeigen, wählen Sie im linken Bereich Metriken aus und suchen Sie dann nach CurrItems. Wählen Sie ElastiCache > Cache-Knoten-Metriken und wählen Sie dann Ihren Cache-Knoten aus, um die Anzahl der Elemente im Cache anzuzeigen.



> **Note**
>
> Achten Sie darauf, dass Sie in der Region suchen, in der Sie die Anwendung bereitgestellt haben.

Wenn Sie Ihre Anwendungs-URL kopieren und in einen anderen Webbrowser einfügen und die Seite aktualisieren, sollte Ihre CurrItem Anzahl nach 5 Minuten steigen.

b. Erstellen Sie einen Snapshot Ihrer Protokolle. Weitere Informationen über das Abrufen von Protokollen finden Sie unter Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen.

c. Überprüfen Sie die Datei `/var/log/nodejs/nodejs.log` im Protokoll-Bundle. Die Ausgabe sollte wie folgt aussehen:

```
Using memcached store nodes:
[ 'aws-my-1oys9co8zt1uo.1iwtrn.0001.use1.cache.amazonaws.com:11211' ]
```

Bereinigen

Wenn Sie Ihre Anwendung nicht mehr ausführen möchten, können Sie Ihr System durch Beenden Ihrer Umgebung und Löschen Ihrer Anwendung bereinigen.

Beenden Sie die Umgebung mit dem Befehl `eb terminate`, und löschen Sie die Anwendung mit dem Befehl `eb delete`.

So beenden Sie Ihre Umgebung

Führen Sie in dem Verzeichnis, in dem Sie Ihr lokales Repository erstellt haben, `eb terminate` aus.

```
$ eb terminate
```

Dieser Vorgang kann einige Minuten dauern. Elastic Beanstalk zeigt eine Meldung an, sobald die Umgebung erfolgreich beendet wurde.

Bereitstellen einer Node.js-Anwendung mit DynamoDB in Elastic Beanstalk

Dieses Tutorial und die dazugehörige Beispielanwendung (nodejs-example-dynamo.zip) führen Sie durch den Prozess der Bereitstellung einer Node.js -Anwendung, die das AWS SDK für JavaScript in Node.js verwendet, um mit dem Amazon DynamoDB-Service zu interagieren. Sie erstellen eine DynamoDB-Tabelle, die sich in einer Datenbank befindet, die von der Umgebung entkoppelt oder extern ist. AWS Elastic Beanstalk Sie werden die Anwendung auch so konfigurieren, dass sie eine entkoppelte Datenbank verwendet. In einer Produktionsumgebung empfiehlt es sich, eine Datenbank zu verwenden, die von der Elastic-Beanstalk-Umgebung entkoppelt ist, sodass sie unabhängig vom Lebenszyklus der Umgebung ist. Diese Vorgehensweise ermöglicht es Ihnen auch, Blau/Grün-Bereitstellungen durchzuführen.

Die Beispielanwendung veranschaulicht das Folgende:

- Eine DynamoDB-Tabelle zum Speichern der vom Benutzer bereitgestellten Textdaten.
- Die Konfigurationsdateien zum Erstellen der Tabelle.
- Ein Amazon-Simple-Notification-Service-Thema.
- Während der Bereitstellung Pakete mit der package.json-Datei installieren.

Sections

- Voraussetzungen
- Erstellen einer Elastic Beanstalk-Umgebung
- Hinzufügen von Berechtigungen zu den Instances Ihrer Umgebung
- Bereitstellen der Beispielanwendung
- Erstellen einer DynamoDB-Tabelle
- Aktualisieren der Konfigurationsdateien der Anwendung
- Konfigurieren der Umgebung für hohe Verfügbarkeit
- Bereinigen
- Nächste Schritte

Voraussetzungen

Für dieses Tutorial sind die folgenden Voraussetzungen erforderlich:

- Die Node.js-Laufzeiten
- Die Standard-Paketmanager-Software von Node.js, npm
- Der Express-Befehlszeilengenerator
- Die Elastic Beanstalk Command Line Interface (EB CLI)

Weitere Informationen zur Installation dieser ersten drei Komponenten und zur Einrichtung Ihrer lokalen Entwicklungsumgebung finden Sie unter Einrichtung Ihrer Node.js Entwicklungsumgebung für Elastic Beanstalk. Für dieses Tutorial müssen Sie das AWS SDK für Node.js nicht installieren, das auch im referenzierten Thema erwähnt wird.

Detaillierte Anweisungen zum Installieren und Konfigurieren der EB-CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Erstellen einer Elastic Beanstalk-Umgebung

Ihr Anwendungsverzeichnis

In diesem Tutorial wird ein Verzeichnis verwendet, das als `nodejs-example-dynamo` für das Anwendungsquellpaket bezeichnet wird. Erstellen Sie das `nodejs-example-dynamo`-Verzeichnis für dieses Tutorial.

```
~$ mkdir nodejs-example-dynamo
```

> **Note**
>
> Jedes Tutorial in diesem Kapitel verwendet ein eigenes Verzeichnis für das Anwendungsquellpaket. Der Verzeichnisname entspricht dem Namen der Beispielanwendung, die im Tutorial verwendet wird.

Ändern Sie Ihr aktuelles Arbeitsverzeichnis zu `nodejs-example-dynamo`.

```
~$ cd nodejs-example-dynamo
```

Jetzt können wir eine Elastic-Beanstalk-Umgebung mit der Node.js und der Beispielanwendung einrichten. Wir werden die Elastic-Beanstalk-Befehlszeilenschnittstelle (EB-CLI) verwenden.

So konfigurieren Sie ein EB-CLI-Repository für Ihre Anwendung und erstellen eine Elastic-Beanstalk-Umgebung mit der Node.js-Plattform

1. Erstellen Sie ein Repository mit dem Befehl eb init.

   ```
   ~/nodejs-example-dynamo$ eb init --platform node.js --region <region>
   ```

   Mit diesem Befehl wird eine Konfigurationsdatei in einem Ordner mit dem Namen `.elasticbeanstalk` generiert. Sie gibt die Einstellungen für die Erstellung von Umgebungen für Ihre Anwendung an und erstellt eine Elastic Beanstalk-Anwendung mit dem Namen des aktuellen Ordners.
2. Erstellen Sie mit dem Befehl eb create eine Umgebung mit einer laufenden Beispielanwendung.

   ```
   ~/nodejs-example-dynamo$ eb create --sample nodejs-example-dynamo
   ```

   Mit diesem Befehl wird eine Umgebung mit Lastausgleich, den Standardeinstellungen für die Node.js-Plattform und den folgenden Ressourcen erstellt:

   - EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.

> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

3. Wenn die Umgebungserstellung abgeschlossen ist, verwenden Sie den Befehl eb open, um die URL der Umgebung im Standardbrowser zu öffnen.

```
~/nodejs-example-dynamo$ eb open
```

Sie haben jetzt eine Node.js Elastic-Beanstalk-Umgebung mit einer Beispielanwendung erstellt. Sie können sie mit Ihrer eigenen Anwendung aktualisieren. Als nächstes aktualisieren wir die Beispielanwendung, um das Express-Framework zu verwenden.

Hinzufügen von Berechtigungen zu den Instances Ihrer Umgebung

Ihre Anwendung läuft auf einer oder mehreren EC2 Instances hinter einem Load Balancer und bedient HTTP-Anfragen aus dem Internet. Wenn sie eine Anfrage erhält, die die Nutzung von AWS Diensten erfordert, verwendet die Anwendung die Berechtigungen der Instance, auf der sie ausgeführt wird, um auf diese Dienste zuzugreifen.

Die Beispielanwendung verwendet Instanzberechtigungen, um Daten in eine DynamoDB-Tabelle zu schreiben und Benachrichtigungen an ein Amazon SNS SNS-Thema mit dem SDK für JavaScript in Node.js zu senden. Fügen Sie dem Standard-Instance-Profil die folgenden verwalteten Richtlinien hinzu, um den EC2 Instances in Ihrer Umgebung Zugriff auf DynamoDB und Amazon SNS zu gewähren:

- AmazonDynamoDBFullAccess
- SNSFullZugriff auf Amazon

So fügen Sie Richtlinien zum Instance-Standardprofil hinzu

1. Öffnen Sie die Seite Roles (Rollen) in der IAM-Konsole.
2. Wählen Sie aws-elasticbeanstalk-ec2 Rollen.

3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Attach policies (Richtlinien anfügen) aus.
4. Wählen Sie die verwaltete Richtlinie für die zusätzlichen Services aus, die von der Anwendung verwendet werden. Wählen Sie in diesem Tutorial `AmazonSNSFullAccess` und `AmazonDynamoDBFullAccess`.
5. Wählen Sie Richtlinie anfügen aus.

Weitere Informationen zum Verwalten von Instance-Profilen finden Sie unter Elastic Beanstalk Instance-Profile verwalten.

Bereitstellen der Beispielanwendung

Jetzt ist Ihre Umgebung bereit, damit Sie die Beispielanwendung für dieses Tutorial bereitstellen und ausführen können: nodejs-example-dynamo.zip.

So stellen Sie die Tutorial-Beispielanwendung bereit und führen sie aus

1. Wechseln Sie von Ihrem aktuellen Arbeitsverzeichnis in das Installationsverzeichnis `nodejs-example-dynamo`.

   ```
   ~$ cd nodejs-example-dynamo
   ```

2. Laden Sie den Inhalt des Quellpakets für die Beispielanwendung (nodejs-example-dynamo.zip) herunter und extrahieren Sie ihn in das Anwendungsverzeichnis. `nodejs-example-dynamo`
3. Stellen Sie die Anwendung in Ihrer Elastic-Beanstalk-Umgebung mit dem Befehl eb deploy bereit.

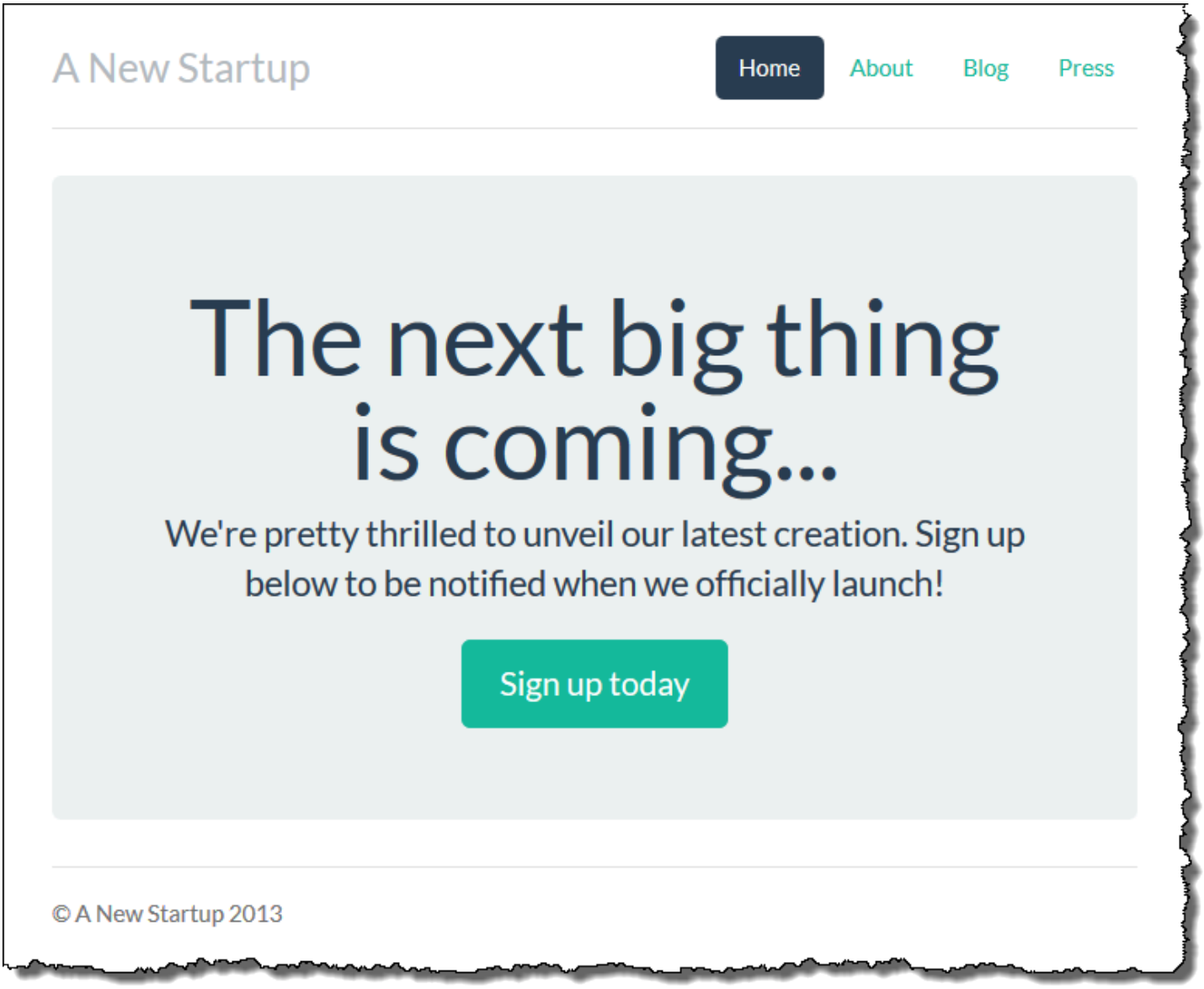
   ```
   ~/nodejs-example-dynamo$ eb deploy
   ```

   > **Note**
   >
   > Standardmäßig erstellt der `eb deploy` Befehl eine ZIP-Datei Ihres Projektordners. Sie können die EB-CLI so konfigurieren, dass sie ein Artefakt aus Ihrem Erstellungsprozess bereitstellt, anstatt eine ZIP-Datei Ihres Projektordners zu erstellen. Weitere Informationen finden Sie unter Bereitstellen eines Artefakts anstelle des Projektordners.

4. Wenn die Umgebungserstellung abgeschlossen ist, verwenden Sie den Befehl eb open, um die URL der Umgebung im Standardbrowser zu öffnen.

```
~/nodejs-example-dynamo$ eb open
```

Die Website sammelt Benutzerkontaktinformationen und speichert die Daten in einer DynamoDB-Tabelle. Zum Hinzufügen eines Eintrags wählen Sie Sign up today (Jetzt registrieren), geben Sie einen Namen und eine E-Mail-Adresse ein, und wählen Sie dann Sign Up! (Registrieren) aus. Die Web-App schreibt die Formularinhalte in die Tabelle und löst eine E-Mail-Benachrichtigung von Amazon SNS aus.



Aktuell ist das Amazon SNS-Thema mit einer Platzhalter-E-Mail für Benachrichtigungen konfiguriert. Sie werden die Konfiguration in Kürze aktualisieren, aber in der Zwischenzeit können Sie die DynamoDB-Tabelle und das Amazon SNS-Thema in der AWS-Managementkonsoleüberprüfen.

So zeigen Sie die Tabelle an

1. Öffnen Sie die Seite Tables (Tabellen) in der DynamoDB-Konsole.
2. Suchen Sie die Tabelle, die die Anwendung erstellt hat. Der Name beginnt mit awseb und enthält. StartupSignupsTable
3. Wählen Sie die Tabelle aus, klicken Sie auf Items (Elemente) und anschließend auf Start search (Suche starten), um alle Elemente in der Tabelle anzuzeigen.

Die Tabelle enthält einen Eintrag für jede E-Mail-Adresse, die auf der Anmeldungs-Website übermittelt wurde. Zusätzlich zum Schreiben in die Tabelle sendet die Anwendung eine Nachricht an ein Amazon SNS-Thema, das zwei Abonnements hat: eines für E-Mail-Benachrichtigungen an Sie und ein anderes für eine Amazon Simple Queue Service-Warteschlange, aus der eine Worker-Anwendung lesen kann, um Anfragen zu verarbeiten und E-Mails an interessierte Kunden zu senden.

So zeigen Sie das Thema an

1. Öffnen Sie die Seite Topics (Themen) in der Amazon SNS-Konsole.
2. Suchen Sie das Thema, das die Anwendung erstellt hat. Der Name beginnt mit awseb und enthält. NewSignupTopic
3. Wählen Sie das Thema, um seine Abonnements anzuzeigen.

Die Anwendung (`app.js`) definiert zwei Routen. Der Root-Pfad (`/`) gibt eine Webseite zurück, die anhand einer Embedded-Vorlage JavaScript (EJS) mit einem Formular gerendert wurde, das der Benutzer ausfüllt, um seinen Namen und seine E-Mail-Adresse zu registrieren. Durch das Übermitteln des Formulars wird eine POST-Anforderung mit den Formulardaten an das `/signup`-Routing gesendet. Es schreibt einen Eintrag in die DynamoDB-Tabelle und veröffentlicht eine Nachricht im Amazon SNS-Thema, um den Eigentümer des Themas zu benachrichtigen.

Die Beispielanwendung enthält Konfigurationsdateien, mit der die DynamoDB-Tabelle, das Amazon SNS-Thema und die Amazon SQS-Warteschlange der Anwendung erstellt werden. Dadurch können Sie eine neue Umgebung erstellen und die Funktionalität sofort testen. Der Nachteil ist, dass die DynamoDB-Tabelle an die Umgebung gebunden ist. Bei einer Produktionsumgebung sollten Sie die

DynamoDB-Tabelle außerhalb Ihrer Umgebung erstellen. Auf diese Weise vermeiden Sie, dass Sie sie verlieren, wenn Sie die Umgebung beenden oder ihre Konfiguration aktualisieren.

Erstellen einer DynamoDB-Tabelle

Um eine externe DynamoDB-Tabelle mit einer Anwendung zu verwenden, die in Elastic Beanstalk ausgeführt wird, erstellen Sie zuerst eine Tabelle in DynamoDB. Wenn Sie eine Tabelle außerhalb von Elastic Beanstalk erstellen, ist sie völlig unabhängig von Elastic Beanstalk und Ihren Elastic Beanstalk-Umgebungen und wird von Elastic Beanstalk nicht beendet.

Erstellen Sie die Tabelle mit folgenden Einstellungen:

- Table name (Name der Tabelle) – **`nodejs-tutorial`**
- Primary key (Primärschlüssel) – **`email`**
- Primary key type (Primärschlüsseltyp) – Zeichenfolge

So erstellen Sie eine DynamoDB-Tabelle

1. Öffnen Sie die Seite Tables (Tabellen)in der DynamoDB Management-Konsole.
2. Wählen Sie Create table (Tabelle erstellen) aus.
3. Geben Sie einen Table name (Name der Tabelle) und einen Primary key (Primärschlüssel) ein.
4. Wählen Sie den Typ des Primärschlüssels aus.
5. Wählen Sie Erstellen aus.

Aktualisieren der Konfigurationsdateien der Anwendung

Aktualisieren Sie die Konfigurationsdateien in der Anwendungsquelle zur Verwendung der nodejs-tutorial-Tabelle, anstatt eine neue Tabelle zu erstellen.

So aktualisieren Sie die Beispielanwendung für die Verwendung in der Produktion

1. Wechseln Sie von Ihrem aktuellen Arbeitsverzeichnis in das Installationsverzeichnis `nodejs-example-dynamo`.

   ```
   ~$ cd nodejs-example-dynamo
   ```

2. Öffnen Sie `.ebextensions/options.config` und ändern Sie die Werte der folgenden Einstellungen:

- NewSignupEmail— Ihre E-Mail-Adresse.
- STARTUP_SIGNUP_TABLE – nodejs-tutorial

Example.ebextensions/options.config

```
option_settings:
  aws:elasticbeanstalk:customoption:
    NewSignupEmail: you@example.com
  aws:elasticbeanstalk:application:environment:
    THEME: "flatly"
    AWS_REGION: '`{"Ref" : "AWS::Region"}`'
    STARTUP_SIGNUP_TABLE: nodejs-tutorial
    NEW_SIGNUP_TOPIC: '`{"Ref" : "NewSignupTopic"}`'
  aws:elasticbeanstalk:container:nodejs:
    ProxyServer: nginx
  aws:elasticbeanstalk:container:nodejs:staticfiles:
    /static: /static
  aws:autoscaling:asg:
    Cooldown: "120"
  aws:autoscaling:trigger:
    Unit: "Percent"
    Period: "1"
    BreachDuration: "2"
    UpperThreshold: "75"
    LowerThreshold: "30"
    MeasureName: "CPUUtilization"
```

Dies gilt für die folgenden Konfigurationen für die Anwendung:

- Die E-Mail-Adresse, die das Amazon-SNS-Thema für Benachrichtigungen verwendet, ist auf Ihre Adresse gesetzt oder auf die, die Sie in die `options.config`-Datei eingegeben.
- Die Tabelle nodejs-tutorial wird anstelle der Tabelle verwendet, die von `.ebextensions/create-dynamodb-table.config` erstellt wurde.

3. Entfernen Sie `.ebextensions/create-dynamodb-table.config`.

```
~/nodejs-tutorial$ rm .ebextensions/create-dynamodb-table.config
```

Wenn Sie die Anwendung das nächste Mal bereitstellen, wird die Tabelle, die von dieser Konfigurationsdatei erstellt wurde, gelöscht.

4. Stellen Sie die aktualisierte Anwendung in Ihrer Elastic-Beanstalk-Umgebung mit dem Befehl eb deploy bereit.

```
~/nodejs-example-dynamo$ eb deploy
```

5. Wenn die Umgebungserstellung abgeschlossen ist, verwenden Sie den Befehl eb open, um die URL der Umgebung im Standardbrowser zu öffnen.

```
~/nodejs-example-dynamo$ eb open
```

Bei der Bereitstellung aktualisiert Elastic Beanstalk die Konfiguration des Amazon SNS-Themas und löscht die DynamoDB-Tabelle, die erstellt wurde, als Sie die erste Version der Anwendung bereitgestellt haben.

Wenn Sie die Umgebung jetzt beenden, wird die nodejs-tutorial-Tabelle nicht gelöscht. Dadurch können Sie blaue/grüne Bereitstellungen durchführen, Konfigurationsdateien ändern oder Ihre Website offline nehmen, ohne Datenverlust zu riskieren.

Öffnen Sie Ihre Webseite in einem Browser und überprüfen Sie, ob das Formular wie erwartet funktioniert. Erstellen Sie einige Einträge und überprüfen Sie die Tabelle in der DynamoDB-Konsole.

So zeigen Sie die Tabelle an

1. Öffnen Sie die Seite Tables (Tabellen) in der DynamoDB-Konsole.
2. Suchen Sie die nodejs-tutorial-Tabelle.
3. Wählen Sie die Tabelle aus, klicken Sie auf Items (Elemente) und anschließend auf Start search (Suche starten), um alle Elemente in der Tabelle anzuzeigen.

Sie können auch sehen, dass Elastic Beanstalk die zuvor erstellte Tabelle gelöscht hat.

Konfigurieren der Umgebung für hohe Verfügbarkeit

Konfigurieren Sie abschließend für die Auto-Scaling-Gruppe der Umgebung für höhere Instance-Mindestanzahl. Führen Sie immer mindestens zwei Instances aus, damit einzelne Webserver in der

Umgebung keine Fehlerquelle darstellen und damit Sie Änderungen ohne Ausfallzeit der Website bereitstellen können.

So konfigurieren Sie die Auto-Scaling-Gruppe in der Umgebung für hohe Verfügbarkeit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Legen Sie im Abschnitt Auto Scaling Group (Gruppe der automatischen Skalierung) für die Option Min instances (Min. Instances) **2** fest.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung über die Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Sie können auch die durch Sie erstellten, externen DynamoDB-Tabellen löschen.

So löschen Sie eine DynamoDB-Tabelle

1. Öffnen Sie die Seite Tables (Tabellen) in der DynamoDB-Konsole.
2. Wählen Sie eine Tabelle aus.
3. Wählen Sie die Option Actions (Aktionen) und anschließend Delete table (Tabelle löschen) aus.
4. Wählen Sie Delete (Löschen) aus.

Nächste Schritte

Die Beispielanwendung verwendet Konfigurationsdateien, um Softwareeinstellungen zu konfigurieren und AWS Ressourcen als Teil Ihrer Umgebung zu erstellen. Weitere Informationen über Konfigurationsdateien und Ihre Verwendung finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

Die Beispielanwendung dieses Tutorials verwendet das Express-Web-Framework für Node.js. Weitere Informationen zu Express finden Sie in der offiziellen Dokumentation unter expressjs.com.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, konfigurieren Sie einen benutzerdefinierten Domainnamen für die Umgebung und aktivieren Sie HTTPS für sichere Verbindungen.

Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Node.js Elastic Beanstalk Beanstalk-Umgebung

Dieses Thema enthält Anweisungen zum Erstellen eines Amazon RDS mit der Elastic-Beanstalk-Konsole. Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann an Ihre Umgebung gekoppelt sein und von Elastic Beanstalk verwaltet werden oder sie kann als entkoppelte Datenbank erstellt und extern von einem anderen Service verwaltet werden. In dieser Anleitung wird die Datenbank an Ihre Umgebung gekoppelt und von Elastic Beanstalk verwaltet. Weitere Informationen zur Integration eines Amazon RDS mit Elastic Beanstalk finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Sections

- Hinzufügen einer DB-Instance zu Ihrer Umgebung
- Herunterladen eines Treibers

- Herstellen einer Verbindung zu einer Datenbank

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftenwert
`RDS_HOSTNAME`	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
`RDS_PORT`	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
`RDS_DB_NAME`	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration)

Eigenschaftenname	Beschreibung	Eigenschaftenwert
		der Amazon RDS-Konsole: DB-Name.
`RDS_USERNAME`	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
`RDS_PASSWORD`	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.

Weitere Informationen zur Konfiguration einer Datenbank-Instance in Verbindung mit einer Elastic-Beanstalk-Umgebung finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Herunterladen eines Treibers

Fügen Sie den Datenbanktreiber zur `package.json`-Datei unter `dependencies` hinzu.

Example **`package.json`** – Express mit MySQL

```
{
  "name": "my-app",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "ejs": "latest",
    "aws-sdk": "latest",
    "express": "latest",
    "body-parser": "latest",
    "mysql": "latest"
  },
  "scripts": {
    "start": "node app.js"
  }
}
```

Gängige Treiberpakete für Node.js

- MySQL – mysql
- PostgreSQL – node-postgress
- SQL Server – node-mssql
- Oracle – node-oracledb

Herstellen einer Verbindung zu einer Datenbank

Elastic Beanstalk stellt Verbindungsinformationen für zugehörige DB-Instances in Umgebungseigenschaften bereit. Mit `process.env.`*`VARIABLE`* können Sie die Eigenschaften lesen und eine Datenbankverbindung konfigurieren.

Example app.js – MySQL-Datenbankverbindung

```
var mysql = require('mysql');

var connection = mysql.createConnection({
  host     : process.env.RDS_HOSTNAME,
  user     : process.env.RDS_USERNAME,
  password : process.env.RDS_PASSWORD,
  port     : process.env.RDS_PORT
});

connection.connect(function(err) {
  if (err) {
    console.error('Database connection failed: ' + err.stack);
    return;
  }

  console.log('Connected to database.');
});

connection.end();
```

Weitere Informationen zum Konstruieren einer Verbindungszeichenfolge mithilfe von node-mysql finden Sie unter npmjs. org/package/mysql.

Tools und Ressourcen für Node.js

Es gibt zahlreiche Möglichkeiten, um zusätzliche Unterstützung bei der Entwicklung von Node.js-Anwendungen zu erhalten:

Ressource	Description
GitHub	Installieren Sie das AWS SDK für Node.js mit GitHub.
AWS SDK für Node.js (Entwickl ervorschau)	Umfangreiches Kit mit Beispiel-Code, Dokumentation, Tools und zusätzlichen Ressourcen.

Bereitstellung von PHP-Anwendungen mit Elastic Beanstalk

Mit der Elastic Beanstalk Command Line Interface (EB CLI) oder der Elastic Beanstalk-Konsole können Sie eine PHP-Anwendung in wenigen Minuten bereitstellen. Um zu erfahren, wie das geht, beginnen Sie mit dem QuickStart Tutorial und schauen Sie sich dann die Beispiele für Fortgeschrittene an.

Ressourcen für die Bereitstellung von PHP-Anwendungen auf Elastic Beanstalk

- QuickStart für PHP— Step-by-step Anweisungen zur Bereitstellung einer Hello World PHP-Anwendung mit der EB-CLI.
- Verwenden der Elastic Beanstalk PHP-Plattform— Wie man Plattformfunktionen speziell für PHP verwendet.
- Fortschrittliche -Beispiele— Anleitungen zu gängigen PHP-Frameworks und -Anwendungen sowie zum Hinzufügen einer Amazon RDS-Datenbank zu Ihrer Umgebung.

Weitere Informationen zur Entwicklung mit PHP in AWS finden Sie in den folgenden Ressourcen:

- GitHub— Installieren Sie das AWS SDK for PHP mit GitHub.
- PHP Developer Center — Tools, Dokumente und Beispielcode zur Entwicklung von PHP-Anwendungen AWS.
- AWS SDK for PHP FAQs — Erhalten Sie Antworten auf häufig gestellte Fragen.

QuickStart: Stellen Sie eine PHP-Anwendung auf Elastic Beanstalk bereit

Im folgenden Tutorial erfahren Sie, wie Sie mithilfe der EB CLI eine PHP-Beispielanwendung erstellen und in einer AWS Elastic Beanstalk Umgebung bereitstellen.

> ⚠ Nicht für den Produktionsgebrauch
>
> Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine PHP-Anwendung
- Schritt 2: Führen Sie Ihre Anwendung lokal aus
- Schritt 3: Initialisieren und implementieren Sie Ihre PHP-Anwendung
- Schritt 4: Durchsuchen Sie Ihre Cloud-Anwendung
- Schritt 5: Aktualisieren Sie Ihre Anwendung und stellen Sie sie erneut bereit
- Bereinigen
- Nächste Schritte

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen, um Elastic Beanstalk nutzen zu können.

Erstellen Sie ein Konto AWS

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center -Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

- Elastic Beanstalk Command Line Interface — Informationen zur Installation finden Sie unter. EB CLI mit Setup-Skript installieren (empfohlen)
- PHP — Installieren Sie PHP auf Ihrem lokalen Computer, indem Sie den Installations- und Konfigurationsanweisungen auf der PHP-Website folgen.

Schritt 1: Erstellen Sie eine PHP-Anwendung

Für diesen Schnellstart erstellen Sie eine Hello World PHP-Anwendung.

Erstellen Sie ein Projektverzeichnis.

```
~$ mkdir eb-php
~$ cd eb-php
```

Erstellen Sie als Nächstes eine `index.php` Datei im Projektverzeichnis und fügen Sie den folgenden Code hinzu.

Example **`index.php`**

```
<?php
  echo "Hello from a PHP application running in Elastic Beanstalk!";
?>
```

Schritt 2: Führen Sie Ihre Anwendung lokal aus

Verwenden Sie den folgenden Befehl, um Ihre Anwendung lokal auszuführen.

```
~$ php -S localhost:5000
```

Öffnen Sie einen Browser für http://localhost:5000.

Sie sollten Ihre Hallo-Nachricht im Browser sehen und die Protokollnachrichten in Ihrem Terminal.

Stoppen Sie den lokalen Server, indem Sie die Eingabe eingeben`Control+c`, damit Sie den Elastic Beanstalk bereitstellen können.

Schritt 3: Initialisieren und implementieren Sie Ihre PHP-Anwendung

Als Nächstes stellen Sie Ihre Anwendung mithilfe der Elastic Beanstalk Beanstalk-Konsole oder der EB-CLI in einer Umgebung bereit. In diesem Tutorial verwenden Sie die EB-CLI mit der interaktiven Option, um eine Umgebung zu initialisieren.

Um Ihre Umgebung zu initialisieren und eine Umgebung zu erstellen

1. Führen Sie den Befehl init aus.

   ```
   ~$ eb init -i
   ```

   Der Befehl init erstellt interaktiv eine Anwendung. Als Anwendungsname wird standardmäßig der lokale Ordner verwendet, d. h. `eb-php`

   Akzeptieren Sie für alle Eingabeaufforderungen, mit Ausnahme des SSH-Zugriffs, die Standardeinstellungen, um eine Umgebung mit der neuesten PHP-Plattformversion zu erstellen. Zur Behebung von Instances können Sie den SSH-Zugriff einrichten, indem Sie den `eb init -i` Befehl zu einem späteren Zeitpunkt erneut ausführen, oder Sie können eine Verbindung über Amazon EC2 Instance Connect oder Session Manager herstellen.
2. Erstellen Sie eine Umgebung und stellen Sie Ihre Anwendung bereit

   Führen Sie den folgenden Befehl aus, um eine Umgebung mit dem Namen zu erstellen`blue-env`.

   ```
   ~$ eb create blue-env
   ```

Wenn Sie den eb create Befehl zum ersten Mal ausführen, erstellt Elastic Beanstalk automatisch eine ZIP-Datei Ihrer Anwendung, ein sogenanntes Quellpaket. Als Nächstes erstellt Elastic Beanstalk eine Umgebung mit einer oder mehreren Amazon EC2 EC2-Instances und stellt die Anwendung dann in der Umgebung bereit.

Die Bereitstellung Ihrer Anwendung auf Elastic Beanstalk kann bis zu fünf Minuten dauern.

Schritt 4: Durchsuchen Sie Ihre Cloud-Anwendung

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, sollte Ihre Anwendung laufen und auf Anfragen auf Port 5000 warten. Stellen Sie mit dem folgenden Befehl eine Connect zu Ihrer Anwendung her:

```
~$ eb open
```

Der `eb open` Befehl öffnet einen Browser-Tab zu einer benutzerdefinierten Subdomain, die für Ihre Anwendung erstellt wurde.

Schritt 5: Aktualisieren Sie Ihre Anwendung und stellen Sie sie erneut bereit

Nachdem Sie eine Anwendung erstellt und in einer Umgebung bereitgestellt haben, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Die Bereitstellung einer neuen Anwendungsversion ist schneller, da dafür keine Bereitstellung oder ein Neustart von Amazon EC2 EC2-Instances erforderlich ist.

Aktualisieren Sie Ihren PHP-Code so, dass er den REQUEST_TIME-Wert aus der Serverumgebung enthält:

```
<?php
  echo "Hello from a PHP application running in Elastic Beanstalk!";

  $timestamp = $_SERVER['REQUEST_TIME'];
  echo '<br/>Request time: ' . date('Y/m/d H:i:s', $timestamp);
?>
```

Stellen Sie Ihren PHP-Code mit dem folgenden Befehl erneut auf Elastic Beanstalk bereit:

```
~$ eb deploy
```

Bei Ausführung von eb deploy werden die Inhalte des Projektverzeichnisses von der EB CLI gebündelt und in der Umgebung bereitgestellt.

Aktualisieren Sie nach Abschluss der Bereitstellung die Seite oder stellen Sie mit erneut eine Verbindung zu Ihrer Anwendung her. `eb open` Sie sollten Ihre Updates sehen. Falls nicht, können Sie das Problem beheben, indem Sie Ihren lokalen Server erneut ausführen, um Ihre Änderungen zu überprüfen.

> **Herzlichen Glückwunsch!**
>
> Sie haben eine PHP-Anwendung mit Elastic Beanstalk erstellt, bereitgestellt und aktualisiert!

Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie Amazon EC2 EC2-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung über die Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Alternativ können Sie Ihre Umgebung mit der EB-CLI mit dem folgenden Befehl beenden:

```
~$ eb terminate
```

Nächste Schritte

Sie können Ihre Anwendungsumgebung mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Weitere Informationen finden Sie unter Erkunden Sie Ihre Umgebung.

Beispiele für Fortgeschrittene mit PHP finden Sie unterFortgeschrittene Beispiele für PHP in Elastic Beanstalk.

Verwenden der Elastic Beanstalk PHP-Plattform

AWS Elastic Beanstalk bietet und unterstützt verschiedene Plattformzweige für verschiedene Versionen von PHP. Die Plattformen unterstützen PHP-Webanwendungen, die eigenständig oder unter Composer ausgeführt werden. Eine vollständige Liste der unterstützten AWS Elastic Beanstalk Plattformzweige finden Sie unter PHP im Dokument Plattformen.

Elastic Beanstalk bietet Konfigurationsoptionen, mit denen Sie die Software anpassen können, die auf den Amazon EC2 EC2-Instances in Ihrer Elastic Beanstalk Beanstalk-Umgebung ausgeführt wird. Sie können die von Ihrer Anwendung erforderlichen Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren, Ordner in der Anwendungsquelle mit statischen Dateien zu vom Proxy-Server verwendeten Pfaden zuordnen und gemeinsame PHP-Initialisierungseinstellungen festlegen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das Ändern der Konfiguration einer ausgeführten Umgebung verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie gespeicherte Konfigurationen verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie Konfigurationsdateien einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Wenn Sie Composer verwenden, können Sie durch das Einbinden einer `composer.json`-Datei in das Quell-Bundle die Pakete während der Bereitstellung installieren.

Für eine erweiterte PHP-Konfiguration und PHP-Einstellungen, die nicht als Konfigurationsoptionen verfügbar sind, können Sie mit Konfigurationsdateien eine `INI`-Datei bereitstellen, mit der die Standardeinstellungen von Elastic Beanstalk erweitert und überschrieben werden, oder Sie können zusätzliche Erweiterungen installieren.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, falls vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe Konfigurationsoptionen.

Details zu den verschiedenen Möglichkeiten für das Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter the section called “Erweitern von Linux-Plattformen”.

Themen zur PHP-Plattform

- Das AWS SDK for PHP installieren
- Überlegungen zu PHP 8.1 auf Amazon Linux 2
- Konfigurieren Ihrer PHP-Umgebung
- Namespaces für die Konfiguration
- Die Abhängigkeiten Ihrer Elastic Beanstalk PHP-Anwendung installieren
- Composer auf Elastic Beanstalk aktualisieren
- php.ini in Ihrer Elastic Beanstalk Beanstalk-Konfiguration erweitern

Das AWS SDK for PHP installieren

Wenn Sie AWS Ressourcen von Ihrer Anwendung aus verwalten müssen, installieren Sie den AWS SDK für PHP. Sie können mit dem SDK for PHP beispielsweise Amazon DynamoDB (DynamoDB) zum Speichern von Benutzer- und Sitzungsinformationen verwenden, ohne eine relationale Datenbank zu erstellen.

Um das SDK for PHP mit Composer zu installieren

```
$ composer require aws/aws-sdk-php
```

Weitere Informationen finden Sie auf der AWS SDK for PHPHomepage. Anweisungen finden Sie unter Installieren von AWS SDK for PHP.

Überlegungen zu PHP 8.1 auf Amazon Linux 2

Lesen Sie diesen Abschnitt, wenn Sie den Plattformzweig PHP 8.1 auf Amazon Linux 2 verwenden.

Überlegungen zu PHP 8.1 auf Amazon Linux 2

> **Note**
>
> Die Informationen in diesem Thema beziehen sich nur auf den Plattformzweig PHP 8.1 auf Amazon Linux 2. Dies gilt nicht für die PHP-Plattformzweige, die auf basieren AL2023. Sie gelten auch nicht für den Plattformzweig PHP 8.0 Amazon Linux 2.

Elastic Beanstalk speichert die PHP-8.1-bezogenen RPM-Pakete für den Plattformzweig PHP 8.1 auf Amazon Linux 2 auf den EC2-Instances in einem lokalen Verzeichnis statt im Amazon-Linux-Repository. Sie können rpm -i verwenden, um Pakete zu installieren. Beginnend mit PHP 8.1 Plattformversion 3.5.0 speichert Elastic Beanstalk die PHP 8.1-bezogenen RPM-Pakete im folgenden lokalen EC2-Verzeichnis:

`/opt/elasticbeanstalk/RPMS`

Im folgenden Beispiel wird das php-debuginfo-Paket installiert.

```
$rpm -i /opt/elasticbeanstalk/RPMS/php-debuginfo-8.1.8-1.amzn2.x86_64.rpm
```

Die Version im Paketnamen hängt von der tatsächlichen Version ab, die im lokalen EC2-Verzeichnis `/opt/elasticbeanstalk/RPMS` aufgeführt ist. Verwenden Sie dieselbe Syntax, um andere PHP 8.1 RPM-Pakete zu installieren.

Erweitern Sie den folgenden Abschnitt, um eine Liste der von uns bereitgestellten RPM-Pakete anzuzeigen.

RPM-Pakete

Die folgende Liste enthält die RMP-Pakete, die die Elastic Beanstalk PHP 8.1-Plattform auf Amazon Linux 2 bereitstellt. Diese befinden sich im lokalen Verzeichnis `/opt/elasticbeanstalk/RPMS`.

Die Versionsnummern 8.1.8-1 und 3.7.0-1 in den aufgelisteten Paketnamen sind nur ein Beispiel.

- `php-8.1.8-1.amzn2.x86_64.rpm`
- `php-bcmath-8.1.8-1.amzn2.x86_64.rpm`
- `php-cli-8.1.8-1.amzn2.x86_64.rpm`
- `php-common-8.1.8-1.amzn2.x86_64.rpm`

- php-dba-8.1.8-1.amzn2.x86_64.rpm
- php-dbg-8.1.8-1.amzn2.x86_64.rpm
- php-debuginfo-8.1.8-1.amzn2.x86_64.rpm
- php-devel-8.1.8-1.amzn2.x86_64.rpm
- php-embedded-8.1.8-1.amzn2.x86_64.rpm
- php-enchant-8.1.8-1.amzn2.x86_64.rpm
- php-fpm-8.1.8-1.amzn2.x86_64.rpm
- php-gd-8.1.8-1.amzn2.x86_64.rpm
- php-gmp-8.1.8-1.amzn2.x86_64.rpm
- php-intl-8.1.8-1.amzn2.x86_64.rpm
- php-ldap-8.1.8-1.amzn2.x86_64.rpm
- php-mbstring-8.1.8-1.amzn2.x86_64.rpm
- php-mysqlnd-8.1.8-1.amzn2.x86_64.rpm
- php-odbc-8.1.8-1.amzn2.x86_64.rpm
- php-opcache-8.1.8-1.amzn2.x86_64.rpm
- php-pdo-8.1.8-1.amzn2.x86_64.rpm
- php-pear-1.10.13-1.amzn2.noarch.rpm
- php-pgsql-8.1.8-1.amzn2.x86_64.rpm
- php-process-8.1.8-1.amzn2.x86_64.rpm
- php-pspell-8.1.8-1.amzn2.x86_64.rpm
- php-snmp-8.1.8-1.amzn2.x86_64.rpm
- php-soap-8.1.8-1.amzn2.x86_64.rpm
- php-sodium-8.1.8-1.amzn2.x86_64.rpm
- php-xml-8.1.8-1.amzn2.x86_64.rpm
- php-pecl-imagick-3.7.0-1.amzn2.x86_64.rpm
- php-pecl-imagick-debuginfo-3.7.0-1.amzn2.x86_64.rpm
- php-pecl-imagick-devel-3.7.0-1.amzn2.noarch.rpm

Sie können die Pakete PEAR und PECL verwenden, um allgemeine Erweiterungen zu installieren. Weitere Informationen zu PEAR finden Sie auf der Website PEAR PHP Extension and Application Repository. Weitere Informationen zu PECL finden Sie auf der Website der PECL-Erweiterung.

Mit den folgenden Beispielbefehlen werden die Memcached-Erweiterungen installiert.

```
$pecl install memcache
```

Sie können auch Folgendes verwenden:

```
$pear install pecl/memcache
```

Die folgenden Beispielbefehle installieren die Redis-Erweiterungen.

```
$pecl install redis
```

Sie können auch Folgendes verwenden:

```
$pear install pecl/redis
```

Konfigurieren Ihrer PHP-Umgebung

Mit der Elastic Beanstalk-Konsole können Sie die Protokollrotation für Amazon S3 aktivieren, Variablen konfigurieren, die die Anwendung in der Umgebung lesen kann, und die PHP-Einstellungen ändern.

So konfigurieren Sie Ihre PHP-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

PHP-Einstellungen

- Proxy server (Proxy-Server) – Der Proxyserver, der in Ihren Umgebungs-Instances verwendet werden soll. Standardmäßig wird nginx verwendet.

- Document root (Dokumentenstamm) – In diesem Ordner ist die Standardseite Ihrer Website enthalten. Falls sich die Willkommenseite nicht im Stamm des Quell-Bundles befindet, geben Sie den entsprechenden Ordner relativ zum Stammpfad an. Beispielsweise `/public`, wenn die Willkommenseite im Ordner `public` ist.
- Memory limit (Speicherlimit) – Der maximale Speicher, der von einem Skript zugeordnet werden kann. Beispiel, `512M`.
- Zlib output compression (Zlib-Ausgabekomprimierung) – Legen Sie den Wert auf `On` fest, um Antworten zu komprimieren.
- Allow URL fopen (DÖffnen für URLs zulassen) – Legen Sie den Wert auf `Off` fest, damit Skripts keine Dateien von Remote-Speicherorten herunterladen können.
- Display errors (Fehler anzeigen) – Legen Sie den Wert auf `On` fest, damit interne Fehlermeldungen zum Debuggen angezeigt werden.
- Max execution time (Max. Ausführungszeit) – Gibt die maximale Dauer (in Sekunden) an, die ein Skript ausgeführt werden kann, bevor es von der Umgebung beendet wird.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen hat zwei Einstellungen:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon-EC2-Instances der Anwendung in den der Anwendung zugeordneten Amazon-S3-Bucket kopiert werden.

Statische Dateien

Um die Leistung zu verbessern, können Sie den Abschnitt Static files (Statische Dateien) benutzen, um den Proxy-Server so zu konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zum Verzeichnis-Mapping fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, anstatt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter the section called “Statische Dateien”.

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon-EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Diese Einstellungen werden als Schlüsselwert-Paare an die Anwendung weitergeleitet.

Ihr Anwendungscode kann mithilfe der Funktion `$_SERVER` oder `get_cfg_var` auf Umgebungseigenschaften zugreifen.

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Namespaces für die Konfiguration

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben durchführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Die folgenden Namespaces konfigurieren sowohl Ihren Proxydienst als auch PHP-spezifische Optionen:

- `aws:elasticbeanstalk:environment:proxy:staticfiles`— konfiguriert den Umgebungs-Proxy so, dass er statische Dateien bereitstellt. Sie definieren Zuordnungen von virtuellen Pfaden zu Anwendungsverzeichnissen.
- `aws:elasticbeanstalk:environment:proxy`— gibt den Proxyserver der Umgebung an.
- `aws:elasticbeanstalk:container:php:phpini`— konfiguriert PHP-spezifische Optionen. Dieser Namespace beinhaltet`composer_options`, was auf der Elastic Beanstalk Beanstalk-Konsole nicht verfügbar ist. Diese Option legt die benutzerdefinierten Optionen fest, die bei der Installation von Abhängigkeiten mit Composer über den Befehl verwendet werden sollen. `composer.phar install` Weitere Informationen zu diesem Befehl, einschließlich der verfügbaren Optionen, finden Sie unter install auf der Website getcomposer.org.

Die folgende Beispielkonfigurationsdatei gibt eine Option für statische Dateien an, mit der ein Verzeichnis mit dem Namen `staticimages` dem Pfad `/images` zugeordnet wird, und zeigt die

Einstellungen jeder der im Namespace `aws:elasticbeanstalk:container:php:phpini` verfügbaren Optionen an:

Example.ebextensions/php-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /images: staticimages
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
    memory_limit: 128M
    zlib.output_compression: "Off"
    allow_url_fopen: "On"
    display_errors: "Off"
    max_execution_time: 60
    composer_options: vendor/package
```

> **Note**
>
> Der Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` ist auf Amazon Linux AMI PHP-Plattformvarianten (Vorgänger von Amazon Linux 2) nicht definiert.

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Die Abhängigkeiten Ihrer Elastic Beanstalk PHP-Anwendung installieren

In diesem Thema wird beschrieben, wie Sie Ihre Anwendung so konfigurieren, dass sie andere PHP-Pakete installiert, die sie benötigt. Ihre Anwendung ist möglicherweise von anderen PHP-Paketen abhängig. Sie können Ihre Anwendung so konfigurieren, dass diese Abhängigkeiten auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances der Umgebung installiert werden. Alternativ können Sie die Abhängigkeiten Ihrer Anwendung in das Quell-Bundle einschließen und sie mit der Anwendung bereitstellen. Im folgenden Abschnitt werden beide Möglichkeiten vorgestellt.

Verwenden einer Composer-Datei zum Installieren von Abhängigkeiten von Instances

Mit einer `composer.json`-Datei im Stammverzeichnis der Projektquelle können Sie die von der Anwendung benötigten Pakete über Composer auf den Amazon EC2-Instances Ihrer Umgebung installieren.

Example composer.json

```
{
    "require": {
        "monolog/monolog": "1.0.*"
    }
}
```

Wenn eine `composer.json`-Datei vorhanden ist, führt Elastic Beanstalk `composer.phar install` aus, um Abhängigkeiten zu installieren. Sie können Optionen für den Befehl anfügen, indem Sie die Option `composer_options` im Namespace `aws:elasticbeanstalk:container:php:phpini` festlegen.

Einschließen von Abhängigkeiten in Quell-Bundles

Wenn Ihre Anwendung viele Abhängigkeiten hat, kann die Installation etwas länger dauern. Dies kann die Bereitstellung und Skalierung von Operationen erhöhen, da auf jeder neuen Instance Abhängigkeiten installiert sind.

Um negative Auswirkungen auf die Bereitstellungszeit zu vermeiden, können Sie mit Composer in Ihrer Entwicklungsumgebung Abhängigkeiten auflösen und sie im Ordner `vendor` zu installieren.

So schließen Sie Abhängigkeiten in das Quell-Bundle Ihrer Anwendung ein

1. Führen Sie den folgenden Befehl aus:

   ```
   % composer install
   ```

2. Fügen Sie den generierten Ordner `vendor` zum Stamm des Quell-Bundles der Anwendung hinzu.

Wenn Elastic Beanstalk einen Ordner `vendor` in der Instance findet, wird die Datei `composer.json` ignoriert (selbst, wenn sie vorhanden ist). Ihre Anwendung verwendet dann die Abhängigkeiten im Ordner `vendor`.

Composer auf Elastic Beanstalk aktualisieren

In diesem Thema wird beschrieben, wie Elastic Beanstalk konfiguriert wird, um Composer auf dem neuesten Stand zu halten. Möglicherweise müssen Sie Composer aktualisieren, wenn beim Versuch, Pakete mit einer Composer-Datei zu installieren, ein Fehler auftritt oder wenn Sie die neueste Plattformversion nicht verwenden können. Zwischen Plattformupdates können Sie Composer in Ihren Umgebungsinstanzen mithilfe der Konfigurationsdateien in Ihrem `.ebextensions`Ordner aktualisieren.

Sie können Composer mit der folgenden Konfiguration selbst aktualisieren.

```
commands:
  01updateComposer:
    command: /usr/bin/composer.phar self-update 2.7.0
```

Die folgende Optionseinstellung legt die `COMPOSER_HOME` Umgebungsvariable fest, die den Speicherort des Composer-Cache konfiguriert.

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: COMPOSER_HOME
    value: /home/webapp/composer-home
```

Sie können beide in derselben Konfigurationsdatei in Ihrem `.ebextensions` Ordner kombinieren.

Example.ebextensions/composer.config

```
commands:
  01updateComposer:
    command: /usr/bin/composer.phar self-update 2.7.0

option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: COMPOSER_HOME
    value: /home/webapp/composer-home
```

> **Note**
>
> Aufgrund von Aktualisierungen der Composer-Installation in der Plattformversion vom 22. Februar 2024 und der AL2023 Plattformversion vom 28. Februar 2024 schlägt das

Composer-Selbstupdate möglicherweise fehl, wenn `COMPOSER_HOME` es bei der Ausführung des Selbstupdates festgelegt wird. AL2
Die folgenden kombinierten Befehle können nicht ausgeführt werden: `export COMPOSER_HOME=/home/webapp/composer-home && /usr/bin/composer.phar self-update 2.7.0`
Das vorherige Beispiel funktioniert jedoch. Im vorherigen Beispiel `COMPOSER_HOME` wird die Optionseinstellung für nicht an die `01updateComposer` Ausführung weitergegeben, und sie wird auch nicht gesetzt, wenn der Self-Update-Befehl ausgeführt wird.

Important

Falls Sie die Versionsnummer aus dem Befehl `composer.phar self-update` weglassen, wird Composer jedes Mal auf die neueste verfügbare Version aktualisiert, wenn Sie neuen Quellcode bereitstellen und neue Instances von der Auto Scaling-Funktion zur Verfügung gestellt werden. Das kann zu fehlerhaften Skalierungen und Bereitstellungen führen, falls eine Composer-Version veröffentlicht wird, die mit Ihrer Anwendung nicht kompatibel ist.

Weitere Informationen zu den Elastic-Beanstalk PHP-Plattformen, einschließlich der Composer-Version, finden Sie unter PHP-Plattformversionen im Dokument AWS Elastic Beanstalk -Plattformen.

php.ini in Ihrer Elastic Beanstalk Beanstalk-Konfiguration erweitern

Verwenden Sie eine Konfigurationsdatei mit einem `files`-Block, um eine `.ini`-Datei zu `/etc/php.d/` auf den Instances der Umgebung hinzuzufügen. Von der Hauptkonfigurationsdatei `php.ini` werden die Einstellungen aus den Dateien in diesem Ordner in alphabetischer Reihenfolge abgerufen. Viele Erweiterungen werden standardmäßig über die Dateien in diesem Ordner aktiviert.

Example.ebextensions/mongo.config

```
files:
  "/etc/php.d/99mongo.ini":
    mode: "000755"
    owner: root
    group: root
    content: |
      extension=mongo.so
```

Fortgeschrittene Beispiele für PHP in Elastic Beanstalk

Um mit PHP-Anwendungen zu beginnen AWS Elastic Beanstalk, benötigen Sie ein Anwendungsquellpaket, das Sie als erste Anwendungsversion hochladen und in einer Umgebung bereitstellen können.

Wir empfehlen, mit einer einfachen PHP-Anwendung QuickStart für PHP zu beginnen, die mit der EB-CLI bereitgestellt wird.

PHP-Beispiele für Fortgeschrittene

- Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer PHP Elastic Beanstalk Beanstalk-Umgebung
- Bereitstellen einer Laravel-Anwendung in Elastic Beanstalk
- Bereitstellen einer CakePHP-Anwendung in Elastic Beanstalk
- Bereitstellen einer Symfony-Anwendung in Elastic Beanstalk
- Bereitstellen einer PHP-Anwendung mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank für Elastic Beanstalk
- Bereitstellung einer WordPress Website mit hoher Verfügbarkeit mit einer externen Amazon RDS-Datenbank auf Elastic Beanstalk
- Bereitstellen einer Drupal-Website mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank für Elastic Beanstalk

Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer PHP Elastic Beanstalk Beanstalk-Umgebung

Dieses Thema enthält Anweisungen zum Erstellen eines Amazon RDS mit der Elastic-Beanstalk-Konsole. Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann an Ihre Umgebung gekoppelt sein und von Elastic Beanstalk verwaltet werden oder sie kann als entkoppelte Datenbank erstellt und extern von einem anderen Service verwaltet werden. In dieser Anleitung wird die Datenbank an Ihre Umgebung gekoppelt und von Elastic Beanstalk verwaltet. Weitere Informationen zur Integration eines Amazon RDS mit Elastic Beanstalk finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Sections

- Hinzufügen einer DB-Instance zu Ihrer Umgebung

- Herunterladen eines Treibers
- Mit einem PDO oder My verbinden Sie sich mit einer Datenbank SQLi
- Verbinden mit einer Datenbank mit Symfony

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftenwert
`RDS_HOSTNAME`	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
`RDS_PORT`	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.

Eigenschaftenname	Beschreibung	Eigenschaftenwert
RDS_DB_NAME	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.

Weitere Informationen zur Konfiguration einer Datenbank-Instance in Verbindung mit einer Elastic-Beanstalk-Umgebung finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Herunterladen eines Treibers

Zur Verwendung von PHP Data Objects (PDO), um eine Verbindung mit der Datenbank herzustellen, installieren Sie den Treiber für die Datenbank-Engine, die Sie ausgewählt haben.

- MySQL – PDO_MYSQL
- PostgreSQL – PDO_PGSQL
- Oracle – PDO_OCI
- SQL Server – PDO_SQLSRV

Weitere Informationen finden Sie unter http://php.net/manual/en/pdo.installation.php.

Mit einem PDO oder My verbinden Sie sich mit einer Datenbank SQLi

Sie können mit $_SERVER[`*VARIABLE*`] die Verbindungsinformationen in der Umgebung lesen.

Erstellen Sie für ein PDO einen Data Source Name (DSN) bestehend aus Host, Port und Namen. Geben Sie den DSN an den Konstruktor für das PDO mit dem Benutzernamen und Passwort der Datenbank weiter.

Example Verbinden mit einer RDS-Datenbank mit PDO – MySQL

```
<?php
$dbhost = $_SERVER['RDS_HOSTNAME'];
$dbport = $_SERVER['RDS_PORT'];
$dbname = $_SERVER['RDS_DB_NAME'];
$charset = 'utf8' ;

$dsn = "mysql:host={$dbhost};port={$dbport};dbname={$dbname};charset={$charset}";
$username = $_SERVER['RDS_USERNAME'];
$password = $_SERVER['RDS_PASSWORD'];

$pdo = new PDO($dsn, $username, $password);
?>
```

Für andere Treiber ersetzen Sie `mysql` durch den Namen Ihres Treibers – `pgsql`, `oci` oder `sqlsrv`.

Übergeben Sie für My SQLi den Hostnamen, den Benutzernamen, das Passwort, den Datenbanknamen und den Port an den `mysqli` Konstruktor.

Example Verbinden mit einer RDS-Datenbank mit mysqli_connect()

```
$link = new mysqli($_SERVER['RDS_HOSTNAME'], $_SERVER['RDS_USERNAME'],
 $_SERVER['RDS_PASSWORD'], $_SERVER['RDS_DB_NAME'], $_SERVER['RDS_PORT']);
```

Verbinden mit einer Datenbank mit Symfony

Für Symfony ab Version 3.2 können Sie mit `%env(`*`PROPERTY_NAME`*`)%` basierend auf den Elastic Beanstalk-Umgebungseigenschaften Datenbankparameter in einer Konfigurationsdatei festlegen.

Example app/config/parameters.yml

```
parameters:
    database_driver:   pdo_mysql
    database_host:     '%env(RDS_HOSTNAME)%'
    database_port:     '%env(RDS_PORT)%'
```

```
    database_name:     '%env(RDS_DB_NAME)%'
    database_user:     '%env(RDS_USERNAME)%'
    database_password: '%env(RDS_PASSWORD)%'
```

Unter Externe Parameter (Symfony 3.4) finden Sie weitere Informationen.

In früheren Versionen von Symfony konnte auf Umgebungsvariablen nur zugegriffen werden, wenn der Name mit `SYMFONY__` begann. Dies bedeutet, dass die für Elastic Beanstalk definierten Umgebungseigenschaften nicht zugänglich sind und Sie eigene Umgebungseigenschaften definieren müssen, um die Verbindungsdaten an Symfony zu übermitteln.

Zum Verbinden einer Datenbank mit Symfony 2 erstellen Sie eine Umgebungseigenschaft für jeden Parameter. Anschließend verwenden Sie `%property.name%`, um in einer Konfigurationsdatei auf die für Symfony umgewandelte Variable zuzugreifen. Beispiel: Auf eine Umgebungseigenschaft namens `SYMFONY__DATABASE__USER` kann als `database.user` zugegriffen werden.

```
    database_user:     "%database.user%"
```

Unter Externe Parameter (Symfony 2.8) finden Sie weitere Informationen.

Bereitstellen einer Laravel-Anwendung in Elastic Beanstalk

Laravel ist ein Open-Source-Framework model-view-controller (MVC) für PHP. Dieses Tutorial führt Sie durch den Prozess, eine Laravel-Anwendung zu generieren, sie in einer AWS Elastic Beanstalk Umgebung bereitzustellen und sie so zu konfigurieren, dass sie eine Verbindung zu einer Amazon Relational Database Service (Amazon RDS) -Datenbank-Instance herstellt.

Sections

- Voraussetzungen
- Starten einer Elastic Beanstalk-Umgebung
- Installieren von Laravel und Erstellen einer Website
- Bereitstellen der Anwendung
- Konfigurieren von Composer-Einstellungen
- Hinzufügen einer Datenbank zu Ihrer Umgebung
- Bereinigen
- Nächste Schritte

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole kennen. Sofern noch nicht geschehen, starten Sie anhand der Anleitung in Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können die erste Elastic-Beanstalk-Umgebung.

Für die Verfahren in diesem Leitfaden benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

Laravel 6 erfordert PHP 7.2 oder höher. Außerde werden die PHP-Erweiterungen benötigt, die unter dem Thema Serveranforderungen in der offiziellen Laravel-Dokumentation aufgeführt sind. Folgen Sie den Anweisungen zur Installation von PHP und Composer.

Weitere Informationen zu Support und Wartung von Laravel finden Sie im Thema Support-Richtlinie in der offiziellen Laravel-Dokumentation.

Starten einer Elastic Beanstalk-Umgebung

Erstellen Sie mit der Elastic Beanstalk-Konsole, eine Elastic Beanstalk-Umgebung. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole über diesen vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication?Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.

4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus und wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.
- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.

- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

> **Note**
>
> Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon S3.

Installieren von Laravel und Erstellen einer Website

Composer kann Laravel installieren und ein Arbeitsprojekt mit einem Befehl erstellen:

```
~$ composer create-project --prefer-dist laravel/laravel eb-laravel
```

Composer installiert Laravel und seine Abhängigkeiten und generiert ein Standardprojekt.

Wenn Sie Probleme bei der Installation von Laravel haben, lesen Sie das Installationsthema in der offiziellen Dokumentation: 6.x. https://laravel.com/docs/

Bereitstellen der Anwendung

Erstellen Sie mit den Dateien, die von Composer erstellt wurden, ein Quell-Bundle . Der folgende Befehl erstellt ein Quell-Bundle mit dem Namen `laravel-default.zip`. Die Dateien im Ordner `vendor` werden ausgeschlossen, weil sie viel Platz benötigen und zum Bereitstellen der Anwendung für Elastic Beanstalk nicht erforderlich sind.

```
~/eb-laravel$ zip ../laravel-default.zip -r * .[^.]* -x "vendor/*"
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Laravel in der Umgebung bereitzustellen.

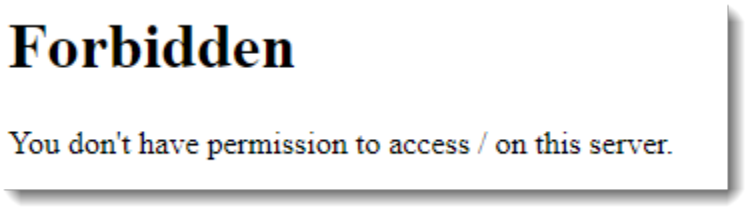
So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

> **Note**
>
> Um das Quell-Bundle weiter zu optimieren, initialisieren Sie ein Git-Repository und verwenden den Befehl `git archive`, um das Quell-Bundle zu erstellen. Das Laravel-Standardprojekt enthält eine `.gitignore`-Datei, die Git anweist, den Ordner `vendor` sowie andere Dateien, die für die Bereitstellung nicht benötigt werden, auszuschließen.

Konfigurieren von Composer-Einstellungen

Klicken Sie nach Abschluss der Bereitstellung auf die URL, um die Laravel-Anwendung im Browser zu öffnen:



Was ist das? Elastic Beanstalk bedient das Stammverzeichnis Ihres Projekts standardmäßig auf dem Stammpfad der Website. In diesem Fall liegt die Standardseite (`index.php`) jedoch eine Ebene weiter unten im `public`-Ordner. Das können Sie überprüfen, indem Sie `/public` zur URL hinzufügen. Beispiel, `http://laravel.us-east-2.elasticbeanstalk.com/public`.

Damit die Anwendung Laravel im Stammverzeichnis verwendet werden kann, konfigurieren Sie mit der Elastic Beanstalk-Konsole das Dokumentenstammverzeichnis der Website.

So konfigurieren Sie das Stammverzeichnis Ihrer Website

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Geben Sie für Document root (Basisverzeichnis) **`/public`** ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.
7. Wenn die Aktualisierung abgeschlossen ist, klicken Sie auf die URL, um Ihre Website erneut im Browser zu öffnen.



So weit, so gut. Jetzt fügen Sie eine Datenbank zu Ihrer Umgebung hinzu und konfigurieren eine Verbindung mit Laravel.

Hinzufügen einer Datenbank zu Ihrer Umgebung

Starten Sie eine RDS-DB-Instance in Ihrer Elastic Beanstalk-Umgebung. Sie können MySQL- oder PostgreSQL-Datenbanken mit Laravel auf Elastic Beanstalk verwenden. SQLServer In diesem Beispiel wird MySQL verwendet.

So fügen Sie Ihrer Elastic Beanstalk-Umgebung eine RDS-DB-Instance hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten) aus.
5. Wählen Sie für Engine mysql aus.
6. Geben Sie einen Master-username (Benutzernamen) und ein Master-password (Passwort) ein. Elastic Beanstalk stellt diese Werte für Ihre Anwendung mithilfe von Umgebungseigenschaften bereit.
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Erstellen einer Datenbank-Instance dauert ca. zehn Minuten. Weitere Informationen zu Datenbanken, die mit einer Elastic-Beanstalk-Umgebung gekoppelt sind, finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

In der Zwischenzeit können Sie Ihren Quell-Code aktualisieren, um Verbindungsinformationen aus der Umgebung zu lesen. Elastic Beanstalk bietet Verbindungsdetails mithilfe von Umgebungsvariablen an, wie z. B. `RDS_HOSTNAME`, auf die Sie in Ihrer Anwendung zugreifen können.

Die Laravel-Datenbankkonfiguration ist in einer Datei mit dem Namen `database.php` im `config`-Ordner im Projektcode gespeichert. Suchen Sie den Eintrag `mysql` und ändern Sie die Variablen `host`, `database`, `username` und `and password`, damit sie die entsprechenden Werte aus Elastic Beanstalk lesen:

Example~/Eb- .php laravel/config/database

```
...
    'connections' => [

        'sqlite' => [
            'driver' => 'sqlite',
            'database' => env('DB_DATABASE', database_path('database.sqlite')),
            'prefix' => '',
        ],

        'mysql' => [
            'driver' => 'mysql',
            'host' => env('RDS_HOSTNAME', '127.0.0.1'),
            'port' => env('RDS_PORT', '3306'),
            'database' => env('RDS_DB_NAME', 'forge'),
            'username' => env('RDS_USERNAME', 'forge'),
            'password' => env('RDS_PASSWORD', ''),
            'unix_socket' => env('DB_SOCKET', ''),
            'charset' => 'utf8mb4',
            'collation' => 'utf8mb4_unicode_ci',
            'prefix' => '',
            'strict' => true,
            'engine' => null,
        ],
...
```

Um zu überprüfen, ob die Datenbankverbindung ordnungsgemäß konfiguriert ist, fügen Sie `index.php` den Code hinzu, um eine Verbindung zur Datenbank herzustellen und fügen Code zur Standardantwort hinzu:

Example~/Eb- .php laravel/public/index

```
...
if(DB::connection()->getDatabaseName())
{
   echo "Connected to database ".DB::connection()->getDatabaseName();
}
$response->send();
...
```

Wenn die DB-Instance den Start abgeschlossen hat, bündeln Sie die aktualisierte Anwendung und stellen Sie sie in Ihrer Umgebung bereit.

So aktualisieren Sie Ihre Elastic Beanstalk-Umgebung

1. Erstellen Sie ein neues Quell-Bundle:

   ```
   ~/eb-laravel$ zip ../laravel-v2-rds.zip -r * .[^.]* -x "vendor/*"
   ```

2. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
3. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
4. Klicken Sie auf die Schaltfläche Upload and Deploy (Hochladen und bereitstellen).
5. Wählen Sie Browse (Durchsuchen) und laden Sie `laravel-v2-rds.zip` hoch.
6. Wählen Sie Bereitstellen.

Die Bereitstellung einer neuen Version Ihrer Anwendung dauert weniger als eine Minute. Wenn die Bereitstellung abgeschlossen ist, aktualisieren Sie die Webseite erneut, um zu überprüfen, ob die Datenbankverbindung erfolgreich war:



Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung über die Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb der Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken aus.

3. Wählen Sie Ihre DB-Instance aus.
4. Wählen Sie Aktionen und anschließend Löschen aus.
5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Weitere Informationen über Laravel finden Sie auf der offiziellen Laravel-Website unter laravel.com.

Im weiteren Verlauf der Anwendungsentwicklung können Sie Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die Elastic Beanstalk Command Line Interface (EB CLI) bietet easy-to-use Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk Beanstalk-Umgebungen von der Befehlszeile aus.

In diesem Tutorial haben Sie die Composer-Optionen mit der Elastic Beanstalk-Konsole konfiguriert. Um diese Konfiguration zum Teil der Anwendungsquelle zu machen, können Sie eine Konfigurationsdatei wie diese verwenden.

Example.ebextensions/composer.config

```
option_settings:
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

Weitere Informationen finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

Das Ausführen einer Amazon RDS-DB-Instance in Ihrer Elastic Beanstalk-Umgebung eignet sich hervorragend für Entwicklung und Tests, sie bindet jedoch den Lebenszyklus Ihrer Datenbank an Ihre Umgebung. Anweisungen zum Herstellen einer Verbindung mit einer Datenbank, die außerhalb der Umgebung ausgeführt wird, finden Sie unter Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer PHP Elastic Beanstalk Beanstalk-Umgebung.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, konfigurieren Sie einen benutzerdefinierten Domainnamen für die Umgebung und aktivieren dann HTTPS für sichere Verbindungen.

Bereitstellen einer CakePHP-Anwendung in Elastic Beanstalk

CakePHP ist ein Open Source MVC-Framework für PHP. In diesem Tutorial wird der Prozess für das Erstellen eines CakePHP-Projekts, dessen Bereitstellung in einer Elastic Beanstalk-Umgebung und die Konfiguration zum Herstellen einer Verbindung mit einer Amazon RDS-Datenbank-Instance erklärt.

Sections

- Voraussetzungen
- Starten einer Elastic Beanstalk-Umgebung
- Installieren von CakePHP und Erstellen einer Website
- Bereitstellen der Anwendung
- Hinzufügen einer Datenbank zu Ihrer Umgebung
- Bereinigen
- Nächste Schritte

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole kennen. Sofern noch nicht geschehen, starten Sie anhand der Anleitung in Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können die erste Elastic-Beanstalk-Umgebung.

Für die Verfahren in diesem Leitfaden benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

CakePHP 4 benötigt PHP 7.4 oder höher. Außerdem benötigt es die PHP-Erweiterungen, die in der offiziellen CakePHP-Installationsdokumentation aufgeführt sind. Sie müssen sowohl PHP als auch Composer installieren.

Starten einer Elastic Beanstalk-Umgebung

Erstellen Sie mit der Elastic Beanstalk-Konsole, eine Elastic Beanstalk-Umgebung. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole über diesen vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus und wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.
- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.

- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

> **Note**
>
> Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon S3.

Installieren von CakePHP und Erstellen einer Website

Composer kann CakePHP installieren und ein Arbeitsprojekt mit einem Befehl erstellen:

```
~$ composer create-project --prefer-dist cakephp/app eb-cake
```

Composer installiert CakePHP und etwa 20 Abhängigkeiten und generiert ein Standardprojekt.

Wenn Probleme beim Installieren von CakePHP auftreten, rufen Sie das Thema Installation in der offiziellen Dokumentation: http://book.cakephp.org/4.0/en/installation.html auf

Bereitstellen der Anwendung

Erstellen Sie mit den Dateien, die von Composer erstellt wurden, ein Quell-Bundle . Der folgende Befehl erstellt ein Quell-Bundle mit dem Namen `cake-default.zip`. Die Dateien im Ordner `vendor` werden ausgeschlossen, weil sie viel Platz benötigen und zum Bereitstellen der Anwendung für Elastic Beanstalk nicht erforderlich sind.

```
eb-cake zip ../cake-default.zip -r * .[^.]* -x "vendor/*"
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um CakePHP in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

> **Note**
>
> Um das Quell-Bundle weiter zu optimieren, initialisieren Sie ein Git-Repository und verwenden den Befehl `git archive`, um das Quell-Bundle zu erstellen. Das Symfony-Standardprojekt enthält eine `.gitignore`-Datei, die Git anweist, den Ordner `vendor` sowie andere Dateien auszuschließen, die für die Bereitstellung nicht benötigt werden.

Wenn der Vorgang abgeschlossen ist, klicken Sie auf die URL, um Ihre CakePHP-Anwendung im Browser zu öffnen.

So weit, so gut. Anschließend fügen Sie Ihrer Umgebung eine Datenbank hinzu und konfigurieren eine CakePHP-Verbindung.

Hinzufügen einer Datenbank zu Ihrer Umgebung

Starten Sie in Ihrer Elastic Beanstalk-Umgebung eine Amazon RDS-Datenbank-Instance. Sie können MySQL- oder PostgreSQL-Datenbanken mit CakePHP auf Elastic Beanstalk verwenden. SQLServer In diesem Beispiel wird PostgreSQL verwendet.

So fügen Sie Ihrer Elastic Beanstalk-Umgebung eine Amazon RDS-DB-Instance hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie unter Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie für DB engine postgres aus.
6. Geben Sie einen Master-username (Benutzernamen) und ein Master-password (Passwort) ein. Elastic Beanstalk stellt diese Werte für Ihre Anwendung mithilfe von Umgebungseigenschaften bereit.
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Erstellen einer Datenbank-Instance dauert ca. zehn Minuten. In der Zwischenzeit können Sie Ihren Quell-Code aktualisieren, um Verbindungsinformationen aus der Umgebung zu lesen. Elastic

Beanstalk bietet Verbindungsdetails mithilfe von Umgebungsvariablen, wie z. B. `RDS_HOSTNAME`, auf die Sie in Ihrer Anwendung zugreifen können.

Die CakePHP-Datenbankkonfiguration befindet sich in der Datei `app.php` im `config`-Ordner in Ihrem Projektcode. Öffnen Sie diese Datei und fügen Sie einigen Code hinzu, der die Umgebungsvariablen aus `$_SERVER` liest und sie den lokalen Variablen zuweist. Fügen Sie die hervorgehobenen Zeilen im untenstehenden Beispiel nach der ersten Zeile ein (`<?php`):

Example~/Eb- .php cake/config/app

```
<?php
if (!defined('RDS_HOSTNAME')) {
  define('RDS_HOSTNAME', $_SERVER['RDS_HOSTNAME']);
  define('RDS_USERNAME', $_SERVER['RDS_USERNAME']);
  define('RDS_PASSWORD', $_SERVER['RDS_PASSWORD']);
  define('RDS_DB_NAME', $_SERVER['RDS_DB_NAME']);
}
return [
...
```

Die Datenbankverbindung wird weiter unten in `app.php` konfiguriert. Suchen Sie den folgenden Abschnitt und ändern Sie die Standard-Datenquellenkonfiguration mit dem Namen des Treibers für Ihre Datenbank-Engine (`Mysql`, `Sqlserver` oder `Postgres`). Legen Sie die `host`-, `username`-, `password`- und `database`-Variablen fest, damit sie die entsprechenden Werte aus Elastic Beanstalk lesen:

Example~/Eb-.php cake/config/app

```
...
     /**
     * Connection information used by the ORM to connect
     * to your application's datastores.
     * Drivers include Mysql Postgres Sqlite Sqlserver
     * See vendor\cakephp\cakephp\src\Database\Driver for complete list
     */
    'Datasources' => [
        'default' => [
            'className' => 'Cake\Database\Connection',
            'driver' => 'Cake\Database\Driver\Postgres',
            'persistent' => false,
            'host' => RDS_HOSTNAME,
```

```
            /*
             * CakePHP will use the default DB port based on the driver selected
             * MySQL on MAMP uses port 8889, MAMP users will want to uncomment
             * the following line and set the port accordingly
             */
            //'port' => 'non_standard_port_number',
            'username' => RDS_USERNAME,
            'password' => RDS_PASSWORD,
            'database' => RDS_DB_NAME,
            /*
             * You do not need to set this flag to use full utf-8 encoding (internal
 default since CakePHP 3.6).
             */
            //'encoding' => 'utf8mb4',
            'timezone' => 'UTC',
            'flags' => [],
            'cacheMetadata' => true,
            'log' => false,
...
```

Wenn die DB-Instance den Start abgeschlossen hat, bündeln Sie die aktualisierte Anwendung und stellen Sie sie in Ihrer Umgebung bereit:

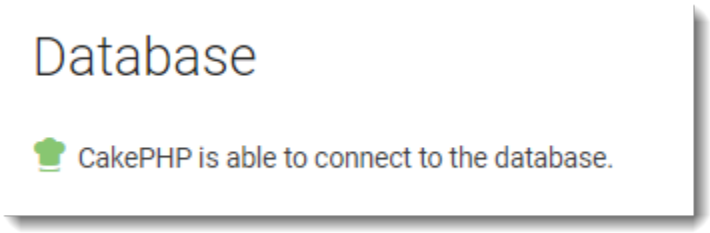
So aktualisieren Sie Ihre Elastic Beanstalk-Umgebung

1. Erstellen Sie ein neues Quell-Bundle:

   ```
   ~/eb-cake$ zip ../cake-v2-rds.zip -r * .[^.]* -x "vendor/*"
   ```

2. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
3. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
4. Klicken Sie auf die Schaltfläche Upload and Deploy (Hochladen und bereitstellen).
5. Wählen Sie Browse (Durchsuchen) und laden Sie `cake-v2-rds.zip` hoch.
6. Wählen Sie Bereitstellen.

Die Bereitstellung einer neuen Version Ihrer Anwendung dauert weniger als eine Minute. Wenn die Bereitstellung abgeschlossen ist, aktualisieren Sie die Webseite erneut, um zu überprüfen, ob die Datenbankverbindung erfolgreich war:



Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung von der Konsole aus

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb der Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie Ihre DB-Instance aus.
4. Wählen Sie Aktionen und anschließend Löschen aus.

5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Weitere Informationen zu CakePHP finden Sie in dem Buch unter book.cakephp.org.

Im weiteren Verlauf der Anwendungsentwicklung können Sie Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die Elastic Beanstalk Command Line Interface (EB CLI) bietet easy-to-use Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk Beanstalk-Umgebungen von der Befehlszeile aus.

Das Ausführen einer Amazon RDS-DB-Instance in Ihrer Elastic Beanstalk-Umgebung eignet sich hervorragend für Entwicklung und Tests, sie bindet jedoch den Lebenszyklus Ihrer Datenbank an Ihre Umgebung. Anweisungen zum Herstellen einer Verbindung mit einer Datenbank, die außerhalb der Umgebung ausgeführt wird, finden Sie unter Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer PHP Elastic Beanstalk Beanstalk-Umgebung.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, konfigurieren Sie einen benutzerdefinierten Domainnamen für die Umgebung und aktivieren dann HTTPS für sichere Verbindungen.

Bereitstellen einer Symfony-Anwendung in Elastic Beanstalk

Symfony ist ein Open-Source-Framework für die Entwicklung dynamischer PHP-Webanwendungen. Dieses Tutorial führt Sie durch den Prozess der Generierung einer Symfony-Anwendung und deren Bereitstellung in einer AWS Elastic Beanstalk Umgebung.

Sections

- Voraussetzungen
- Starten einer Elastic Beanstalk-Umgebung
- Installieren von Symfony und Generieren einer Website
- Bereitstellen der Anwendung
- Konfigurieren von Composer-Einstellungen
- Bereinigen
- Nächste Schritte

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole kennen. Sofern noch nicht geschehen, starten Sie anhand der Anleitung in Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können die erste Elastic-Beanstalk-Umgebung.

Für die Verfahren in diesem Leitfaden benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

Symfony 4.4.9 setzt PHP 7.1.3 oder höher voraus. Es benötigt auch die PHP-Erweiterungen, die unter dem Thema Technische Anforderungen in der offiziellen Symfony-Installationsdokumentation aufgeführt sind. In diesem Tutorial verwenden wir PHP 7.2 und die entsprechende Elastic Beanstalk-Plattformversion. Bevor Sie fortfahren, müssen Sie sowohl PHP als auch Composer installieren.

Weitere Informationen zu Support und Wartung von Symfony finden Sie im Thema Symfony-Versionen auf der Symfony-Website. Weitere Informationen zu Updates im Zusammenhang mit der Unterstützung der PHP-Version für Symfony 4.4.9 finden Sie im Thema Versionshinweise zu Symfony 4.4.9 auf der Symfony-Website.

Starten einer Elastic Beanstalk-Umgebung

Erstellen Sie mit der Elastic Beanstalk-Konsole, eine Elastic Beanstalk-Umgebung. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole über diesen vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication?Anwendungsname=Tutorials&EnvironmentType= LoadBalanced

2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus und wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.
- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.

- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

> **Note**
>
> Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon S3.

Installieren von Symfony und Generieren einer Website

Composer kann Symfony installieren und ein Arbeitsprojekt mit einem Befehl erstellen:

```
~$ composer create-project symfony/website-skeleton eb-symfony
```

Composer installiert Symfony und seine Abhängigkeiten und generiert ein Standardprojekt.

Wenn Probleme bei der Installation von Symfony auftreten, rufen Sie das Thema Installation in der offiziellen Symfony-Dokumentation auf.

Bereitstellen der Anwendung

Rufen Sie das Projektverzeichnis auf.

```
~$ cd eb-symfony
```

Erstellen Sie mit den Dateien, die von Composer erstellt wurden, ein Quell-Bundle . Der folgende Befehl erstellt ein Quell-Bundle mit dem Namen `symfony-default.zip`. Die Dateien im Ordner `vendor` werden ausgeschlossen, weil sie viel Platz benötigen und zum Bereitstellen der Anwendung für Elastic Beanstalk nicht erforderlich sind.

```
eb-symfony$ zip ../symfony-default.zip -r * .[^.]* -x "vendor/*"
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Symfony in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

> **Note**
>
> Um das Quell-Bundle weiter zu optimieren, initialisieren Sie ein Git-Repository und verwenden den Befehl `git archive`, um das Quell-Bundle zu erstellen. Das Symfony-

Standardprojekt enthält eine `.gitignore`-Datei, die Git anweist, den Ordner `vendor` sowie andere Dateien auszuschließen, die für die Bereitstellung nicht benötigt werden.

Konfigurieren von Composer-Einstellungen

Klicken Sie nach Abschluss der Bereitstellung auf die URL, um die Symfony-Anwendung im Browser zu öffnen.

Was ist das? Standardmäßig bedient Elastic Beanstalk das Stammverzeichnis Ihres Projekts im Stammpfad der Website. In diesem Fall liegt die Standardseite (`app.php`) jedoch eine Ebene weiter unten im `web`-Ordner. Das können Sie überprüfen, indem Sie `/public` zur URL hinzufügen. Beispiel, `http://symfony.us-east-2.elasticbeanstalk.com/public`.

Damit die Anwendung Symfony vom Stammverzeichnis aus genutzt werden kann, konfigurieren Sie mit der Elastic Beanstalk-Konsole das Dokumentenstammverzeichnis für die Website.

So konfigurieren Sie das Stammverzeichnis Ihrer Website

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Geben Sie für Document root (Basisverzeichnis) **`/public`** ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.
7. Wenn die Aktualisierung abgeschlossen ist, klicken Sie auf die URL, um Ihre Website erneut im Browser zu öffnen.

Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung über die Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Nächste Schritte

Weitere Informationen über Symfony finden Sie in Was ist Symfony? unter symfony.com.

Im weiteren Verlauf der Anwendungsentwicklung können Sie Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die Elastic Beanstalk Command Line Interface (EB CLI) bietet easy-to-use Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk Beanstalk-Umgebungen von der Befehlszeile aus.

In diesem Tutorial haben Sie die Composer-Optionen mit der Elastic Beanstalk-Konsole konfiguriert. Um diese Konfiguration zum Teil der Anwendungsquelle zu machen, können Sie eine Konfigurationsdatei wie diese verwenden.

Example.ebextensions/composer.config

```
option_settings:
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

Weitere Informationen finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

Symfony verwendet eigener Konfigurationsdateien zum Konfigurieren von Datenbankverbindungen. Zum Abrufen bestimmter PHP-Paketversionen, die der Version auf Ihrer Elastic Beanstalk PHP-

Plattformversion entsprechen, suchen Sie unter Homebrew Formulae verfügbare PHP-Versionen, wie z. B. Verbinden mit einer Datenbank mit Symfony.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, konfigurieren Sie einen benutzerdefinierten Domainnamen für die Umgebung und aktivieren dann HTTPS für sichere Verbindungen.

Bereitstellen einer PHP-Anwendung mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank für Elastic Beanstalk

In diesem Tutorial erfahren Sie, wie Sie eine externe RDS-DB-Instance starten und eine Hochverfügbarkeitsumgebung konfigurieren AWS Elastic Beanstalk, in der eine PHP-Anwendung ausgeführt wird, um eine Verbindung zu ihr herzustellen. Wenn Sie eine DB-Instance außerhalb von Elastic Beanstalk ausführen, wird die Datenbank vom Lebenszyklus Ihrer Umgebung abgekoppelt. Auf diese Weise können Sie mehrere Datenbank mit derselben Datenbank verbinden, eine Datenbank gegen eine andere austauschen oder eine Blau/Grün-Bereitstellung durchführen, ohne dass dies Auswirkungen auf Ihre Datenbank hat.

Das Tutorial verwendet eine PHP-Beispielanwendung mit einer MySQL-Datenbank zum Speichern der vom Benutzer bereitgestellten Textdaten. In der Beispielanwendung werden Konfigurationsdateien verwendet, um PHP-Einstellungen zu konfigurieren und eine Tabelle in der Datenbank zu erstellen. Außerdem erfahren Sie, wie Sie mit einer Composer-Datei Paketen während der Bereitstellung installieren.

Sections

- Voraussetzungen
- Starten einer DB-Instance in Amazon RDS
- Erstellen einer Elastic Beanstalk-Umgebung
- Konfigurieren von Sicherheitsgruppen, Umgebungseigenschaften und der Skalierung
- Bereitstellen der Beispielanwendung
- Bereinigen
- Nächste Schritte

Voraussetzungen

Bevor Sie beginnen, laden Sie das Quellpaket für die Beispielanwendung von GitHub: eb-demo-php-simple -app-1.3.zip herunter

Die Verfahren in diesem Tutorial für Amazon Relational Database Service (Amazon RDS)-Aufgaben gehen davon aus, dass Sie Ressourcen in einer standardmäßigen Amazon Virtual Private Cloud (Amazon VPC) starten. Alle neuen Konten umfassen eine Standard-VPC in jeder Region. Wenn Sie nicht über eine Standard-VPC verfügen, sind die Verfahren unterschiedlich. Anweisungen Verwenden von Elastic Beanstalk mit Amazon RDS für EC2 -Classic- und benutzerdefinierte VPC-Plattformen finden Sie unter.

Starten einer DB-Instance in Amazon RDS

Um eine externe Datenbank mit einer Anwendung zu verwenden, die in Elastic Beanstalk ausgeführt wird, starten Sie zuerst eine DB-Instance mit Amazon RDS. Wenn Sie eine Instance mit Amazon RDS starten, ist diese gänzlich unabhängig von Elastic Beanstalk und den Elastic Beanstalk-Umgebungen. Sie wird von Elastic Beanstalk weder beendet noch überwacht.

Verwenden Sie die Amazon RDS-Konsole, um eine DB-Instance mit MySQL und Multi-AZ-Bereitstellung zu starten. Eine Multi-AZ-Bereitstellung gewährleistet die Ausführung von Failovern für Ihre Datenbank, sodass sie auch bei einem Ausfall der DB-Quell-Instance verfügbar bleibt.

So starten Sie eine RDS DB-Instance in einer Standard-VPC

1. Öffnen Sie die RDS-Konsole.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.
4. Wählen Sie Standard Create (Standarderstellung).

> **Important**
>
> Wählen Sie nicht Easy Create (Einfache Erstellung) aus. Wenn Sie diese Option auswählen, können Sie die erforderlichen Einstellungen zum Starten dieser RDS-Datenbank nicht konfigurieren.

5. Geben Sie unter Additional configuration (Zusätzliche Konfiguration) für Initial database name (Anfangsdatenbankname) „**ebdb**“ ein.
6. Überprüfen Sie die Standardeinstellungen und passen Sie diese Einstellungen an Ihre spezifischen Anforderungen an. Achten Sie auf die folgenden Optionen:

   - DB instance class (DB-Instance-Klasse) – Wählen Sie eine Instance-Größe, die eine geeignete Speichermenge und CPU-Leistung für den Workload bereitstellt.

- Multi-AZ-Bereitstellung — Für hohe Verfügbarkeit stellen Sie diese Option auf Einen Replica/Reader Aurora-Knoten in einer anderen AZ erstellen ein.
- Master username (Master-Benutzername) und Master password (Master-Passwort) – Der Benutzername und das Passwort für die Datenbank. Notieren Sie sich diese Einstellungen, um sie später zu verwenden.

7. Überprüfen Sie die Standardeinstellungen für die übrigen Optionen und wählen Sie dann Create database (Datenbank starten) aus.

Bearbeiten Sie anschließend die Sicherheitsgruppe für die DB-Instance, damit eingehender Datenverkehr über den entsprechenden Port zulässig ist. Es handelt sich um die gleiche Sicherheitsgruppe, die später mit der Elastic Beanstalk-Umgebung verknüpft wird. Die von Ihnen hinzugefügte Regel gewährt damit Eingangsberechtigungen für andere Ressourcen in derselben Sicherheitsgruppe.

So ändern Sie die Eingangsregeln für die Sicherheitsgruppe, die an Ihre RDS-Instance angehängt ist

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance, damit die Details der Instance angezeigt werden.
4. Notieren Sie im Abschnitt Konnektivität die auf dieser Seite angezeigten Subnetze, Sicherheitsgruppen und Endpunkte. Auf diese Weise können Sie diese Informationen später verwenden.
5. In Security (Sicherheit) wird die mit der DB-Instance verknüpfte Sicherheitsgruppe angezeigt. Öffnen Sie den Link, um die Sicherheitsgruppe in der EC2 Amazon-Konsole anzuzeigen.
6. Klicken Sie in den Details der Sicherheitsgruppe auf Inbound (Eingehend).
7. Wählen Sie Edit aus.
8. Klicken Sie auf Add Rule (Regel hinzufügen).
9. Wählen Sie unter Type (Typ) die von der Anwendung verwendete DB-Engine aus.
10. Geben Sie für Source (Quelle) **`sg-`** ein, um eine Liste der verfügbaren Sicherheitsgruppen anzuzeigen. Wählen Sie die Sicherheitsgruppe aus, die mit der Auto-Scaling-Gruppe verknüpft ist, die mit Ihrer Elastic-Beanstalk-Umgebung verwendet wird. Auf diese Weise können EC2 Amazon-Instances in der Umgebung auf die Datenbank zugreifen.



11. Wählen Sie Speichern.

Das Erstellen einer DB-Instance dauert ca. 10 Minuten. In der Zwischenzeit erstellen Sie die Elastic Beanstalk-Umgebung.

Erstellen einer Elastic Beanstalk-Umgebung

Erstellen Sie mit der Elastic Beanstalk-Konsole, eine Elastic Beanstalk-Umgebung. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode. Nachdem Sie die Umgebung gestartet haben, können Sie die Umgebung so konfigurieren, dass sie eine Verbindung zur Datenbank herstellt, und dann die Beispielanwendung bereitstellen, von GitHub der Sie heruntergeladen haben.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole über diesen vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus und wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.
- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet. Die RDS-DB-Instance wurde von Ihnen außerhalb der Umgebung gestartet, daher müssen Sie die Verwaltung ihres Lebenszyklus übernehmen.

> **Note**
>
> Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon S3.

Konfigurieren von Sicherheitsgruppen, Umgebungseigenschaften und der Skalierung

Fügen Sie die Sicherheitsgruppe der DB-Instance zur ausgeführten Umgebung hinzu. Dadurch stellt Elastic Beanstalk alle Instances in der Umgebung mit der zusätzlich zugeordneten Sicherheitsgruppe erneut bereit.

So fügen Sie eine Sicherheitsgruppe zur Umgebung hinzu

- Gehen Sie folgendermaßen vor:
  - So fügen Sie eine Sicherheitsgruppe mit der Elastic Beanstalk-Konsole hinzu
    a. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region

   b. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
   c. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
   d. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).
   e. Wählen Sie unter EC2 Sicherheitsgruppen die Sicherheitsgruppe aus, die an die Instances angehängt werden soll, zusätzlich zu der Instance-Sicherheitsgruppe, die Elastic Beanstalk erstellt.
   f. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.
   g. Lesen Sie die Warnmeldung und klicken Sie dann auf Bestätigen.

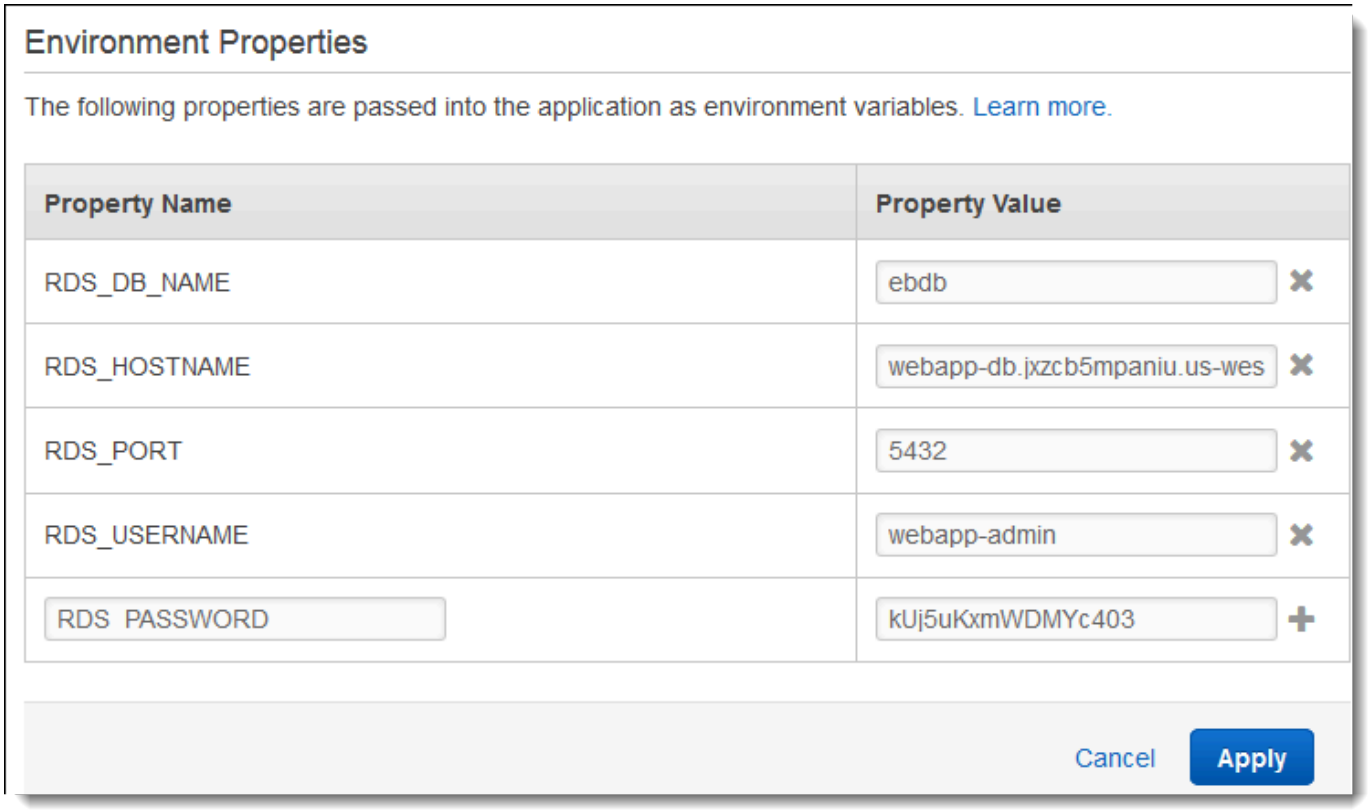
- Fügen Sie eine Sicherheitsgruppe mit einer Konfigurationsdatei hinzufügen. Verwenden Sie die `securitygroup-addexisting.config`-Beispieldatei.

Anschließend werden die Verbindungsinformationen mithilfe von Umgebungseigenschaften an die Umgebung übergeben. Die Beispielanwendung hat Standardeigenschaften, die den Eigenschaften entsprechen, die von Elastic Beanstalk im Rahmen der Datenbankbereitstellung in der Umgebung konfiguriert werden.

So konfigurieren Sie Umgebungseigenschaften für eine DB-Instance von Amazon RDS

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Definieren Sie im Abschnitt Environment properties (Umgebungseigenschaften) die Variablen, die Ihre Anwendung liest, um eine Verbindungszeichenfolge zu bilden. Zur Sicherstellung der Kompatibilität mit Umgebungen, die eine integrierte RDS-DB-Instance haben, verwenden Sie die folgenden Namen und Werte. Sie finden alle Werte außer dem Passwort in der RDS-Konsole.

Name der Eigenschaft	Beschreibung	Eigenschaftenwert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.



6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Konfigurieren Sie abschließend für die Auto-Scaling-Gruppe der Umgebung eine höhere Instance-Mindestanzahl. Führen Sie immer mindestens zwei Instances aus, damit einzelne Webserver in der Umgebung keine Fehlerquelle darstellen und damit Sie Änderungen ohne Ausfallzeit der Website bereitstellen können.

So konfigurieren Sie die Auto-Scaling-Gruppe in der Umgebung für hohe Verfügbarkeit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).

5. Legen Sie im Abschnitt Auto Scaling Group (Gruppe der automatischen Skalierung) für die Option Min instances (Min. Instances) **2** fest.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Bereitstellen der Beispielanwendung

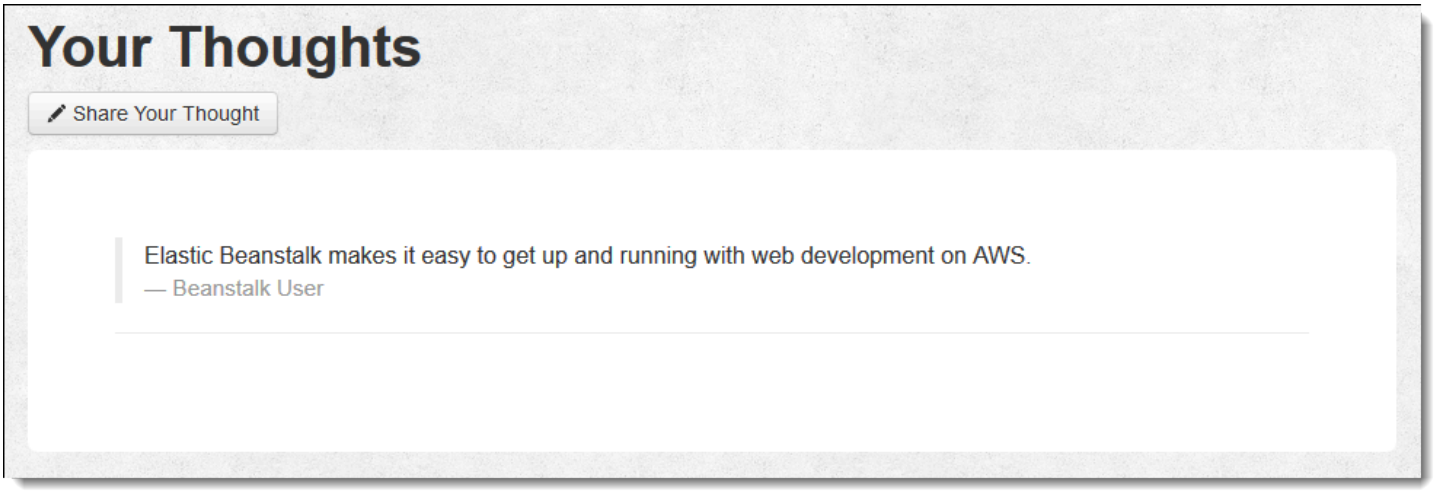
Ihre Umgebung ist jetzt bereit für die Ausführung der Beispielanwendung und die Herstellung einer Verbindung mit Amazon RDS. Stellen Sie die Beispielanwendung in Ihrer Umgebung bereit.

> **Note**
>
> Laden Sie das Quellpaket von herunter GitHub, falls Sie es noch nicht getan haben: - app-1.3.zip eb-demo-php-simple

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Die Website sammelt Benutzerkommentare und speichert die Daten in einer MySQL-Datenbank. Zum Hinzufügen eines Kommentars wählen Sie Share Your Thought (Kommentar eingeben), geben Sie einen Kommentar ein und wählen Sie dann Submit Your Thought (Kommentar senden). Die Web-App schreibt den Kommentar in die Datenbank, sodass alle Instances in der Umgebung ihn lesen können und er nicht verloren geht, wenn Instances außer Betrieb sind.



Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung über die Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb der Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie Ihre DB-Instance aus.
4. Wählen Sie Aktionen und anschließend Löschen aus.
5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Im weiteren Verlauf der Anwendungsentwicklung können Sie Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die Elastic Beanstalk Command Line Interface (EB CLI) bietet easy-to-use Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk Beanstalk-Umgebungen von der Befehlszeile aus.

In der Beispielanwendung werden Konfigurationsdateien verwendet, um PHP-Einstellungen zu konfigurieren und eine Tabelle in der Datenbank zu erstellen (sofern nicht bereits vorhanden). Mit einer Konfigurationsdatei können Sie zudem während des Erstellens der Umgebung die Sicherheitsgruppeneinstellungen der Instance konfigurieren und so zeitaufwendige Aktualisierungen der Konfiguration verhindern. Weitere Informationen finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

Für die Entwicklung sowie für Tests können Sie die Elastic Beanstalk-Funktionalität nutzen und eine verwaltete DB-Instance direkt in der Umgebung hinzufügen. Detaillierte Anweisungen zum Einrichten einer Datenbank in der Umgebung finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Wenn Sie eine Datenbank mit hoher Verfügbarkeit benötigen, können Sie Amazon Aurora verwenden. Amazon Aurora ist eine mit MySQL kompatible Datenbank-Engine, die kommerzielle Datenbank-Features bei geringen Kosten bietet. Um die Verbindung der Anwendung mit einer anderen Datenbank herzustellen, wiederholen Sie die Schritte der Sicherheitsgruppenkonfiguration und aktualisieren Sie die RDS-bezogenen Umgebungseigenschaften.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, konfigurieren Sie einen benutzerdefinierten Domainnamen für die Umgebung und aktivieren dann HTTPS für sichere Verbindungen.

Bereitstellung einer WordPress Website mit hoher Verfügbarkeit mit einer externen Amazon RDS-Datenbank auf Elastic Beanstalk

In diesem Tutorial wird beschrieben, wie Sie eine externe Amazon RDS-DB-Instance starten und anschließend eine Hochverfügbarkeitsumgebung konfigurieren AWS Elastic Beanstalk, auf der eine WordPress Website ausgeführt wird, um eine Verbindung zu ihr herzustellen. Die Website verwendet Amazon Elastic File System (Amazon EFS) als gemeinsamen Speicher für hochgeladene Dateien.

Wenn Sie eine DB-Instance außerhalb von Elastic Beanstalk ausführen, wird die Datenbank vom Lebenszyklus Ihrer Umgebung abgekoppelt. Auf diese Weise können Sie mehrere Umgebungen mit derselben Datenbank verbinden, eine Datenbank gegen eine andere austauschen oder eine Blau/Grün-Bereitstellung durchführen, ohne dass dies Auswirkungen auf Ihre Datenbank hat.

> **Note**
>
> Aktuelle Informationen zur Kompatibilität von WordPress PHP-Versionen mit Versionen finden Sie unter PHP-Kompatibilität und WordPress Versionen auf der WordPress Website. Sie sollten sich diese Informationen ansehen, bevor Sie für Ihre WordPress Implementierungen auf eine neue Version von PHP aktualisieren.

Topics

- Voraussetzungen
- Starten einer DB-Instance in Amazon RDS
- Laden Sie herunter WordPress
- Starten einer Elastic Beanstalk-Umgebung
- Konfigurieren der Sicherheitsgruppen und Umgebungseigenschaften
- Konfigurieren und Bereitstellen Ihrer Anwendung
- Installieren Sie WordPress
- Aktualisieren von Schlüsseln und Salts
- Entfernen von Zugriffsbeschränkungen
- Konfigurieren der Auto-Scaling-Gruppe
- Upgraden WordPress
- Bereinigen
- Nächste Schritte

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole kennen. Sofern noch nicht geschehen, starten Sie anhand der Anleitung in Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können die erste Elastic-Beanstalk-Umgebung.

Für die Verfahren in diesem Leitfaden benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

Standard-VPC

Bei den Verfahren in diesem Tutorial für Amazon Relational Database Service (Amazon RDS) wird davon ausgegangen, dass Sie Ressourcen in einer standardmäßigen Amazon Virtual Private Cloud (Amazon VPC) starten. Alle neuen Konten enthalten eine Standard-VPC in jeder AWS Region. Wenn Sie nicht über eine Standard-VPC verfügen, sind die Verfahren unterschiedlich. Anweisungen Verwenden von Elastic Beanstalk mit Amazon RDS für EC2 -Classic- und benutzerdefinierte VPC-Plattformen finden Sie unter.

AWS Regionen

Die Beispielanwendung verwendet Amazon EFS, das nur in AWS Regionen funktioniert, die Amazon EFS unterstützen. Weitere Informationen zu den unterstützten AWS Regionen finden Sie unter Amazon Elastic File System Endpoints and Quotas in der Allgemeine AWS-Referenz.

Starten einer DB-Instance in Amazon RDS

Wenn Sie eine Instance mit Amazon RDS starten, ist diese gänzlich unabhängig von Elastic Beanstalk und den Elastic Beanstalk-Umgebungen. Sie wird von Elastic Beanstalk weder beendet noch überwacht.

In den folgenden Schritten verwenden Sie die Amazon RDS-Konsole für Folgendes:

- Starten Sie eine Datenbank mit der MySQL-Engine.
- Aktivieren Sie eine Multi-AZ-Bereitstellung. Dadurch wird ein Standby-Modus in einer anderen Availability Zone (AZ) eingerichtet, um Datenredundanz zu gewährleisten, Blockierungen zu vermeiden I/O und Latenzspitzen bei System-Backups zu minimieren.

So starten Sie eine RDS DB-Instance in einer Standard-VPC

1. Öffnen Sie die RDS-Konsole.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.
4. Wählen Sie Standard Create (Standarderstellung).

> **Important**
>
> Wählen Sie nicht Easy Create (Einfache Erstellung) aus. Wenn Sie diese Option auswählen, können Sie die erforderlichen Einstellungen zum Starten dieser RDS-Datenbank nicht konfigurieren.

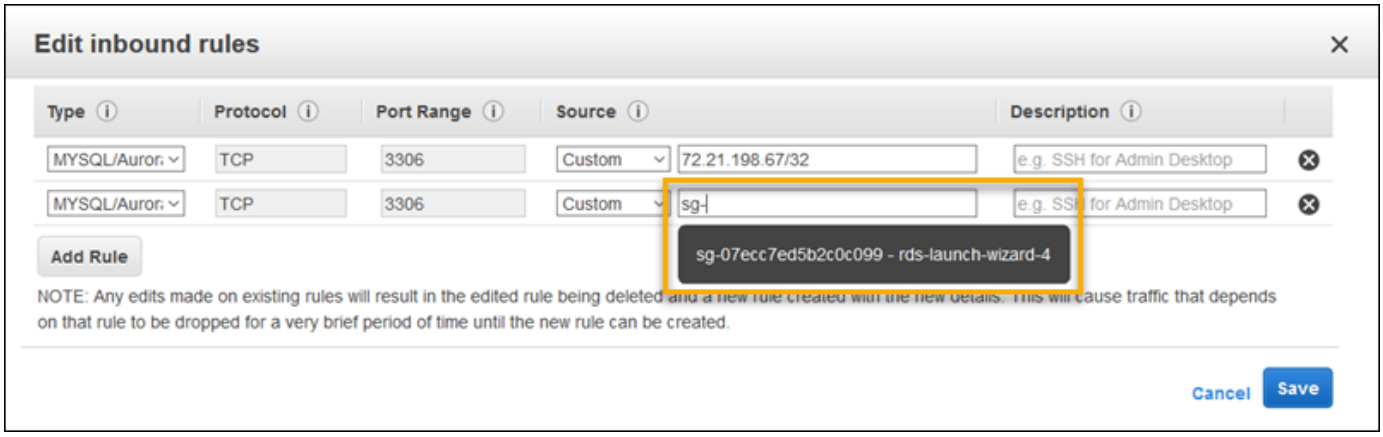
5. Geben Sie unter Additional configuration (Zusätzliche Konfiguration) für Initial database name (Anfangsdatenbankname) „**ebdb**“ ein.
6. Überprüfen Sie die Standardeinstellungen und passen Sie diese Einstellungen an Ihre spezifischen Anforderungen an. Achten Sie auf die folgenden Optionen:
   - DB instance class (DB-Instance-Klasse) – Wählen Sie eine Instance-Größe, die eine geeignete Speichermenge und CPU-Leistung für den Workload bereitstellt.
   - Multi-AZ-Bereitstellung — Für hohe Verfügbarkeit stellen Sie diese Option auf Einen Replica/ Reader Aurora-Knoten in einer anderen AZ erstellen ein.
   - Master username (Master-Benutzername) und Master password (Master-Passwort) – Der Benutzername und das Passwort für die Datenbank. Notieren Sie sich diese Einstellungen, um sie später zu verwenden.
7. Überprüfen Sie die Standardeinstellungen für die übrigen Optionen und wählen Sie dann Create database (Datenbank starten) aus.

Nachdem Ihre DB-Instance erstellt wurde, ändern Sie die ihr zugewiesene Sicherheitsgruppe, um eingehenden Datenverkehr auf dem entsprechenden Port zuzulassen.

> **Note**
>
> Es handelt sich um die gleiche Sicherheitsgruppe, die später der Elastic Beanstalk-Umgebung zugewiesen wird. Die von Ihnen jetzt hinzugefügte Regel gewährt damit Eingangsberechtigungen für andere Ressourcen in derselben Sicherheitsgruppe.

So ändern Sie die Eingangsregeln für die Sicherheitsgruppe, die an Ihre RDS-Instance angehängt ist

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance, damit die Details der Instance angezeigt werden.
4. Notieren Sie im Abschnitt Konnektivität die auf dieser Seite angezeigten Subnetze, Sicherheitsgruppen und Endpunkte. Auf diese Weise können Sie diese Informationen später verwenden.
5. In Security (Sicherheit) wird die mit der DB-Instance verknüpfte Sicherheitsgruppe angezeigt. Öffnen Sie den Link, um die Sicherheitsgruppe in der EC2 Amazon-Konsole anzuzeigen.
6. Klicken Sie in den Details der Sicherheitsgruppe auf Inbound (Eingehend).
7. Wählen Sie Edit aus.
8. Klicken Sie auf Add Rule (Regel hinzufügen).
9. Wählen Sie unter Type (Typ) die von der Anwendung verwendete DB-Engine aus.
10. Geben Sie für Source (Quelle) **`sg-`** ein, um eine Liste der verfügbaren Sicherheitsgruppen anzuzeigen. Wählen Sie die Sicherheitsgruppe aus, die mit der Auto-Scaling-Gruppe verknüpft ist, die mit Ihrer Elastic-Beanstalk-Umgebung verwendet wird. Auf diese Weise können EC2 Amazon-Instances in der Umgebung auf die Datenbank zugreifen.



11. Wählen Sie Speichern.

Das Erstellen einer DB-Instance dauert ca. 10 Minuten. Laden Sie in der Zwischenzeit Ihre Elastic Beanstalk Beanstalk-Umgebung herunter WordPress und erstellen Sie sie.

Laden Sie herunter WordPress

Um die Bereitstellung WordPress vorzubereiten AWS Elastic Beanstalk, müssen Sie die WordPress Dateien auf Ihren Computer kopieren und die richtigen Konfigurationsinformationen angeben.

Um ein WordPress Projekt zu erstellen

1. WordPress Von wordpress.org herunterladen.

```
~$curl https://wordpress.org/wordpress-6.2.tar.gz -o wordpress.tar.gz
```

2. Laden Sie die Konfigurationsdateien aus dem Beispielrepository herunter.

```
~$ wget https://github.com/aws-samples/eb-php-wordpress/releases/download/v1.1/eb-
php-wordpress-v1.zip
```

3. Extrahieren WordPress und ändern Sie den Namen des Ordners.

```
~$ tar -xvf wordpress.tar.gz
~$ mv wordpress wordpress-beanstalk
~$ cd wordpress-beanstalk
```

4. Extrahieren Sie die Konfigurationsdateien über die WordPress Installation.

```
~/wordpress-beanstalk$ unzip ../eb-php-wordpress-v1.zip
 creating: .ebextensions/
inflating: .ebextensions/dev.config
inflating: .ebextensions/efs-create.config
inflating: .ebextensions/efs-mount.config
inflating: .ebextensions/loadbalancer-sg.config
inflating: .ebextensions/wordpress.config
inflating: LICENSE
inflating: README.md
inflating: wp-config.php
```

Starten einer Elastic Beanstalk-Umgebung

Erstellen Sie mit der Elastic Beanstalk-Konsole, eine Elastic Beanstalk-Umgebung. Nachdem Sie die Umgebung gestartet haben, können Sie sie so konfigurieren, dass sie eine Verbindung zur Datenbank herstellt, und dann den WordPress Code in der Umgebung bereitstellen.

In den folgenden Schritten verwenden Sie die Elastic Beanstalk-Konsole für Folgendes:

- Erstellen Sie eine Elastic Beanstalk-Anwendung mit der verwalteten PHP-Plattform.
- Übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole über diesen vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication?Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus und wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung dauert etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt.

Elastic Beanstalk hat Ressourcen erstellt

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *`subdomain region`*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Da die Amazon RDS-Instance von Ihnen außerhalb der Umgebung gestartet wurde, müssen Sie die Verwaltung ihres Lebenszyklus übernehmen.

> **Note**
>
> Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon S3.

Konfigurieren der Sicherheitsgruppen und Umgebungseigenschaften

Fügen Sie die Sicherheitsgruppe der DB-Instance zur ausgeführten Umgebung hinzu. Dadurch stellt Elastic Beanstalk alle Instances in der Umgebung mit der zusätzlich zugeordneten Sicherheitsgruppe erneut bereit.

So fügen Sie eine Sicherheitsgruppe zur Umgebung hinzu

- Gehen Sie folgendermaßen vor:
  - So fügen Sie eine Sicherheitsgruppe mit der Elastic Beanstalk-Konsole hinzu

    a. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region

    b. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

    c. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

    d. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).

    e. Wählen Sie unter EC2 Sicherheitsgruppen die Sicherheitsgruppe aus, die an die Instances angehängt werden soll, zusätzlich zu der Instance-Sicherheitsgruppe, die Elastic Beanstalk erstellt.

    f. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

    g. Lesen Sie die Warnmeldung und klicken Sie dann auf Bestätigen.

- Fügen Sie eine Sicherheitsgruppe mit einer Konfigurationsdatei hinzufügen. Verwenden Sie die `securitygroup-addexisting.config`-Beispieldatei.

Anschließend werden die Verbindungsinformationen mithilfe von Umgebungseigenschaften an die Umgebung übergeben.

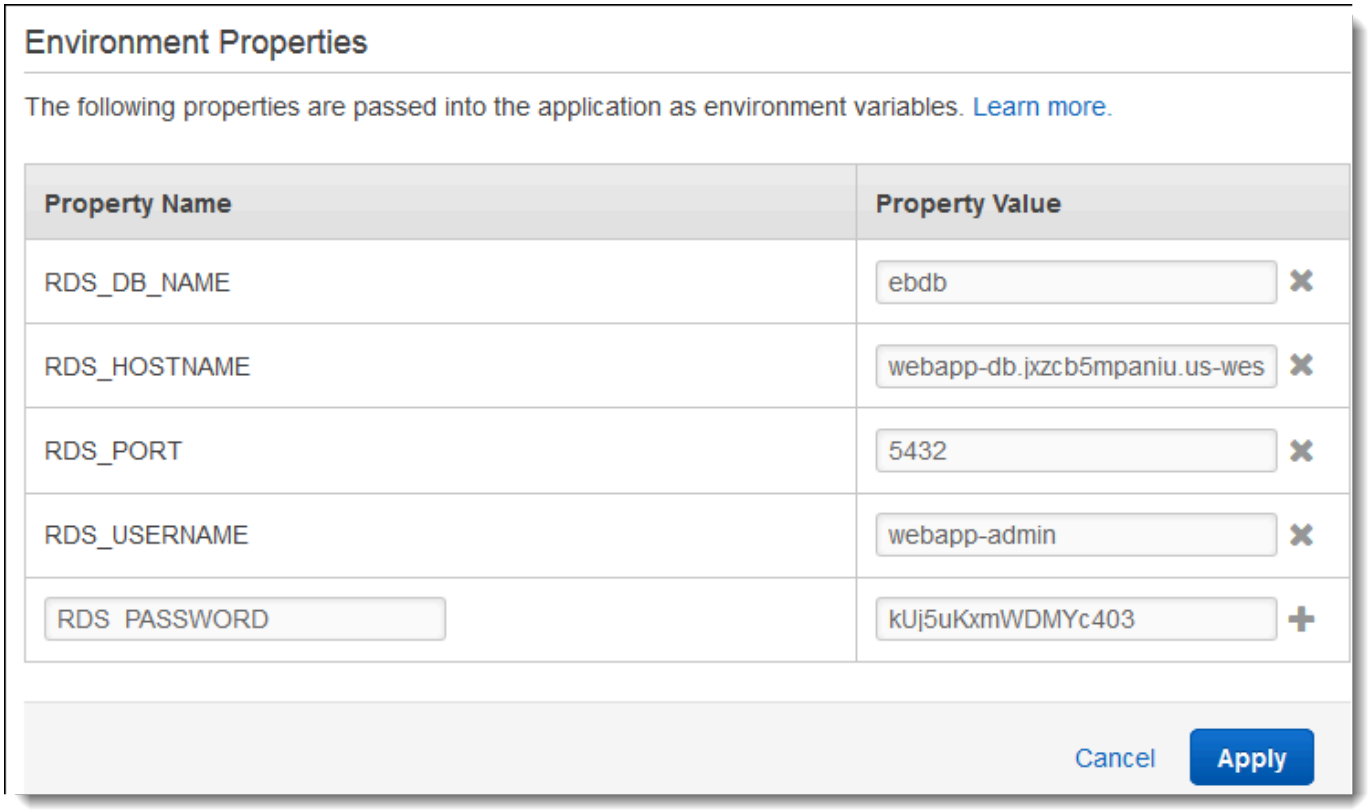
Die WordPress Anwendung verwendet einen Standardsatz von Eigenschaften, die denen entsprechen, die Elastic Beanstalk konfiguriert, wenn Sie eine Datenbank in Ihrer Umgebung bereitstellen.

So konfigurieren Sie Umgebungseigenschaften für eine DB-Instance von Amazon RDS

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Definieren Sie im Abschnitt Environment properties (Umgebungseigenschaften) die Variablen, die Ihre Anwendung liest, um eine Verbindungszeichenfolge zu bilden. Zur Sicherstellung der Kompatibilität mit Umgebungen, die eine integrierte RDS-DB-Instance haben, verwenden Sie die folgenden Namen und Werte. Sie finden alle Werte außer dem Passwort in der RDS-Konsole.

Name der Eigenschaft	Beschreibung	Eigenschaftenwert
`RDS_HOSTNAME`	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
`RDS_PORT`	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engine	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherhei

Name der Eigenschaft	Beschreibung	Eigenschaftenwert
	s haben unterschiedliche Standardwerte.	t) auf der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfigur ation) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.



6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Konfigurieren und Bereitstellen Ihrer Anwendung

Überprüfen Sie wie gezeigt, ob die Struktur des Ordners `wordpress-beanstalk` korrekt ist.

```
wordpress-beanstalk$ tree -aL 1
.
### .ebextensions
### index.php
### LICENSE
### license.txt
### readme.html
### README.md
### wp-activate.php
### wp-admin
### wp-blog-header.php
### wp-comments-post.php
### wp-config.php
```

```
### wp-config-sample.php
### wp-content
### wp-cron.php
### wp-includes
### wp-links-opml.php
### wp-load.php
### wp-login.php
### wp-mail.php
### wp-settings.php
### wp-signup.php
### wp-trackback.php
### xmlrpc.php
```

Die angepasste Datei `wp-config.php` aus dem Projektrepository verwendet die Umgebungsvariablen, die Sie im vorherigen Schritt zum Konfigurieren der Datenbankverbindung definiert haben. Der Ordner `.ebextensions` enthält Konfigurationsdateien, die weitere Ressourcen in Ihrer Elastic Beanstalk-Umgebung erstellen.

Die Konfigurationsdateien müssen für die Arbeit mit dem Konto geändert werden. Ersetzen Sie die Platzhalterwerte in den Dateien durch die entsprechenden Werte IDs und erstellen Sie ein Quellpaket.

So aktualisieren Sie die Konfigurationsdateien und erstellen ein Quell-Bundle

1. Ändern Sie die Konfigurationsdateien wie folgt.

   - `.ebextensions/dev.config`— Schränkt den Zugriff auf Ihre Umgebung ein, um sie während des WordPress Installationsvorgangs zu schützen. Ersetzen Sie die Platzhalter-IP-Adresse oben in der Datei durch die öffentliche IP-Adresse des Computers, mit dem Sie auf die Website Ihrer Umgebung zugreifen und Ihre WordPress Installation abschließen möchten.

     > **Note**
     >
     > Abhängig von Ihrem Netzwerk benötigen Sie möglicherweise einen IP-Adressblock.

   - `.ebextensions/efs-create.config`— Erstellt ein EFS-Dateisystem und Bereitstellungspunkte in jeder Availability Zone/subnet in Ihrer VPC. Identifizieren Sie Ihre Standard-VPC und Ihr Subnetz IDs in der Amazon VPC-Konsole.

2. Erstellen Sie ein Quell-Bundle mit den Dateien im Projektordner. Der folgende Befehl erstellt ein Quell-Bundle mit dem Namen `wordpress-beanstalk.zip`.

```
~/eb-wordpress$ zip ../wordpress-beanstalk.zip -r * .[^.]*
```

Laden Sie das Quellpaket auf Elastic Beanstalk hoch, um es in Ihrer WordPress Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Installieren Sie WordPress

Um Ihre WordPress Installation abzuschließen

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie die Umgebungs-URL aus, um die Website in einem Browser zu öffnen. Sie werden zu einem WordPress Installationsassistenten weitergeleitet, da Sie die Site noch nicht konfiguriert haben.
4. Führen Sie eine Standardinstallation aus. Die Datei `wp-config.php` ist bereits im Quellcode vorhanden und so konfiguriert, dass sie die Informationen für die Datenbankverbindung aus der Umgebung ausliest. Sie sollten nicht zur Verbindungskonfiguration aufgefordert werden.

Die Installation dauert etwa eine Minute.

Aktualisieren von Schlüsseln und Salts

Die WordPress Konfigurationsdatei liest `wp-config.php` auch Werte für Schlüssel und Salze aus den Umgebungseigenschaften. Zurzeit werden diese Eigenschaften alle von der Datei `test` im Ordner `wordpress.config` auf den Wert `.ebextensions` festgelegt.

Der Hash-Salt kann jeder Wert sein, der die Anforderungen an Umgebungseigenschaften erfüllt, Sie sollten ihn jedoch nicht in der Versionskontrolle speichern. Legen Sie diese Eigenschaften mithilfe der Elastic Beanstalk-Konsole direkt in der Umgebung fest.

So aktualisieren Sie die Umgebungseigenschaften

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie unter Software die Option Edit (Bearbeiten).
5. Ändern Sie für `Environment properties` die folgenden Eigenschaften:
   - `AUTH_KEY` – Der gewählte Wert für `AUTH_KEY`.
   - `SECURE_AUTH_KEY` – Der gewählte Wert für `SECURE_AUTH_KEY`.
   - `LOGGED_IN_KEY` – Der gewählte Wert für `LOGGED_IN_KEY`.
   - `NONCE_KEY` – Der gewählte Wert für `NONCE_KEY`.
   - `AUTH_SALT` – Der gewählte Wert für `AUTH_SALT`.
   - `SECURE_AUTH_SALT` – Der gewählte Wert für `SECURE_AUTH_SALT`.
   - `LOGGED_IN_SALT` – Der gewählte Wert für `LOGGED_IN_SALT`.
   - `NONCE_SALT` – Der gewählte Wert für `NONCE_SALT`.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

> **Note**
>
> Wenn Sie die Eigenschaften direkt in der Umgebung festlegen, werden die Werte in `wordpress.config`überschrieben.

Entfernen von Zugriffsbeschränkungen

Das Beispielprojekt enthält die Konfigurationsdatei `loadbalancer-sg.config`. Es erstellt eine Sicherheitsgruppe und weist sie dem Load Balancer der Umgebung unter Verwendung der IP-Adresse zu, die Sie in `dev.config` konfiguriert haben. Sie schränkt den HTTP-Zugriff auf Port 80 auf die Verbindungen in Ihrem Netzwerk ein. Andernfalls könnte eine externe Partei möglicherweise eine Verbindung zu Ihrer Site herstellen, bevor Sie Ihr Administratorkonto installiert WordPress und konfiguriert haben.

Nachdem Sie die Installation abgeschlossen haben WordPress, entfernen Sie die Konfigurationsdatei, um die Site der ganzen Welt zugänglich zu machen.

So entfernen Sie die Beschränkung und aktualisieren die Umgebung

1. Löschen Sie die Datei `.ebextensions/loadbalancer-sg.config` aus dem Projektverzeichnis.

   ```
   ~/wordpress-beanstalk$ rm .ebextensions/loadbalancer-sg.config
   ```

2. Erstellen Sie ein Quell-Bundle.

   ```
   ~/eb-wordpress$ zip ../wordpress-beanstalk-v2.zip -r * .[^.]*
   ```

Laden Sie das Quellpaket auf Elastic Beanstalk hoch, um es in Ihrer WordPress Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.

6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Konfigurieren der Auto-Scaling-Gruppe

Konfigurieren Sie abschließend für die Auto-Scaling-Gruppe der Umgebung eine höhere Instance-Mindestanzahl. Führen Sie stets mindestens zwei Instances aus, um zu verhindern, dass die Webserver in Ihrer Umgebung eine einzelne Fehlerquelle darstellen. Dadurch können Sie Änderungen bereitstellen, ohne Ihre Website außer Betrieb zu nehmen.

So konfigurieren Sie die Auto-Scaling-Gruppe in der Umgebung für hohe Verfügbarkeit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Legen Sie im Abschnitt Auto Scaling Group (Gruppe der automatischen Skalierung) für die Option Min instances (Min. Instances) **2** fest.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Beispielprojekt erstellt mit Amazon EFS ein freigegebenes Dateisystem und ermöglicht das Hochladen von Inhalten über mehrere Instances hinweg. Erstellen Sie einen Beitrag auf der Website und laden Sie Inhalte hoch, die im freigegebenen Dateisystem gespeichert werden sollen. Zeigen Sie den Beitrag an und aktualisieren Sie die Seite mehrfach, damit Instances genutzt werden und zugleich verifiziert wird, dass das freigegebene Dateisystem funktioniert.

Upgraden WordPress

Um auf eine neue Version von zu aktualisieren WordPress, sichern Sie Ihre Site und stellen Sie sie in einer neuen Umgebung bereit.

> **Important**
>
> Verwenden Sie nicht die darin enthaltene Aktualisierungsfunktion und aktualisieren Sie Ihre Quelldateien nicht, um eine neue Version zu verwenden. WordPress Beide Aktionen können dazu führen, dass Ihr Beitrag 404-Fehler URLs zurückgibt, obwohl sie sich noch in der Datenbank und im Dateisystem befinden.

Um ein Upgrade durchzuführen WordPress

1. Verwenden Sie in der WordPress Admin-Konsole das Export-Tool, um Ihre Beiträge in eine XML-Datei zu exportieren.
2. Stellen Sie die neue Version von WordPress to Elastic Beanstalk bereit und installieren Sie sie. Gehen Sie dabei genauso vor wie bei der Installation der vorherigen Version. Wenn Sie eine Umgebung mit der neuen Version erstellen, können Sie Ausfallzeiten vermeiden.
3. Installieren Sie in der neuen Version das WordPress Importer-Tool in der Admin-Konsole und importieren Sie damit die XML-Datei mit Ihren Posts. Wenn die Beiträge vom Benutzer "admin" mit der alten Version erstellt wurden, weisen Sie sie dem Benutzer "admin" der neuen Website zu, anstatt den Benutzer "admin" zu importieren.
4. Wenn Sie die neue Version in einer separaten Umgebung bereitgestellt haben, führen Sie einen CNAME-Austausch durch, um Benutzer von der alten zur neuen Site umzuleiten.

Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung von der Konsole aus

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb der Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie Ihre DB-Instance aus.
4. Wählen Sie Aktionen und anschließend Löschen aus.
5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Im weiteren Verlauf der Anwendungsentwicklung können Sie Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die Elastic Beanstalk Command Line Interface (EB CLI) bietet easy-to-use Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk Beanstalk-Umgebungen von der Befehlszeile aus.

In der Beispielanwendung werden Konfigurationsdateien verwendet, um PHP-Einstellungen zu konfigurieren und eine Tabelle in der Datenbank zu erstellen (sofern nicht bereits vorhanden). Mit einer Konfigurationsdatei können Sie zudem während des Erstellens der Umgebung die Sicherheitsgruppeneinstellungen der Instance konfigurieren und so zeitaufwendige Aktualisierungen der Konfiguration verhindern. Weitere Informationen finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

Für die Entwicklung sowie für Tests können Sie die Elastic Beanstalk-Funktionalität nutzen und eine verwaltete DB-Instance direkt in der Umgebung hinzufügen. Detaillierte Anweisungen zum Einrichten einer Datenbank in der Umgebung finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Wenn Sie eine Datenbank mit hoher Verfügbarkeit benötigen, können Sie Amazon Aurora verwenden. Amazon Aurora ist eine mit MySQL kompatible Datenbank-Engine, die kommerzielle Datenbank-Features bei geringen Kosten bietet. Um die Verbindung der Anwendung mit einer anderen Datenbank herzustellen, wiederholen Sie die Schritte der Sicherheitsgruppenkonfiguration und aktualisieren Sie die RDS-bezogenen Umgebungseigenschaften.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, konfigurieren Sie einen benutzerdefinierten Domainnamen für die Umgebung und aktivieren dann HTTPS für sichere Verbindungen.

Bereitstellen einer Drupal-Website mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank für Elastic Beanstalk

Dieses Tutorial führt Sie durch den Prozess des Startens einer externen RDS-DB-Instance AWS Elastic Beanstalk. Anschließend wird beschrieben, wie Sie eine hochverfügbare Umgebung konfigurieren, in der eine Drupal-Website ausgeführt wird, um eine Verbindung zu ihr herzustellen. Die Website verwendet Amazon Elastic File System (Amazon EFS) als gemeinsamen Speicher für hochgeladene Dateien. Wenn Sie eine DB-Instance außerhalb von Elastic Beanstalk ausführen, wird die Datenbank vom Lebenszyklus Ihrer Umgebung entkoppelt, sodass Sie von mehreren Umgebungen aus eine Verbindung zu derselben Datenbank herstellen, eine Datenbank gegen eine andere austauschen oder ein blue/green Deployment durchführen können, ohne dass Ihre Datenbank beeinträchtigt wird.

Sections

- Voraussetzungen
- Starten einer DB-Instance in Amazon RDS
- Starten einer Elastic Beanstalk-Umgebung
- Konfigurieren von Sicherheitseinstellungen und Umgebungseigenschaften
- Konfigurieren und Bereitstellen Ihrer Anwendung
- Installation von Drupal
- Aktualisieren der Drupal-Konfiguration und Entfernen von Zugriffsbeschränkungen
- Konfigurieren der Auto-Scaling-Gruppe
- Bereinigen
- Nächste Schritte

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole kennen. Sofern noch nicht geschehen, starten Sie anhand der Anleitung in Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können die erste Elastic-Beanstalk-Umgebung.

Für die Verfahren in diesem Leitfaden benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

Die Verfahren in diesem Tutorial für Amazon Relational Database Service (Amazon RDS)-Aufgaben gehen davon aus, dass Sie Ressourcen in einer standardmäßigen Amazon Virtual Private Cloud (Amazon VPC) starten. Alle neuen Konten umfassen eine Standard-VPC in jeder Region. Wenn Sie nicht über eine Standard-VPC verfügen, sind die Verfahren unterschiedlich. Anweisungen Verwenden von Elastic Beanstalk mit Amazon RDS für EC2 -Classic- und benutzerdefinierte VPC-Plattformen finden Sie unter.

Die Beispielanwendung verwendet Amazon EFS. Es funktioniert nur in AWS Regionen, die Amazon EFS unterstützen. Weitere Informationen zur Unterstützung von AWS Regionen finden Sie unter Amazon Elastic File System Endpoints and Quotas in der Allgemeine AWS-Referenz.

Wenn die Plattform Ihrer Elastic Beanstalk-Umgebung PHP 7.4 oder älter verwendet, empfehlen wir Ihnen, die Drupal-Version 8.9.13 für dieses Tutorial zu verwenden. Für Plattformen, die mit PHP 8.0 oder höher installiert werden, empfehlen wir die Verwendung von Drupal 9.1.5.

Weitere Informationen zu Drupal-Versionen und den unterstützten PHP-Versionen finden Sie unter PHP-Anforderungen auf der Drupal-Website. Die von Drupal empfohlenen Kernversionen sind auf der Website https://www.drupal aufgeführt. org/project/drupal.

Starten einer DB-Instance in Amazon RDS

Um eine externe Datenbank mit einer Anwendung zu verwenden, die in Elastic Beanstalk ausgeführt wird, starten Sie zuerst eine DB-Instance mit Amazon RDS. Wenn Sie eine Instance mit Amazon RDS starten, ist diese gänzlich unabhängig von Elastic Beanstalk und den Elastic Beanstalk-Umgebungen. Sie wird von Elastic Beanstalk weder beendet noch überwacht.

Verwenden Sie die Amazon RDS-Konsole, um eine DB-Instance mit MySQL und Multi-AZ-Bereitstellung zu starten. Eine Multi-AZ-Bereitstellung gewährleistet die Ausführung von Failovern für Ihre Datenbank, damit sie auch bei einem Ausfall der DB-Quell-Instance verfügbar ist.

So starten Sie eine RDS DB-Instance in einer Standard-VPC

1. Öffnen Sie die RDS-Konsole.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.
4. Wählen Sie Standard Create (Standarderstellung).

   > **Important**
   >
   > Wählen Sie nicht Easy Create (Einfache Erstellung) aus. Wenn Sie diese Option auswählen, können Sie die erforderlichen Einstellungen zum Starten dieser RDS-Datenbank nicht konfigurieren.

5. Geben Sie unter Additional configuration (Zusätzliche Konfiguration) für Initial database name (Anfangsdatenbankname) „**ebdb**“ ein.
6. Überprüfen Sie die Standardeinstellungen und passen Sie diese Einstellungen an Ihre spezifischen Anforderungen an. Achten Sie auf die folgenden Optionen:

   - DB instance class (DB-Instance-Klasse) – Wählen Sie eine Instance-Größe, die eine geeignete Speichermenge und CPU-Leistung für den Workload bereitstellt.
   - Multi-AZ-Bereitstellung — Für hohe Verfügbarkeit stellen Sie diese Option auf Einen Replica/ Reader Aurora-Knoten in einer anderen AZ erstellen ein.
   - Master username (Master-Benutzername) und Master password (Master-Passwort) – Der Benutzername und das Passwort für die Datenbank. Notieren Sie sich diese Einstellungen, um sie später zu verwenden.

7. Überprüfen Sie die Standardeinstellungen für die übrigen Optionen und wählen Sie dann Create database (Datenbank starten) aus.

Bearbeiten Sie anschließend die Sicherheitsgruppe für die DB-Instance, damit eingehender Datenverkehr über den entsprechenden Port zulässig ist. Es handelt sich um die gleiche Sicherheitsgruppe, die später mit der Elastic Beanstalk-Umgebung verknüpft wird. Die von Ihnen hinzugefügte Regel gewährt damit Eingangsberechtigungen für andere Ressourcen in derselben Sicherheitsgruppe.

So ändern Sie die Eingangsregeln für die Sicherheitsgruppe, die an Ihre RDS-Instance angehängt ist

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance, damit die Details der Instance angezeigt werden.
4. Notieren Sie im Abschnitt Konnektivität die auf dieser Seite angezeigten Subnetze, Sicherheitsgruppen und Endpunkte. Auf diese Weise können Sie diese Informationen später verwenden.
5. In Security (Sicherheit) wird die mit der DB-Instance verknüpfte Sicherheitsgruppe angezeigt. Öffnen Sie den Link, um die Sicherheitsgruppe in der EC2 Amazon-Konsole anzuzeigen.
6. Klicken Sie in den Details der Sicherheitsgruppe auf Inbound (Eingehend).
7. Wählen Sie Edit aus.
8. Klicken Sie auf Add Rule (Regel hinzufügen).
9. Wählen Sie unter Type (Typ) die von der Anwendung verwendete DB-Engine aus.
10. Geben Sie für Source (Quelle) **`sg-`** ein, um eine Liste der verfügbaren Sicherheitsgruppen anzuzeigen. Wählen Sie die Sicherheitsgruppe aus, die mit der Auto-Scaling-Gruppe verknüpft ist, die mit Ihrer Elastic-Beanstalk-Umgebung verwendet wird. Auf diese Weise können EC2 Amazon-Instances in der Umgebung auf die Datenbank zugreifen.



11. Wählen Sie Speichern.

Das Erstellen einer DB-Instance dauert ca. 10 Minuten. Starten Sie unterdessen die Elastic Beanstalk-Umgebung.

Starten einer Elastic Beanstalk-Umgebung

Erstellen Sie mit der Elastic Beanstalk-Konsole, eine Elastic Beanstalk-Umgebung. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode. Nachdem Sie die Umgebung gestartet haben, können Sie die Umgebung für die Verbindung zur Datenbank konfigurieren und danach den Drupal-Code in der Umgebung bereitstellen.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole über diesen vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication?Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus und wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.
- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation Stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet. Die RDS-DB-Instance wurde von Ihnen außerhalb der Umgebung gestartet, daher müssen Sie die Verwaltung ihres Lebenszyklus übernehmen.

> **Note**
>
> Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon S3.

Konfigurieren von Sicherheitseinstellungen und Umgebungseigenschaften

Fügen Sie die Sicherheitsgruppe der DB-Instance zur ausgeführten Umgebung hinzu. Dadurch stellt Elastic Beanstalk alle Instances in der Umgebung mit der zusätzlich zugeordneten Sicherheitsgruppe erneut bereit.

So fügen Sie eine Sicherheitsgruppe zur Umgebung hinzu

- Gehen Sie folgendermaßen vor:
  - So fügen Sie eine Sicherheitsgruppe mit der Elastic Beanstalk-Konsole hinzu
    a. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region

b. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

c. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

d. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).

e. Wählen Sie unter EC2 Sicherheitsgruppen die Sicherheitsgruppe aus, die an die Instances angehängt werden soll, zusätzlich zu der Instance-Sicherheitsgruppe, die Elastic Beanstalk erstellt.

f. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

g. Lesen Sie die Warnmeldung und klicken Sie dann auf Bestätigen.

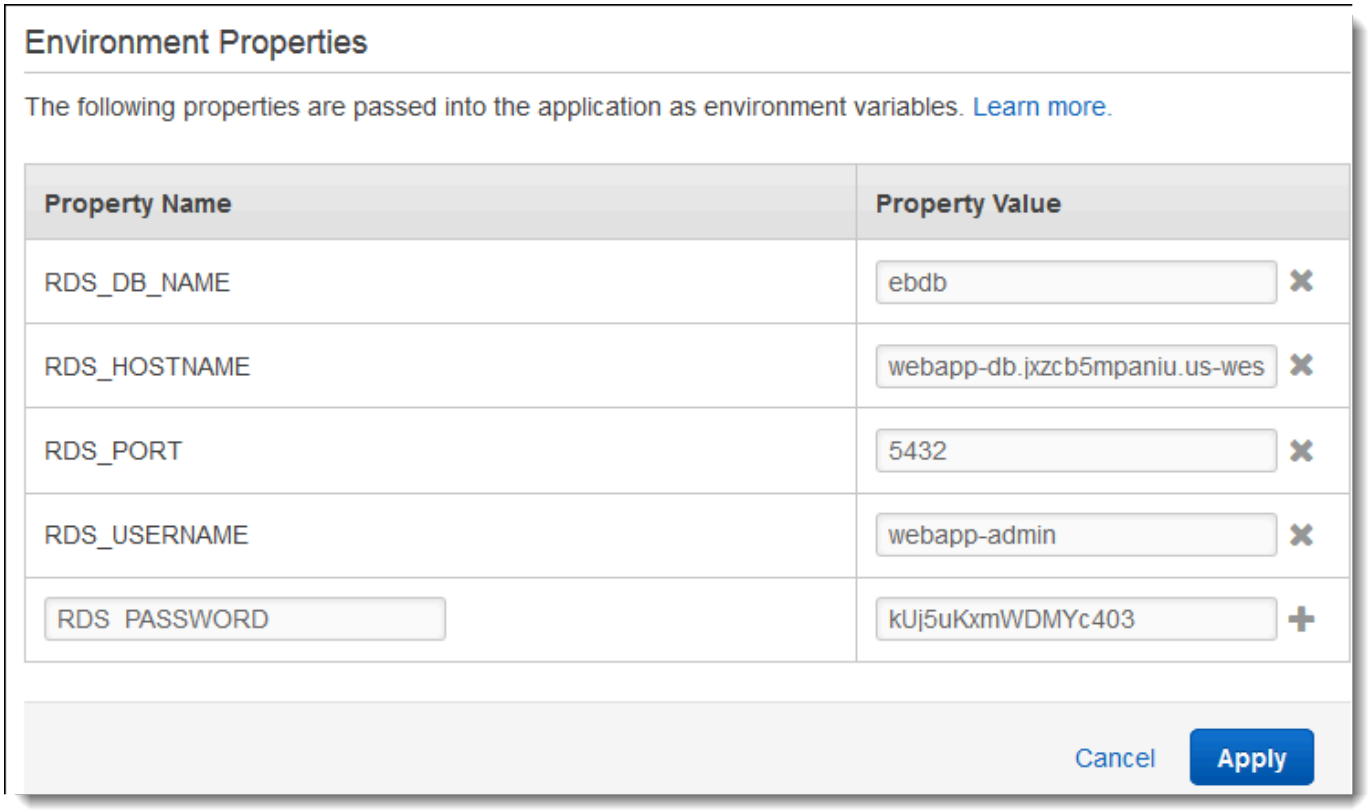
- Fügen Sie eine Sicherheitsgruppe mit einer Konfigurationsdatei hinzufügen. Verwenden Sie die `securitygroup-addexisting.config`-Beispieldatei.

Anschließend werden die Verbindungsinformationen mithilfe von Umgebungseigenschaften an die Umgebung übergeben. Die Beispielanwendung hat Standardeigenschaften, die den Eigenschaften entsprechen, die von Elastic Beanstalk im Rahmen der Datenbankbereitstellung in der Umgebung konfiguriert werden.

So konfigurieren Sie Umgebungseigenschaften für eine DB-Instance von Amazon RDS

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Definieren Sie im Abschnitt Environment properties (Umgebungseigenschaften) die Variablen, die Ihre Anwendung liest, um eine Verbindungszeichenfolge zu bilden. Zur Sicherstellung der Kompatibilität mit Umgebungen, die eine integrierte RDS-DB-Instance haben, verwenden Sie die folgenden Namen und Werte. Sie finden alle Werte außer dem Passwort in der RDS-Konsole.

Name der Eigenschaft	Beschreibung	Eigenschaftenwert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.



6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Nach dem Installieren von Drupal müssen Sie über SSH eine Verbindung zur Instance herstellen, um einige Konfigurationsdaten abzurufen. Weisen Sie den Instances der Umgebung einen SSH-Schlüssel zu.

So konfigurieren Sie SSH

1. Wenn Sie noch kein key pair erstellt haben, öffnen Sie die Schlüsselpaarseite der EC2 Amazon-Konsole und folgen Sie den Anweisungen, um eines zu erstellen.
2. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
3. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
4. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
5. Wählen Sie unter Security (Sicherheit) die Option Edit (Bearbeiten) aus.

6. Wählen Sie für EC2 key pair Ihr key pair aus.
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Konfigurieren und Bereitstellen Ihrer Anwendung

Um ein Drupal-Projekt für Elastic Beanstalk zu erstellen, laden Sie den Drupal-Quellcode herunter und kombinieren Sie ihn mit den Dateien im aws-samples/ Repository auf. eb-php-drupal GitHub

So erstellen Sie ein Drupal-Projekt

1. Laden Sie Drupal mit dem folgenden Befehl unter www.drupal.org/download herunter. Weitere Informationen zu Downloads finden Sie auf der Drupal-Website.

   Wenn die Plattform Ihrer Elastic Beanstalk-Umgebung PHP 7.4 oder älter verwendet, empfehlen wir Ihnen, die Drupal-Version 8.9.13 für dieses Tutorial herunterzuladen. Führen Sie zum Herunterladen den folgenden Befehl aus.

   ```
   ~$ curl https://ftp.drupal.org/files/projects/drupal-8.9.13.tar.gz -o drupal.tar.gz
   ```

   Wenn Ihre Plattform PHP 8.0 oder höher verwendet, empfehlen wir Ihnen, Drupal 9.1.5 herunterzuladen. Zum Herunterladen können Sie den folgenden Befehl verwenden.

   ```
   ~$ curl https://ftp.drupal.org/files/projects/drupal-9.1.5.tar.gz -o drupal.tar.gz
   ```

   Weitere Informationen zu Drupal-Versionen und den unterstützten PHP-Versionen finden Sie unter PHP-Anforderungen in der offiziellen Drupal-Dokumentation. Die von Drupal empfohlenen Core-Versionen sind auf der Drupal-Website aufgeführt.
2. Laden Sie die Konfigurationsdateien mit folgendem Befehl aus dem Beispiel-Repository herunter:

   ```
   ~$ wget https://github.com/aws-samples/eb-php-drupal/releases/download/v1.1/eb-php-drupal-v1.zip
   ```
3. Extrahieren Sie Drupal und ändern Sie den Namen des Ordners.

   Wenn Sie Drupal 8.9.13 heruntergeladen haben:

   ```
   ~$ tar -xvf drupal.tar.gz
   ```

```
~$ mv drupal-8.9.13 drupal-beanstalk
~$ cd drupal-beanstalk
```

Wenn Sie Drupal 9.1.5 heruntergeladen haben:

```
~$ tar -xvf drupal.tar.gz
~$ mv drupal-9.1.5 drupal-beanstalk
~$ cd drupal-beanstalk
```

4. Extrahieren während der Drupal-Installation die Konfigurationsdateien.

```
~/drupal-beanstalk$ unzip ../eb-php-drupal-v1.zip
 creating: .ebextensions/
 inflating: .ebextensions/dev.config
 inflating: .ebextensions/drupal.config
 inflating: .ebextensions/efs-create.config
 inflating: .ebextensions/efs-filesystem.template
 inflating: .ebextensions/efs-mount.config
 inflating: .ebextensions/loadbalancer-sg.config
 inflating: LICENSE
 inflating: README.md
 inflating: beanstalk-settings.php
```

Überprüfen Sie wie gezeigt, ob die Struktur des Ordners `drupal-beanstalk` korrekt ist.

```
drupal-beanstalk$ tree -aL 1
.
### autoload.php
### beanstalk-settings.php
### composer.json
### composer.lock
### core
### .csslintrc
### .ebextensions
### .ebextensions
### .editorconfig
### .eslintignore
### .eslintrc.json
### example.gitignore
### .gitattributes
### .htaccess
```

```
### .ht.router.php
### index.php
### LICENSE
### LICENSE.txt
### modules
### profiles
### README.md
### README.txt
### robots.txt
### sites
### themes
### update.php
### vendor
### web.config
```

Die Datei `beanstalk-settings.php` aus dem Projektrepository verwendet die Umgebungsvariablen, die Sie im vorherigen Schritt zum Konfigurieren der Datenbankverbindung definiert haben. Der Ordner `.ebextensions` enthält Konfigurationsdateien, die weitere Ressourcen in Ihrer Elastic Beanstalk-Umgebung erstellen.

Die Konfigurationsdateien müssen für die Arbeit mit dem Konto geändert werden. Ersetzen Sie die Platzhalterwerte in den Dateien durch die entsprechenden Werte und erstellen Sie ein Quellpaket. IDs

Zum Aktualisieren der Konfigurationsdateien und Erstellen eines Quell-Bundles.

1. Ändern Sie die Konfigurationsdateien wie folgt.

   - `.ebextensions/dev.config` – Beschränkt den Zugriff auf die Umgebung auf Ihre IP-Adresse, um die Umgebung während der Drupal-Installation zu schützen. Geben Sie anstelle des Platzhalters für die IP-Adresse oben in der Datei Ihre öffentliche IP-Adresse ein.
   - `.ebextensions/efs-create.config` – Erstellt ein EFS-Dateisystem und Mountingpunkte in allen Availability Zones/Subnetzen der VPC. Identifizieren Sie Ihre Standard-VPC und Ihr Standard-Subnetz IDs in der Amazon VPC-Konsole.

2. Erstellen Sie ein Quell-Bundle mit den Dateien im Projektordner. Der folgende Befehl erstellt ein Quell-Bundle mit dem Namen `drupal-beanstalk.zip`. Die Dateien im Ordner `vendor` werden ausgeschlossen, weil sie viel Platz benötigen und zum Bereitstellen der Anwendung für Elastic Beanstalk nicht erforderlich sind.

   ```
   ~/eb-drupal$ zip ../drupal-beanstalk.zip -r * .[^.]* -x "vendor/*"
   ```

Laden Sie das Quell-Bundle zu Elastic Beanstalk hoch, um Drupal in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Installation von Drupal

So installieren Sie Drupal

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie die Umgebungs-URL aus, um die Website in einem Browser zu öffnen. Sie werden zu einem Drupal-Installationsassistenten umgeleitet, da die Website noch nicht konfiguriert ist.
4. Führen Sie eine Standardinstallation mit den folgenden Datenbank-Einstellungen durch:

   - Datenbankname – Der DB Name (DB-Name), der in der Amazon RDS-Konsole angezeigt wird.
   - Benutzername und Passwort der Datenbank – Die Werte für Master Username (Master-Benutzername) und Master Password (Master-Passwort), die Sie beim Erstellen Ihrer Datenbank eingegeben haben.
   - Erweiterte Optionen > Host – Der Endpoint (Endpunkt) der DB-Instance, die in der Amazon RDS-Konsole angezeigt wird.

Die Installation dauert etwa eine Minute.

Aktualisieren der Drupal-Konfiguration und Entfernen von Zugriffsbeschränkungen

Im Rahmen der Drupal-Installation wurde eine Datei namens `settings.php` im `sites/default`-Ordner in der Instance erstellt. Sie benötigen diese Datei im Quellcode, damit Ihre Website bei späteren Bereitstellungen nicht zurückgesetzt wird. Derzeit enthält die Datei jedoch vertrauliche Informationen und sollte deshalb nicht in die Quelle geschrieben werden. Stellen Sie eine Verbindung zur Anwendungs-Instance her, damit die Daten aus der Einstellungsdatei abgerufen werden.

So verbinden Sie die Anwendungs-Instance mit SSH

1. Öffnen Sie die Instance-Seite der EC2 Amazon-Konsole.
2. Wählen Sie die Anwendungs-Instance. Es handelt sich um die Instance mit dem Namen der Elastic Beanstalk-Umgebung.
3. Wählen Sie Connect aus.
4. Befolgen Sie die Anleitung, um die Instance mit SSH zu verbinden. Der Befehl sieht so ähnlich aus wie dieser.

```
$ ssh -i ~/.ssh/mykey ec2-user@ec2-00-55-33-222.us-west-2.compute.amazonaws.com
```

Entnehmen Sie der letzten Zeile der Einstellungsdatei die ID des Synchronisierungsverzeichnisses.

```
[ec2-user ~]$ tail -n 1 /var/app/current/sites/default/settings.php
$config_directories['sync'] = 'sites/default/files/
config_4ccfX2sPQm79p1mk5IbUq9S_FokcENO4mxyC-L18-4g_xKj_7j9ydn31kDOYOgnzMu071Tvc4Q/
sync';
```

Die Datei enthält außerdem den aktuellen Hash-Schlüssel der Website. Sie können den aktuellen Wert aber ignorieren und einen eigenen Wert verwenden.

Weisen Sie den Umgebungseigenschaften ein Sync-Verzeichnispfad und den Hash-Schlüssel zu. Die benutzerdefinierte Einstellungsdatei aus dem Projektrepository liest diese Eigenschaften, um die Website im Rahmen der Bereitstellung sowie die zuvor eingerichteten Eigenschaften der Datenbankverbindung zu konfigurieren.

Eigenschaften der Drupal-Konfiguration

- `SYNC_DIR` – Der Pfad zum Sync-Verzeichnis.

- `HASH_SALT` – Jeder Zeichenfolgenwert, der die Anforderungen an Umgebungseigenschaften erfüllt.

So konfigurieren Sie Umgebungsvariablen in der Elastic Beanstalk Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Scrollen Sie nach unten zu Runtime-Umgebungsvariablen.
6. Wählen Sie Umgebungsvariable hinzufügen aus.
7. Wählen Sie als Quelle die Option Klartext aus.

> **Note**
>
> Die Werte Secrets Manager und SSM Parameter Store in der Drop-down-Liste dienen der Konfiguration von Umgebungsvariablen als Secrets zum Speichern sensibler Daten wie Anmeldeinformationen und API-Schlüsseln. Weitere Informationen finden Sie unter Elastic Beanstalk mit AWS Secrets Manager und AWS Systems Manager Parameter Store verwenden.

8. Geben Sie den Namen der Umgebungsvariablen und die Wertepaare der Umgebungsvariablen ein.
9. Wenn Sie weitere Variablen hinzufügen müssen, wiederholen Sie Schritt 6 bis Schritt 8.
10. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Schließlich enthält das Beispielprojekt eine Konfigurationsdatei (`loadbalancer-sg.config`), mit der eine Sicherheitsgruppe erstellt und dem Load Balancer der Umgebung zugeordnet wird. Dabei wird die IP-Adresse verwendet, die Sie in der Datei `dev.config` konfiguriert haben, um den HTTP-Zugriff über Port 80 auf Verbindungen in Ihrem Netzwerk zu beschränken. Andernfalls könnte eine

externe Partei potenziell eine Verbindung zur Website herstellen, bevor Sie Drupal installiert und das Administratorkonto konfiguriert haben.

So aktualisieren Sie die Drupal-Konfiguration und entfernen Zugriffsbeschränkungen

1. Löschen Sie die Datei `.ebextensions/loadbalancer-sg.config` aus dem Projektverzeichnis.

   ```
   ~/drupal-beanstalk$ rm .ebextensions/loadbalancer-sg.config
   ```

2. Kopieren Sie die benutzerdefinierte Datei `settings.php` in den Ordner der Website.

   ```
   ~/drupal-beanstalk$ cp beanstalk-settings.php sites/default/settings.php
   ```

3. Erstellen Sie ein Quell-Bundle.

   ```
   ~/eb-drupal$ zip ../drupal-beanstalk-v2.zip -r * .[^.]* -x "vendor/*"
   ```

Laden Sie das Quell-Bundle zu Elastic Beanstalk hoch, um Drupal in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Konfigurieren der Auto-Scaling-Gruppe

Konfigurieren Sie abschließend für die Auto-Scaling-Gruppe der Umgebung eine höhere Instance-Mindestanzahl. Führen Sie immer mindestens zwei Instances aus, damit einzelne Webserver in der

Umgebung keine Fehlerquelle darstellen und damit Sie Änderungen ohne Ausfallzeit der Website bereitstellen können.

So konfigurieren Sie die Auto-Scaling-Gruppe in der Umgebung für hohe Verfügbarkeit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Legen Sie im Abschnitt Auto Scaling Group (Gruppe der automatischen Skalierung) für die Option Min instances (Min. Instances) **2** fest.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Beispielprojekt erstellt mit Amazon Elastic File System ein freigegebenes Dateisystem für das Hochladen von Inhalten über mehrere Instances hinweg. Erstellen Sie einen Beitrag auf der Website und laden Sie Inhalte hoch, die im freigegebenen Dateisystem gespeichert werden sollen. Zeigen Sie den Beitrag an und aktualisieren Sie die Seite mehrfach, damit Instances genutzt werden und zugleich verifiziert wird, dass das freigegebene Dateisystem funktioniert.

Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung über die Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb der Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie Ihre DB-Instance aus.
4. Wählen Sie Aktionen und anschließend Löschen aus.
5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Im weiteren Verlauf der Anwendungsentwicklung können Sie Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die Elastic Beanstalk Command Line Interface (EB CLI) bietet easy-to-use Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk Beanstalk-Umgebungen von der Befehlszeile aus.

In der Beispielanwendung werden Konfigurationsdateien verwendet, um PHP-Einstellungen zu konfigurieren und eine Tabelle in der Datenbank zu erstellen (sofern nicht bereits vorhanden). Mit einer Konfigurationsdatei können Sie zudem während des Erstellens der Umgebung die Sicherheitsgruppeneinstellungen der Instance konfigurieren und so zeitaufwendige Konfigurationsaktualisierungen verhindern. Weitere Informationen finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

Für die Entwicklung sowie für Tests können Sie die Elastic Beanstalk-Funktionalität nutzen und eine verwaltete DB-Instance direkt in der Umgebung hinzufügen. Detaillierte Anweisungen zum Einrichten einer Datenbank in der Umgebung finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Wenn Sie eine Datenbank mit hoher Verfügbarkeit benötigen, können Sie Amazon Aurora verwenden. Amazon Aurora ist eine mit MySQL kompatible Datenbank-Engine, die kommerzielle Datenbank-Features bei geringen Kosten bietet. Um die Verbindung der Anwendung mit einer anderen Datenbank herzustellen, wiederholen Sie die Schritte der Sicherheitsgruppenkonfiguration und aktualisieren Sie die RDS-bezogenen Umgebungseigenschaften.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, konfigurieren Sie einen benutzerdefinierten Domainnamen für die Umgebung und aktivieren dann HTTPS für sichere Verbindungen.

Bereitstellung von Python-Anwendungen mit Elastic Beanstalk

Dieses Kapitel enthält Anweisungen zur Konfiguration und Bereitstellung Ihrer Python-Webanwendung auf AWS Elastic Beanstalk. Elastic Beanstalk macht es einfach, Ihre Python-Webanwendungen mithilfe von Amazon Web Services bereitzustellen, zu verwalten und zu skalieren.

Sie können Ihre Anwendung in nur wenigen Minuten mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) oder der Elastic Beanstalk Beanstalk-Konsole bereitstellen. Nachdem Sie Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt haben, können Sie weiterhin die EB-CLI verwenden, um Ihre Anwendung und Umgebung zu verwalten, oder Sie können die Elastic Beanstalk Beanstalk-Konsole oder die verwenden. AWS CLI APIs

Folgen Sie den step-by-step Anweisungen unter, um eine Python Hello World-Webanwendung mit der EB CLI zu erstellen und bereitzustellen. QuickStart für Python

Themen

- QuickStart: Stellen Sie eine Python-Anwendung auf Elastic Beanstalk bereit
- Einrichtung Ihrer Python-Entwicklungsumgebung für Elastic Beanstalk
- Verwenden der Elastic Beanstalk Python-Plattform
- Bereitstellen einer Flask-Anwendung in Elastic Beanstalk
- Bereitstellen einer Django-Anwendung in Elastic Beanstalk
- Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Python Elastic Beanstalk Beanstalk-Umgebung
- Python-Tools und -Ressourcen

QuickStart: Stellen Sie eine Python-Anwendung auf Elastic Beanstalk bereit

Dieses QuickStart Tutorial führt Sie durch den Prozess der Erstellung einer Python-Anwendung und deren Bereitstellung in einer AWS Elastic Beanstalk Umgebung.

⚠ Nicht für Produktionszwecke

Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine Python-Anwendung
- Schritt 2: Führen Sie Ihre Anwendung lokal aus
- Schritt 3: Stellen Sie Ihre Python-Anwendung mit der EB CLI bereit
- Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus
- Schritt 5: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center -Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

EB-CLI

Dieses Tutorial verwendet die Befehlszeilenschnittstelle Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Python- und Flask-Framework

Vergewissern Sie sich, dass Sie eine funktionierende Python-Version `pip` installiert haben, indem Sie die folgenden Befehle ausführen.

```
~$ python3 --version
Python 3.N.N
>~$ python3 -m pip --version
pip X.Y.Z from ... (python 3.N.N)
```

Wenn einer der vorherigen Befehle „Python wurde nicht gefunden“ zurückgibt, führen Sie die folgenden Befehle aus, die `python` anstelle von verwendenpython3. Die Einrichtung von Aliasnamen und symbolischen Links kann je nach Betriebssystem und individuellen Anpassungen variieren, sodass der `python3` Befehl auf Ihrem Computer möglicherweise nicht funktioniert.

```
~$ python --version
Python 3.N.N
>~$ python -m pip --version
pip X.Y.Z from ... (python 3.N.N)
```

Wenn Sie Python nicht auf Ihrem lokalen Computer installiert haben, können Sie es von der Python-Downloadseite auf der Python-Website herunterladen. Eine Liste der von Elastic Beanstalk unterstützten Python-Sprachversionen finden Sie unter Unterstützte AWS Elastic Beanstalk Python-Plattformen im Plattformhandbuch. Die Python-Download-Website bietet einen Link zum Python Developer's Guide, in dem Sie Installations- und Einrichtungsanweisungen finden.

> **Note**
>
> Das `pip` Python-Paket ist standardmäßig in Python 3.4 oder höher enthalten.

Wenn Ihre Ausgabe darauf hinweist, dass Sie eine unterstützte Version von Python haben, aber nicht`pip`, finden Sie weitere Informationen auf der Installationsseite auf der pip.pypa.io-Website. Es bietet Anleitungen zur Installation von Pip in einer Python-Umgebung, in der es nicht vorhanden ist.

Bestätigen Sie, ob Flask installiert ist, indem Sie den folgenden Befehl ausführen:

```
~$ pip list | grep Flask
```

Wenn Flask nicht installiert ist, können Sie es mit dem folgenden Befehl installieren:

```
~$ pip install Flask
```

Schritt 1: Erstellen Sie eine Python-Anwendung

Erstellen Sie ein Projektverzeichnis.

```
~$ mkdir eb-python
~$ cd eb-python
```

Erstellen Sie ein Beispiel für „Hello Elastic Beanstalk!“ Python-Anwendung, die Sie mit Elastic Beanstalk bereitstellen.

Erstellen Sie eine Textdatei mit dem Namen `application.py` in dem Verzeichnis, das Sie gerade erstellt haben, mit dem folgenden Inhalt.

Example~/eb-python/application.py

```
from flask import Flask
application = Flask(__name__)

@application.route('/')
def hello_elastic_beanstalk():
        return 'Hello Elastic Beanstalk!'
```

Erstellen Sie eine Textdatei `requirements.txt` mit dem Namen der folgenden Zeile. Diese Datei enthält die erforderlichen `pip` Pakete, damit die Anwendung ausgeführt werden kann.

Example~/eb-python/requirements.txt

```
Flask
```

Schritt 2: Führen Sie Ihre Anwendung lokal aus

Führen Sie den folgenden Befehl aus, um Ihre Anwendung lokal auszuführen.

```
~/eb-python$ export FLASK_APP=application.py && flask run --port 5000
```

Sie sollten eine Ausgabe sehen, die der folgenden ähnelt

```
Serving Flask app 'application.py'
Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a
 production WSGI server instead.
Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [01/Jan/1970 00:00:00] "GET / HTTP/1.1" 200 -
```

Navigieren Sie `http://localhost:5000` in Ihrem Webbrowser zu. Der Webbrowser sollte „Hello Elastic Beanstalk!“ anzeigen.

Schritt 3: Stellen Sie Ihre Python-Anwendung mit der EB CLI bereit

Führen Sie die folgenden Befehle aus, um eine Elastic Beanstalk Beanstalk-Umgebung für diese Anwendung zu erstellen.

Um eine Umgebung zu erstellen und Ihre Python-Anwendung bereitzustellen

1. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

   ```
   ~/eb-python$ eb init -p python-3.9 python-tutorial --region us-east-2
   ```

   Dieser Befehl erstellt eine Anwendung mit dem Namen `python-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der bereitgestellten Python-Plattformversion erstellt werden.
2. (Optional) Führen Sie eb init erneut aus, um ein Standardschlüsselpaar für die SSH-Verbindung zur EC2-Instance (auf der die Anwendung ausgeführt wird) zu konfigurieren.

   ```
   ~/eb-python$ eb init
   Do you want to set up SSH for your instances?
   ```

```
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein Schlüsselpaar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erstellt automatisch eine Zip-Datei für Ihre Anwendung und startet sie auf Port 5000.

```
~/eb-python$ eb create python-env
```

Es dauert etwa fünf Minuten, bis Elastic Beanstalk Ihre Umgebung erstellt hat.

Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, öffnen Sie Ihre Website mit. eb open

```
~/eb-python$ eb open
```

Herzlichen Glückwunsch! Sie haben eine Python-Anwendung mit Elastic Beanstalk bereitgestellt! Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet.

Schritt 5: Bereinigen

Sie können Ihre Umgebung beenden, wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind. Elastic Beanstalk beendet alle AWS Ressourcen, die mit Ihrer Umgebung verknüpft sind.

Führen Sie den folgenden Befehl aus, um Ihre Elastic Beanstalk Beanstalk-Umgebung mit der EB-CLI zu beenden.

```
~/eb-python$ eb terminate
```

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt. Es dient als einfache Beispielanwendung mit einer einzigen EC2-Instance, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 instance (EC2-Instance) – Eine virtuelle Amazon EC2-Maschine zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 zugelassen wird. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Web-App erreichen. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen.

Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen. Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

> **Probieren Sie weitere Tutorials aus**
>
> Wenn Sie andere Tutorials mit anderen Beispielanwendungen ausprobieren möchten, sehen Sie sich die folgenden Tutorials an:
>
> - Bereitstellen einer Flask-Anwendung in Elastic Beanstalk
> - Bereitstellen einer Django-Anwendung in Elastic Beanstalk

Nachdem Sie eine oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, Python-Anwendungen lokal zu entwickeln und auszuführen, finden Sie weitere Informationen unterEinrichtung Ihrer Python-Entwicklungsumgebung für Elastic Beanstalk.

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

Einrichtung Ihrer Python-Entwicklungsumgebung für Elastic Beanstalk

Dieses Thema enthält Anweisungen zum Einrichten einer Python-Entwicklungsumgebung, in der Sie Ihre Anwendung lokal testen können, bevor Sie sie bereitstellen AWS Elastic Beanstalk. Es verweist auch auf Websites, die Installationsanweisungen für nützliche Tools bereitstellen.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
```

```
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

Sections

- Voraussetzungen
- Verwenden einer virtuellen Umgebung
- Konfigurieren eines Python-Projekts für Elastic Beanstalk

Voraussetzungen

Die folgende Liste enthält die allgemeinen Voraussetzungen für die Arbeit mit Elastic Beanstalk und Ihren Python-Anwendungen:

- Python-Sprache — Installieren Sie die Version der Python-Sprache, die in der von Ihnen ausgewählten Version der Elastic Beanstalk-Python-Plattform enthalten ist. Eine Liste unserer unterstützten Python-Sprachversionen finden Sie unter Unterstützte AWS Elastic Beanstalk Python-Plattformen im Plattformhandbuch. Wenn Sie Python noch nicht auf Ihrem Entwicklungscomputer eingerichtet haben, besuchen Sie die Python-Downloadseite auf der Python-Website.
- **`pip`**Hilfsprogramm — Das `pip` Hilfsprogramm ist Pythons Paket-Installer. Es installiert und listet Abhängigkeiten für Ihr Projekt auf, sodass Elastic Beanstalk weiß, wie die Umgebung Ihrer Anwendung eingerichtet wird. Weitere Informationen *`pip`* dazu finden Sie auf der Pip-Seite auf der pip.pypa.io-Website.
- (Optional) Die Elastic Beanstalk Command Line Interface (EB CLI) — Die EB-CLI kann Ihre Anwendung mit den erforderlichen Bereitstellungsdateien verpacken. Es kann auch eine Elastic Beanstalk Beanstalk-Umgebung erstellen und Ihre Anwendung darauf bereitstellen. Sie können Bereitstellungen auch über die Elastic Beanstalk Beanstalk-Konsole vornehmen, sodass die EB-CLI nicht unbedingt erforderlich ist.
- Eine funktionierende **`SSH`** Installation — Sie können über das SSH-Protokoll eine Verbindung zu Ihren laufenden Instances herstellen, um ein Deployment zu untersuchen oder zu debuggen.
- **`virtualenv`**Paket — Dieses `virtualenv` Tool erstellt eine Entwicklungs- und Testumgebung für Ihre Anwendung. Elastic Beanstalk kann diese Umgebung replizieren, ohne zusätzliche Pakete zu installieren, die für Ihre Anwendung nicht erforderlich sind. Weitere Informationen finden Sie auf der

virtualenv-Website. Nach der Installation von Python können Sie das `virtualenv` Paket mit dem folgenden Befehl installieren:

```
$ pip install virtualenv
```

Verwenden einer virtuellen Umgebung

Sofern die Voraussetzungen erfüllt sind, können Sie eine virtuelle Umgebung mit `virtualenv` einrichten und die Anwendungsabhängigkeiten installieren. Mithilfe einer virtuellen Umgebung können Sie genau erkennen, welche Pakete von Ihrer Anwendung benötigt werden, sodass die erforderlichen Pakete auf den EC2 Instanzen installiert werden, auf denen Ihre Anwendung ausgeführt wird.

So richten Sie eine virtuelle Umgebung ein

1. Öffnen Sie ein Befehlszeilenfenster und geben Sie Folgendes ein:

   ```
   $ virtualenv /tmp/eb_python_app
   ```

   *eb_python_app*Ersetzen Sie es durch einen Namen, der für Ihre Anwendung sinnvoll ist (es ist eine gute Idee, den Namen Ihrer Anwendung zu verwenden). Der Befehl `virtualenv` erstellt für Sie eine virtuelle Umgebung im angegebenen Verzeichnis und druckt die Ergebnisse seiner Aktionen:

   ```
   Running virtualenv with interpreter /usr/bin/python
   New python executable in /tmp/eb_python_app/bin/python3.12
   Also creating executable in /tmp/eb_python_app/bin/python
   Installing setuptools, pip...done.
   ```

2. Sobald die virtuelle Umgebung bereit ist, starten Sie diese mithilfe des Skripts `activate`, das Sie im Verzeichnis `bin` der Umgebung finden. Um die im vorherigen Schritt erstellte Umgebung eb_python_app zu starten, geben Sie beispielsweise Folgendes ein:

   ```
   $ source /tmp/eb_python_app/bin/activate
   ```

   In der virtuellen Umgebung wird der Name (z. B. `(eb_python_app)`) am Anfang jeder Eingabeaufforderung angezeigt, damit Sie wissen, dass es sich um eine virtuelle Python-Umgebung handelt.

3. Führen Sie den Befehl `deactivate` aus, um die Verwendung Ihrer virtuellen Umgebung zu beenden und zum standardmäßigen Python-Interpreter des Systems mit allen installierten Bibliotheken zurückzukehren.

```
(eb_python_app) $ deactivate
```

> **Note**
>
> Wenn sie erstellt ist, können Sie die virtuelle Umgebung jederzeit starten, indem Sie das Skript `activate` erneut ausführen.

Konfigurieren eines Python-Projekts für Elastic Beanstalk

Sie können die Elastic Beanstalk CLI verwenden, um Ihre Python-Anwendungen auf die Bereitstellung mit Elastic Beanstalk vorzubereiten.

So konfigurieren Sie eine Python-Anwendung für die Bereitstellung mit Elastic Beanstalk

1. Kehren Sie in der virtuellen Umgebung nach oben zum Verzeichnisbaum des Projekts zurück (`python_eb_app`) und geben Sie Folgendes ein:

```
pip freeze >requirements.txt
```

   Mit diesem Befehl werden die Namen und Versionen der Pakete, die in der virtuellen Umgebung installiert sind, zu `requirements.txt` kopiert. Wenn beispielsweise das Paket PyYAML in der Version 3.11 in der virtuellen Umgebung installiert ist, enthält die Datei die folgende Zeile:

```
PyYAML==3.11
```

   Auf diese Weise verwendet Elastic Beanstalk dieselben Pakete und Versionen, mit denen Sie die Anwendung entwickelt und getestet haben, um die Python-Umgebung der Anwendung zu replizieren.

2. Konfigurieren Sie das Repository der EB CLI mit dem Befehl eb init. Folgen Sie den Anweisungen und wählen Sie die Region, die Plattform und andere Optionen aus.

Standardmäßig sucht Elastic Beanstalk nach einer Datei namens `application.py`, um die Anwendung zu starten. Falls diese im erstellten Python-Projekt nicht vorhanden ist, sind bestimmte Anpassungen an der Anwendungsumgebung erforderlich. Außerdem müssen Sie Umgebungsvariablen festlegen, damit die Anwendungsmodule geladen werden können. Weitere Informationen finden Sie unter Verwenden der Elastic Beanstalk Python-Plattform.

Verwenden der Elastic Beanstalk Python-Plattform

In diesem Thema wird beschrieben, wie Sie Ihre Python-Anwendungen auf Elastic Beanstalk konfigurieren, erstellen und ausführen.

AWS Elastic Beanstalk unterstützt eine Reihe von Plattformzweigen für verschiedene Versionen der Programmiersprache Python. Eine vollständige Liste finden Sie unter Python im Dokument AWS Elastic Beanstalk Plattformen.

Die Python-Webanwendungen können hinter einem Proxyserver mit WSGI ausgeführt werden. Elastic Beanstalk stellt Gunicorn als Standard-WSGI-Server bereit.

Sie können eine `Procfile` zu Ihrem Quell-Bundle hinzufügen, um den WSGI-Server für Ihre Anwendung anzugeben und zu konfigurieren. Details hierzu finden Sie unter the section called “Procfile”.

Mit den von Pipenv erstellten `Pipfile`- und `Pipfile.lock`-Dateien können Sie Python-Paketabhängigkeiten und andere Anforderungen angeben. Details zur Angabe von Abhängigkeiten finden Sie unter the section called “Angeben von Abhängigkeiten”.

Elastic Beanstalk bietet Konfigurationsoptionen, mit denen Sie die Software anpassen können, die auf den EC2 Instances in Ihrer Elastic Beanstalk Beanstalk-Umgebung ausgeführt wird. Sie können die für Ihre Anwendung erforderlichen Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren und die Ordner in der Anwendungsquelle, die statische Dateien enthalten, den vom Proxy-Server verwendeten Pfaden zuordnen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das Ändern der Konfiguration einer ausgeführten Umgebung verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie gespeicherte Konfigurationen verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie Konfigurationsdateien einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung

erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe Konfigurationsoptionen.

Bei Python-Paketen, die über `pip` bereitgestellt werden, können Sie eine Anforderungsdatei in den Stamm des Anwendungsquellcodes aufnehmen. Elastic Beanstalk installiert alle Abhängigkeitspakete, die während der Bereitstellung in einer Anforderungsdatei angegeben sind. Details hierzu finden Sie unter the section called “Angeben von Abhängigkeiten”.

Details zu den verschiedenen Erweiterungsmöglichkeiten einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter the section called “Erweitern von Linux-Plattformen”.

Konfigurieren Ihrer Python-Umgebung

Mit den Python-Plattformeinstellungen können Sie das Verhalten Ihrer EC2 Amazon-Instances optimieren. Sie können die EC2 Amazon-Instance-Konfiguration der Elastic Beanstalk Beanstalk-Umgebung mithilfe der Elastic Beanstalk Beanstalk-Konsole bearbeiten.

Verwenden Sie die Elastic Beanstalk Beanstalk-Konsole, um Python-Prozesseinstellungen zu konfigurieren AWS X-Ray, zu aktivieren, die Protokollrotation zu Amazon S3 zu aktivieren und Variablen zu konfigurieren, die Ihre Anwendung aus der Umgebung lesen kann.

So konfigurieren Sie Ihre Python-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

Python-Einstellungen

- Proxy server (Proxy-Server) – Der Proxyserver, der in Ihren Umgebungs-Instances verwendet werden soll. Standardmäßig wird nginx verwendet.
- WSGI-Pfad – Der Name oder Pfad zu Ihrer Hauptanwendungsdatei. Beispiel: `application.py` oder `django/wsgi.py`.
- NumProcesses— Die Anzahl der Prozesse, die auf jeder Anwendungsinstanz ausgeführt werden sollen.
- NumThreads— Die Anzahl der Threads, die in jedem Prozess ausgeführt werden sollen.

AWS X-Ray Einstellungen

- X-Ray-Daemon — Führen Sie den AWS X-Ray Daemon aus, um Trace-Daten aus dem zu verarbeiten. AWS X-Ray-SDK für Python

Protokolloptionen

Im Abschnitt mit den Protokolloptionen hat zwei Einstellungen:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Rotation der Protokolldateien nach Amazon S3 aktivieren — Gibt an, ob Protokolldateien für die EC2 Amazon-Instances Ihrer Anwendung in den Amazon S3-Bucket kopiert werden, der Ihrer Anwendung zugeordnet ist.

Statische Dateien

Um die Leistung zu verbessern, können Sie den Abschnitt Static files (Statische Dateien) benutzen, um den Proxy-Server so zu konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zum Verzeichnis-Mapping fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, anstatt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter the section called “Statische Dateien”.

Standardmäßig stellt der Proxy-Server in einer Python-Umgebung alle Dateien im Ordner `static` unter dem Pfad `/static` bereit. Wenn beispielsweise die Anwendungsquelle eine Datei mit dem Namen `logo.png` im Ordner `static` enthält, wird diese vom Proxy-Server unter *`subdomain`*`.elasticbeanstalk.com/static/logo.png` an die Benutzer weitergeleitet. Sie können zusätzliche Zuweisungen konfigurieren, wie in diesem Abschnitt erklärt.

Umgebungseigenschaften

Mithilfe von Umgebungseigenschaften stellen Sie Informationen für die Anwendung bereit und konfigurieren Umgebungsvariablen. Beispielsweise können Sie eine Umgebungseigenschaft mit dem Namen `CONNECTION_STRING` mit einer Verbindungszeichenfolge erstellen, über die Ihre Anwendung eine Verbindung zu einer Datenbank herstellen kann.

Innerhalb der in Elastic Beanstalk ausgeführten Python-Umgebung können Sie diese Werte im `os.environ`-Wörterbuch von Python finden. Weitere Informationen finden Sie unter http:// docs.python. org/library/os.html.

Der Code für den Zugriff auf Schlüssel und Parameter sollte wie folgt aussehen:

```
import os
endpoint = os.environ['API_ENDPOINT']
```

Umgebungseigenschaften können auch Informationen für ein Framework bereitstellen. Sie können beispielsweise eine Eigenschaft mit dem Namen `DJANGO_SETTINGS_MODULE` erstellen und so Django zur Verwendung eines bestimmten Einstellungsmoduls konfigurieren. Abhängig von der Umgebung könnte der Wert z. B. `development.settings` oder `production.settings` lauten.

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Namespaces der Python-Konfiguration

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben durchführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Die Python-Plattform definiert Optionen in den Namespaces `aws:elasticbeanstalk:environment:proxy`, `aws:elasticbeanstalk:environment:proxy:staticfiles` und `aws:elasticbeanstalk:container:python`.

In der folgenden Beispielkonfigurationsdatei werden Einstellungen für Konfigurationsoptionen angegeben, mit denen die `DJANGO_SETTINGS_MODULE`-Umgebungseigenschaft, zwei Optionen für statische Dateien, die das Verzeichnis `statichtml` dem Pfad `/html` und das Verzeichnis `staticimages` dem Pfad `/images` zuordnen, sowie weitere Einstellungen im Namespace `aws:elasticbeanstalk:container:python` erstellt und der Apache-Proxy-Server ausgewählt werden. Dieser Namespace enthält Optionen, mit denen Sie den Speicherort des WSGI-Skripts im Quellcode und die Anzahl der Threads und Prozesse, die in WSGI ausgeführt werden sollen, angeben können.

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    DJANGO_SETTINGS_MODULE: production.settings
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /html: statichtml
    /images: staticimages
  aws:elasticbeanstalk:container:python:
    WSGIPath: ebdjango.wsgi:application
    NumProcesses: 3
    NumThreads: 20
```

Hinweise

- Wenn Sie eine Amazon Linux AMI Python-Plattformversion (Vorgängerversion von Amazon Linux 2) verwenden, ersetzen Sie den Wert für `WSGIPath` durch `ebdjango/wsgi.py`. Der Wert in diesem Beispiel ist für einen Gunicorn WSGI-Server bestimmt, der von Amazon Linux AMI-Plattformversionen nicht unterstützt wird.
- Darüber hinaus verwenden diese älteren Plattformversionen einen anderen Namespace für die Konfiguration statischer Dateien:. —`aws:elasticbeanstalk:container:python:staticfiles`. Er hat die gleichen Optionsnamen und die gleiche Semantik wie der standardmäßige Namespace für statische Dateien.

Konfigurationsdateien unterstützen außerdem zahlreiche Schlüssel für die weitere Anpassung der Software auf den Umgebungs-Instances. In diesem Beispiel wird der Schlüssel packages (Pakete)

für die Memcached-Installation mit `yum` und Container-Befehlen verwendet, um Befehle für die Konfiguration des Servers während der Bereitstellung auszuführen.

```
packages:
  yum:
    libmemcached-devel: '0.31'

container_commands:
  collectstatic:
    command: "django-admin.py collectstatic --noinput"
  01syncdb:
    command: "django-admin.py syncdb --noinput"
    leader_only: true
  02migrate:
    command: "django-admin.py migrate"
    leader_only: true
  03wsgipass:
    command: 'echo "WSGIPassAuthorization On" >> ../wsgi.conf'
  99customize:
    command: "scripts/customize.sh"
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

`python3`Die ausführbare Datei

Die Version der `python3` ausführbaren Datei, die auf EC2 Instances in Elastic Beanstalk-Python-Umgebungen verfügbar ist, entspricht nicht immer derselben Python-Version, die von der Plattform verwendet wird. `/usr/bin/python3`Verweist beispielsweise auf der Plattform Python 3.12 AL2 023 auf Python 3.9. Dies liegt daran, dass Python 3.9 das System Python auf AL2 023 ist. Weitere Informationen finden Sie unter Python in AL2 023 im Amazon Linux 2023 User Guide. Sie können auf eine ausführbare Datei zugreifen, die der von der Plattform verwendeten Python-Version entspricht, an einem versionierten Speicherort (z. B.`/usr/bin/python3.12`) oder im `bin` Verzeichnis der virtuellen Anwendungsumgebung (z. B.`/var/app/venv/staging-LQM1lest/bin/python3`). Die Plattform verwendet die richtige ausführbare Python-Datei, die dem Plattformzweig entspricht.

Konfiguration des WSGI-Servers mit einem Profil auf Elastic Beanstalk

Sie können eine `Procfile` zu Ihrem Quell-Bundle hinzufügen, um den WSGI-Server für Ihre Anwendung anzugeben und zu konfigurieren. Sie können benutzerdefinierte Start- und Ausführungsbefehle in der angeben. `Procfile`

Wenn Sie eine `Procfile` verwenden, überschreibt sie die `aws:elasticbeanstalk:container:python`-Namespaceoptionen, die Sie mithilfe von Konfigurationsdateien festlegen.

Im folgenden Beispiel wird eine `Procfile` verwendet, um uWSGI als Server anzugeben und ihn zu konfigurieren.

Example Procfile

```
web: uwsgi --http :8000 --wsgi-file application.py --master --processes 4 --threads 2
```

Im folgenden Beispiel wird eine `Procfile` verwendet, um Gunicorn, den Standard-WSGI-Server, zu konfigurieren.

Example Procfile

```
web: gunicorn --bind :8000 --workers 3 --threads 2 project.wsgi:application
```

Hinweise

- Wenn Sie einen anderen WSGI-Server als Gunicorn konfigurieren, müssen Sie ihn auch als Abhängigkeit Ihrer Anwendung angeben, damit er auf Ihren Umgebungs-Instances installiert wird. Weitere Informationen zur Abhängigkeitsspezifikation finden Sie unter the section called “Angeben von Abhängigkeiten”.
- Der Standardport für den WSGI-Server ist 8000. Wenn Sie in Ihrem `Procfile`-Befehl eine andere Portnummer angeben, legen Sie auch die `PORT`-Umgebungseigenschaft auf diese Portnummer fest.

Angeben von Abhängigkeiten mithilfe einer Anforderungsdatei auf Elastic Beanstalk

In diesem Thema wird beschrieben, wie Sie Ihre Anwendung für die Installation anderer Python Python-Pakete konfigurieren. Eine typische Python-Anwendung weist Abhängigkeiten zu anderen Python-Paketen von Drittanbietern auf. Mit der Python-Plattform von Elastic Beanstalk haben Sie mehrere Möglichkeiten, Python-Pakete anzugeben, von denen Ihre Anwendung abhängt.

Verwendung von **`pip`** und **`requirements.txt`**

Das Standardwerkzeug für die Installation von Python-Paketen ist `pip`. Es hat eine Funktion, mit der Sie alle benötigten Pakete (und deren Versionen) in einer einzelnen Anforderungsdatei spezifizieren können. Weitere Informationen zur Anforderungsdatei finden Sie unter Format der Anforderungsdatei in der Website zur pip-Dokumentation.

Erstellen Sie eine Datei mit dem Namen `requirements.txt`, und speichern Sie diese im höchsten Verzeichnis Ihres Quell-Bundles. Das folgende Beispiel zeigt eine `requirements.txt`-Beispieldatei für Django.

```
Django==2.2
mysqlclient==2.0.3
```

In der Entwicklungsumgebung können Sie die Anforderungsdatei mit dem Befehl `pip freeze` generieren.

```
~/my-app$ pip freeze > requirements.txt
```

Damit sichergestellt ist, dass die Anforderungsdatei nur die tatsächlich von der Anwendung genutzten Pakete enthält, sollten Sie eine virtuelle Umgebung verwenden, in der nur diese Pakete installiert sind. Außerhalb einer virtuellen Umgebung sind in der Ausgabe von `pip freeze` alle `pip`-Pakete enthalten, die auf dem Entwicklungscomputer installiert sind, darunter auch solche, die mit dem Betriebssystem ausgeliefert wurden.

> **Note**
>
> Bei Amazon Linux AMI Python-Plattformversionen unterstützt Elastic Beanstalk Pipenv oder Pipfiles nicht nativ. Wenn Sie mit Pipenv Abhängigkeiten Ihrer Anwendung verwalten, führen Sie den folgenden Befehl aus, um eine `requirements.txt`-Datei zu generieren.

```
~/my-app$ pipenv lock -r > requirements.txt
```

Weitere Informationen finden Sie unter Generating a requirements.txt (Generieren einer Anforderungen.txt) in der Pipenv-Dokumentation.

Verwendung von Pipenv und `Pipfile`

Pipenv ist ein moderneres Python-Verpackungs-Tool. Es kombiniert die Paketinstallation mit der Erstellung und Verwaltung einer Abhängigkeitsdatei und einem virtualenv für Ihre Anwendung. Weitere Informationen finden Sie unter Pipenv: Python Dev Workflow for Humans (Pipenv: Python Dev Workflow für Menschen).

Pipenv verwaltet zwei Dateien:

- `Pipfile` – Diese Datei enthält verschiedene Arten von Abhängigkeiten und Anforderungen.
- `Pipfile.lock` – Diese Datei enthält einen Versions-Snapshot, der deterministische Builds ermöglicht.

Sie können die entsprechenden Dateien in Ihrer Entwicklungsumgebung erstellen und in das oberste Verzeichnis des Quellpakets aufnehmen, das Sie in Elastic Beanstalk bereitstellen. Weitere Informationen zu diesen beiden Dateien finden Sie unter Beispiel für Pipfile und Pipfile.lock.

Im folgenden Beispiel wird Pipenv verwendet, um Django und das Django REST-Framework zu installieren. Diese Befehle erstellen die Dateien `Pipfile` und `Pipfile.lock`.

```
~/my-app$ pipenv install django
~/my-app$ pipenv install djangorestframework
```

Precedence

Wenn Sie mehr als eine der hier beschriebenen Anforderungsdateien hinzufügen, wird Elastic Beanstalk nur eine davon verwenden. Die folgende Liste zeigt die Rangfolge in absteigender Reihenfolge.

1. `requirements.txt`

2. `Pipfile.lock`
3. `Pipfile`

> **Note**
>
> Ab der Amazon-Linux-2-Plattformversion vom 7. März 2023 gibt Elastic Beanstalk eine Konsolenmeldung aus, die angibt, welche der Abhängigkeitsdateien während einer Bereitstellung verwendet wurde, wenn Sie mehr als eine dieser Dateien bereitstellen.

In den folgenden Schritten wird die Logik beschrieben, der Elastic Beanstalk folgt, um die Abhängigkeiten zu installieren, wenn es eine Instance bereitstellt.

- Wenn es eine `requirements.txt`-Datei gibt, verwenden wir den Befehl `pip install -r requirements.txt`.
- Ab der Amazon Linux 2-Plattformversion vom 7. März 2023 verwenden wir den Befehl `pipenv sync`, wenn es keine `requirements.txt`-Datei gibt, aber eine `Pipfile.lock`. Vor dieser Veröffentlichung haben wir `pipenv install --ignore-pipfile` verwendet.
- Wenn es weder eine `requirements.txt`-Datei noch eine `Pipfile.lock` gibt, aber eine `Pipfile`, verwenden wir den Befehl `pipenv install --skip-lock`.
- Wenn keine der drei Anforderungsdateien gefunden wird, installieren wir keine Anwendungsabhängigkeiten.

Bereitstellen einer Flask-Anwendung in Elastic Beanstalk

Dieses Tutorial führt Sie durch den Prozess der Generierung einer Flask-Anwendung und deren Bereitstellung in einer AWS Elastic Beanstalk Umgebung. Flask ist ein Open Source-Framework für Python-Webanwendungen.

In diesem Tutorial führen Sie folgende Aufgaben durch:

- Einrichten einer virtuellen Python-Umgebung mit Flask
- Erstellen einer Flask-Anwendung
- Bereitstellen Ihrer Website mit der EB-CLI
- Bereinigen

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole kennen. Sofern noch nicht geschehen, starten Sie anhand der Anleitung in Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können die erste Elastic-Beanstalk-Umgebung.

Für die Verfahren in diesem Leitfaden benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine in Windows integrierte Version von Ubuntu und Bash zu erhalten.

In diesem Tutorial verwenden wir Python 3.11 und die entsprechende Elastic Beanstalk-Plattformversion. Installieren Sie Python mithilfe der Anleitung unter Einrichtung Ihrer Python-Entwicklungsumgebung für Elastic Beanstalk.

Das Flask-Framework wird im Rahmen des Tutorials installiert.

Dieses Tutorial verwendet die Elastic Beanstalk Befehlszeilenschnittstelle (Elastic Beanstalk Command Line, EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Einrichten einer virtuellen Python-Umgebung mit Flask

Erstellen Sie ein Projektverzeichnis sowie eine virtuelle Umgebung für Ihre Anwendung, und installieren Sie Flask.

So richten Sie die Projektumgebung ein

1. Erstellen Sie ein Projektverzeichnis.

   ```
   ~$ mkdir eb-flask
   ~$ cd eb-flask
   ```

2. Erstellen und aktivieren Sie eine virtuelle Umgebung mit dem Namen `virt`:

```
~/eb-flask$ virtualenv virt
~$ source virt/bin/activate
(virt) ~/eb-flask$
```

In der Eingabeaufforderung wird `(virt)` vorangestellt, sodass Sie wissen, dass Sie in einer virtuellen Umgebung arbeiten. Verwenden Sie die virtuelle Umgebung für den Rest dieses Tutorials.

3. Installieren Sie Flask mit `pip install`:

```
(virt)~/eb-flask$ pip install flask==3.0.3
```

4. Zeigen Sie die installierten Bibliotheken mit `pip freeze` an:

```
(virt)~/eb-flask$ pip freeze
blinker==1.8.2
click==8.1.7
Flask==3.0.3
importlib_metadata==8.5.0
itsdangerous==2.2.0
Jinja2==3.1.4
MarkupSafe==2.1.5
Werkzeug==3.0.4
zipp==3.20.2
```

Mit diesem Befehl werden alle Pakete aufgeführt, die in der virtuellen Umgebung installiert sind. Da Sie sich in einer virtuellen Umgebung befinden, werden global installierte Pakete wie die EB CLI nicht angezeigt.

5. Speichern Sie die Ausgabe aus `pip freeze` als Datei mit dem Namen `requirements.txt`.

```
(virt)~/eb-flask$ pip freeze > requirements.txt
```

Diese Datei weist Elastic Beanstalk an, die Bibliotheken während der Bereitstellung zu installieren. Weitere Informationen finden Sie unter Angeben von Abhängigkeiten mithilfe einer Anforderungsdatei auf Elastic Beanstalk.

Erstellen einer Flask-Anwendung

Erstellen Sie anschließend eine Anwendung, die Sie mit Elastic Beanstalk bereitstellen. Wir werden einen „Hello World“ RESTful -Webservice erstellen.

Erstellen Sie eine neue Textdatei in diesem Verzeichnis mit dem Namen `application.py` mit folgendem Inhalt:

Example~/eb-flask/application.py

```
from flask import Flask

# print a nice greeting.
def say_hello(username = "World"):
    return '<p>Hello %s!</p>\n' % username

# some bits of text for the page.
header_text = '''
    <html>\n<head> <title>EB Flask Test</title> </head>\n<body>'''
instructions = '''
    <p><em>Hint</em>: This is a RESTful web service! Append a username
    to the URL (for example: <code>/Thelonious</code>) to say hello to
    someone specific.</p>\n'''
home_link = '<p><a href="/">Back</a></p>\n'
footer_text = '</body>\n</html>'

# EB looks for an 'application' callable by default.
application = Flask(__name__)

# add a rule for the index page.
application.add_url_rule('/', 'index', (lambda: header_text +
    say_hello() + instructions + footer_text))

# add a rule when the page is accessed with a name appended to the site
# URL.
application.add_url_rule('/<username>', 'hello', (lambda username:
    header_text + say_hello(username) + home_link + footer_text))

# run the app.
if __name__ == "__main__":
    # Setting debug to True enables debug output. This line should be
    # removed before deploying a production app.
    application.debug = True
```

```
    application.run()
```

In diesem Beispiel wird eine benutzerdefinierte Begrüßung gedruckt, die je nach Pfad, der für den Zugriff auf den Service verwendet wurde, variiert.

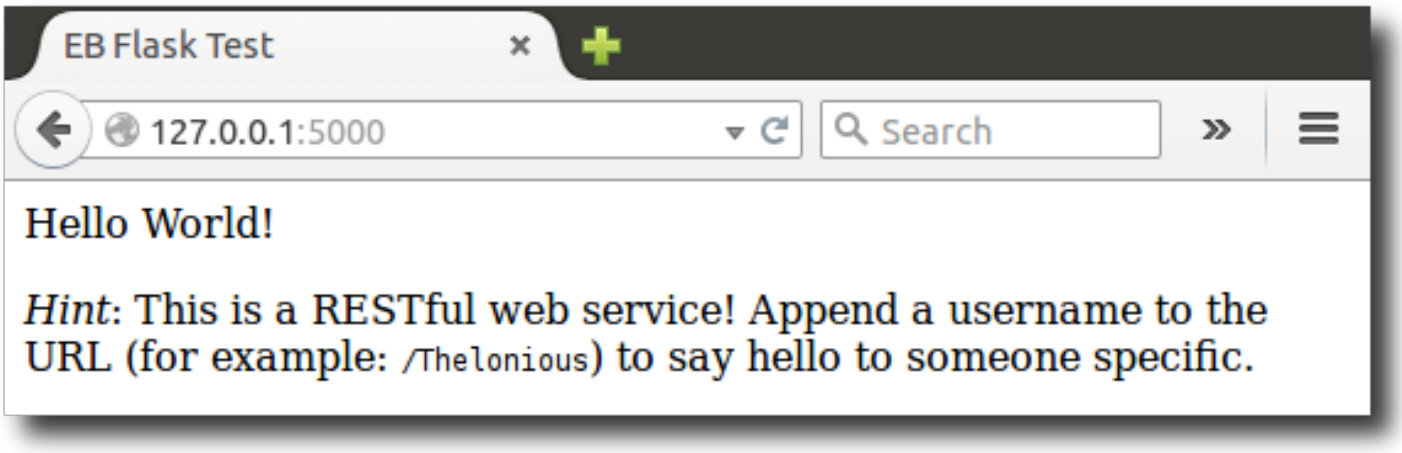
> **Note**
>
> Indem Sie `application.debug = True` vor der Ausführung der Anwendung hinzufügen, ist das Debuggen der Ausgabe aktiviert, falls ein Problem auftritt. Es ist eine bewährte Methode für die Entwicklung, Sie sollten jedoch Debug-Anweisungen im Produktionscode entfernen, da eine Debug-Ausgabe interne Aspekte Ihrer Anwendung offenlegen kann.

Verwenden Sie `application.py` als Dateinamen und geben Sie ein aufrufbares `application`-Objekt an (in diesem Fall das Flask-Objekt), damit Elastic Beanstalk den Code Ihrer Anwendung leichter finden kann.

Führen Sie `application.py` mit Python aus:

```
(virt) ~/eb-flask$ python application.py
 * Serving Flask app "application" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 313-155-123
```

Öffnen Sie `http://127.0.0.1:5000/` in Ihrem Webbrowser. Sie sollten sehen, dass die Anwendung ausgeführt wird. Dabei wird folgende Indexseite angezeigt:



Rufen Sie die Ausgabe der Anforderung im Serverprotokoll auf. Mit Ctrl+C stoppen Sie den Server und kehren in die virtuelle Umgebung zurück.

Wenn Sie stattdessen eine Debug-Ausgabe erhalten, beheben Sie die Fehler und stellen Sie sicher, dass die Anwendung lokal ausgeführt wird, bevor Sie sie für Elastic Beanstalk konfigurieren.

Bereitstellen Ihrer Website mit der EB-CLI

Die Anwendung kann nun in Elastic Beanstalk bereitgestellt werden. Das Projektverzeichnis sollte wie folgt aussehen:

```
~/eb-flask/
|-- virt
|-- application.py
`-- requirements.txt
```

Der `virt`-Ordner ist jedoch für die Ausführung der Anwendung in Elastic Beanstalk nicht erforderlich. Bei der Bereitstellung erstellt Elastic Beanstalk eine neue virtuelle Umgebung auf den Server-Instances und installiert die in `requirements.txt` aufgeführten Bibliotheken. Um die Größe des Quell-Bundles zu minimieren, das Sie während der Bereitstellung hochladen, fügen Sie eine .ebignore-Datei hinzu, die der EB CLI mitteilt, dass der Ordner `virt` ausgelassen werden soll.

Example~/eb-flask/.ebignore

```
virt
```

Im nächsten Schritt erstellen Sie die Anwendungsumgebung und stellen die konfigurierte Anwendung in Elastic Beanstalk bereit.

So erstellen Sie eine Umgebung und stellen die Flask-Anwendung bereit

1. Initialisieren Sie das Repository der EB CLI mit dem Befehl eb init:

   ```
   ~/eb-flask$ eb init -p python-3.11 flask-tutorial --region us-east-2
   Application flask-tutorial has been created.
   ```

   Dieser Befehl erstellt eine neue Anwendung mit dem Namen `flask-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der neuesten Python 3.11-Plattformversion erstellt werden.

2. (optional) Führen Sie den Befehl eb init erneut aus, um ein Standardschlüsselpaar zu konfigurieren, sodass Sie sich mit SSH mit der EC2 Instanz verbinden können, auf der Ihre Anwendung ausgeführt wird:

   ```
   ~/eb-flask$ eb init
   Do you want to set up SSH for your instances?
   (y/n): y
   Select a keypair.
   1) my-keypair
   2) [ Create new KeyPair ]
   ```

   Wählen Sie ein vorhandenes Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Aufforderungen, um ein neues Paar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen:

   ```
   ~/eb-flask$ eb create flask-env
   ```

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder

einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
>
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise

schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

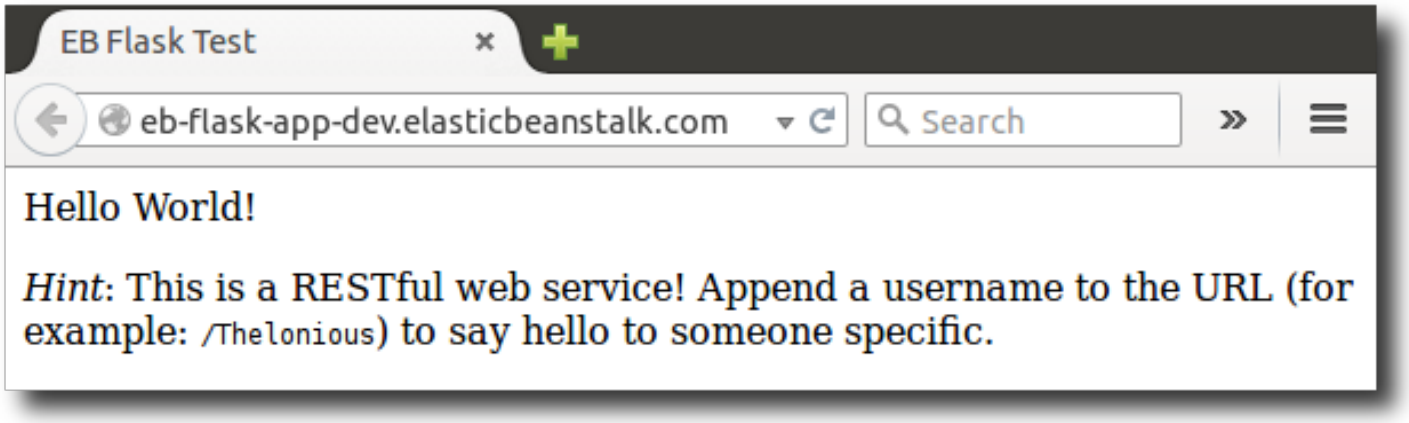
All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

> **Note**
>
> Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon S3.

Wenn die Umgebungserstellung abgeschlossen ist, öffnen Sie die Website mit eb open:

```
~/eb-flask$ eb open
```

Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet. Sie sollten dieselbe Flask-Website sehen, die Sie erstellt und lokal getestet haben.



Falls die Anwendung nicht ausgeführt wird oder Sie eine Fehlermeldung erhalten, finden Sie unter Troubleshooting deployments (Behebung von Bereitstellungsfehlern) weitere Informationen zur Ermittlung der Fehlerursache.

Wenn Sie die Anwendung ausführen, haben Sie Ihre erste Flask-Anwendung mit Elastic Beanstalk bereitgestellt. Herzlichen Glückwunsch!

Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung über die Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Oder mit der EB CLI:

```
~/eb-flask$ eb terminate flask-env
```

Nächste Schritte

Weitere Informationen über Flask finden Sie unter flask.pocoo.org.

Unter Bereitstellen einer Django-Anwendung in Elastic Beanstalk steht Ihnen ein weiteres Python-Web-Framework zur Verfügung.

Bereitstellen einer Django-Anwendung in Elastic Beanstalk

In diesem Tutorial erfahren Sie, wie die Bereitstellung einer standardmäßigen, automatisch generierten Django-Website in einer AWS Elastic Beanstalk -Umgebung mit Python erfolgt. In diesem Tutorial erfahren Sie, wie Sie eine Python-Webanwendung in der Cloud mithilfe einer Elastic Beanstalk-Umgebung hosten.

In diesem Tutorial führen Sie folgende Aufgaben durch:

- Einrichten einer virtuellen Python-Umgebung und Installieren von Django
- Erstellen eines Django-Projekts
- Konfigurieren der Django-Anwendung für Elastic Beanstalk
- Bereitstellen Ihrer Website mit der EB-CLI
- Aktualisieren der Anwendung
- Bereinigen

Voraussetzungen

Damit Sie die Schritte in diesem Tutorial durchführen können, müssen alle allgemeinen Voraussetzungen für Python einschließlich der folgenden Pakete installiert sein:

- Python 3.7 oder höher
- `pip`
- `virtualenv`
- `awsebcli`

Das Django-Framework wird im Rahmen des Tutorials installiert.

> **Note**
>
> Zum Erstellen von Umgebungen mit der EB CLI ist eine Servicerolle erforderlich. Eine Servicerolle generieren Sie, indem Sie eine Umgebung in der Elastic-Beanstalk-Konsole erstellen. Falls keine Servicerolle vorhanden ist, versucht die EB CLI eine Rolle zu erstellen, wenn Sie ausführen `eb create`.

Einrichten einer virtuellen Python-Umgebung und Installieren von Django

Erstellen Sie eine virtuelle Umgebung mit `virtualenv` und verwenden Sie sie anschließend zur Installation von Django und der entsprechenden Abhängigkeiten. Durch die Verwendung einer virtuellen Umgebung können Sie genau wissen, welche Pakete Ihre Anwendung benötigt, sodass die erforderlichen Pakete auf den EC2 Amazon-Instances installiert werden, auf denen Ihre Anwendung ausgeführt wird.

Die folgenden Schritte zeigen die Befehle, die Sie für Unix-basierte Systeme und Windows eingeben müssen. Sie werden auf separaten Registerkarten angezeigt.

So richten Sie eine virtuelle Umgebung ein

1. Erstellen Sie eine virtuelle Umgebung mit dem Namen `eb-virt`.

   Unix-based systems

   ```
   ~$ virtualenv ~/eb-virt
   ```

   Windows

   ```
   C:\> virtualenv %HOMEPATH%\eb-virt
   ```

2. Aktivieren Sie die virtuelle Umgebung.

   Unix-based systems

   ```
   ~$ source ~/eb-virt/bin/activate
   (eb-virt) ~$
   ```

   Windows

   ```
   C:\>%HOMEPATH%\eb-virt\Scripts\activate
   (eb-virt) C:\>
   ```

   In der Eingabeaufforderung wird `(eb-virt)` vorangestellt, sodass Sie wissen, dass Sie in einer virtuellen Umgebung arbeiten.

   > **Note**
   >
   > In den restlichen Anleitungsschritten wird die Linux-Eingabeaufforderung im Stammverzeichnis ~$ angezeigt. Unter Windows ist das`C:\Users\`*USERNAME*>, wo *USERNAME* ist Ihr Windows-Anmeldename.

3. Installieren Sie mit `pip` Django.

```
(eb-virt)~$ pip install django==2.2
```

> **Note**
>
> Die Django-Version, die Sie installieren, muss mit der Python-Version in der Phyton-Konfiguration von Elastic Beanstalk, die Sie für die Bereitstellung Ihrer Anwendung auswählen, kompatibel sein. Weitere Informationen zur Bereitstellung finden Sie unter ??? in diesem Thema.
> Weitere Informationen zu aktuellen Python-Plattformversionen finden Sie unter Python im Dokument AWS Elastic Beanstalk -Plattformen.
> Informationen zur Kompatibilität von Django-Versionen mit Python finden Sie unter Welche Phyton-Version kann ich mit Django verwenden?

4. Zur Überprüfung, ob Django installiert ist, geben Sie Folgendes ein.

```
(eb-virt)~$ pip freeze
Django==2.2
...
```

Mit diesem Befehl werden alle Pakete aufgeführt, die in der virtuellen Umgebung installiert sind. Später verwenden Sie die Ausgabe dieses Befehls, um Ihr Projekt zur Verwendung mit Elastic Beanstalk zu konfigurieren.

Erstellen eines Django-Projekts

Nun können Sie in der virtuellen Umgebung ein Django-Projekt erstellen und auf dem Computer ausführen.

> **Note**
>
> Dieses Tutorial verwendet SQLite, eine Datenbank-Engine, die in Python enthalten ist. Die Datenbank wird mit Ihren Projektdateien bereitgestellt. Für Produktionsumgebungen wird empfohlen, Amazon Relational Database Service (Amazon RDS) zu verwenden und von der Umgebung zu trennen. Weitere Informationen finden Sie unter Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Python Elastic Beanstalk Beanstalk-Umgebung.

So erstellen Sie eine Django-Anwendung

1. Aktivieren Sie die virtuelle Umgebung.

   Unix-based systems

   ```
   ~$ source ~/eb-virt/bin/activate
   (eb-virt) ~$
   ```

   Windows

   ```
   C:\>%HOMEPATH%\eb-virt\Scripts\activate
   (eb-virt) C:\>
   ```

   In der Eingabeaufforderung wird das Präfix `(eb-virt)` vorangestellt, sodass Sie wissen, dass Sie in einer virtuellen Umgebung arbeiten.

   > **Note**
   >
   > In den restlichen Anleitungsschritten werden die Linux-Eingabeaufforderung ~$ in Ihrem Stammverzeichnis und das Linux-Stammverzeichnis ~/ angezeigt. Unter Windows sind dies`C:\Users\`*USERNAME*>, wo *USERNAME* ist Ihr Windows-Anmeldename.

2. Verwenden Sie den Befehl `django-admin startproject` zum Erstellen eines Django-Projekts mit dem Namen `ebdjango`.

   ```
   (eb-virt)~$ django-admin startproject ebdjango
   ```

   Mit diesem Befehl wird eine Django-Standard-Website mit dem Namen ebdjango und folgender Verzeichnisstruktur erstellt.

   ```
   ~/ebdjango
     |-- ebdjango
     |   |-- __init__.py
     |   |-- settings.py
     |   |-- urls.py
     |   `-- wsgi.py
     `-- manage.py
   ```

3. Führen Sie die Django-Website lokal mit `manage.py runserver` aus.

```
(eb-virt) ~$ cd ebdjango
```

```
(eb-virt) ~/ebdjango$ python manage.py runserver
```

4. Öffnen Sie zum Anzeigen der Website `http://127.0.0.1:8000/` in einem Webbrowser.
5. Rufen Sie die Ausgabe der Anforderung im Serverprotokoll auf. Um den Webserver zu stoppen und zu Ihrer virtuellen Umgebung zurückzukehren, drücken Sie Strg+C.

```
Django version 2.2, using settings 'ebdjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[07/Sep/2018 20:14:09] "GET / HTTP/1.1" 200 16348
Ctrl+C
```

Konfigurieren der Django-Anwendung für Elastic Beanstalk

Sie verfügen nun auf Ihrem lokalen Computer über eine Website mit Django-Unterstützung und können diese für die Bereitstellung in Elastic Beanstalk konfigurieren.

Standardmäßig sucht Elastic Beanstalk nach einer Datei mit dem Namen `application.py`, um Ihre Anwendung zu starten. Da diese im Django-Projekt, das Sie erstellt haben, nicht vorhanden ist, nehmen Sie einige Anpassungen an der Anwendungsumgebung vor. Außerdem legen Sie Umgebungsvariablen fest, damit die Anwendungsmodule geladen werden können.

So konfigurieren Sie Ihre Website für Elastic Beanstalk

1. Aktivieren Sie die virtuelle Umgebung.

   Unix-based systems

   ```
   ~/ebdjango$ source ~/eb-virt/bin/activate
   ```

   Windows

   ```
   C:\Users\USERNAME\ebdjango>%HOMEPATH%\eb-virt\Scripts\activate
   ```

2. Führen Sie `pip freeze` aus und speichern Sie die Ausgabe in der Datei `requirements.txt`.

```
(eb-virt) ~/ebdjango$ pip freeze > requirements.txt
```

Elastic Beanstalk verwendet`requirements.txt`, um zu bestimmen, welches Paket auf den EC2 Instances installiert werden soll, auf denen Ihre Anwendung ausgeführt wird.

3. Erstellen Sie ein Verzeichnis mit dem Namen `.ebextensions`.

```
(eb-virt) ~/ebdjango$ mkdir .ebextensions
```

4. Fügen Sie im Verzeichnis `.ebextensions` eine Konfigurationsdatei mit dem Namen `django.config` und folgendem Text hinzu.

   Example~/ebdjango/.ebextensions/django.config

```
option_settings:
  aws:elasticbeanstalk:container:python:
    WSGIPath: ebdjango.wsgi:application
```

   Mit der Einstellung `WSGIPath` wird der Speicherort des WSGI-Skripts angegeben, das von Elastic Beanstalk für den Start der Anwendung verwendet wird.

> **Note**
>
> Wenn Sie eine Amazon Linux AMI Python-Plattformversion (Vorgängerversion von Amazon Linux 2) verwenden, ersetzen Sie den Wert für `WSGIPath` durch `ebdjango/wsgi.py`. Der Wert in diesem Beispiel ist für einen Gunicorn WSGI-Server bestimmt, der von Amazon Linux AMI-Plattformversionen nicht unterstützt wird.

5. Deaktivieren Sie die virtuelle Umgebung mit dem Befehl `deactivate`.

```
(eb-virt) ~/ebdjango$ deactivate
```

Sie können die virtuelle Umgebung jederzeit wieder aktivieren, um Pakete zur Anwendung hinzuzufügen oder diese lokal auszuführen.

Bereitstellen Ihrer Website mit der EB-CLI

Die Anwendung kann nun in Elastic Beanstalk bereitgestellt werden. Das Projektverzeichnis sollte wie folgt aussehen.

```
~/ebdjango/
|-- .ebextensions
|   `-- django.config
|-- ebdjango
|   |-- __init__.py
|   |-- settings.py
|   |-- urls.py
|   `-- wsgi.py
|-- db.sqlite3
|-- manage.py
`-- requirements.txt
```

Im nächsten Schritt erstellen Sie die Anwendungsumgebung und stellen die konfigurierte Anwendung in Elastic Beanstalk bereit.

Unmittelbar nach der Bereitstellung bearbeiten Sie die Django-Konfiguration. Fügen Sie der Django-`ALLOWED_HOSTS` den Domänennamen hinzu, den Elastic Beanstalk der Anwendung zugewiesen hat. Anschließend stellen Sie Ihre Anwendung erneut bereit. Dies ist eine Django-Sicherheitsanforderung, um HTTP-`Host`-Header-Angriffe zu verhindern. Weitere Informationen finden Sie unter Host header validation (Host-Header-Validierung).

So erstellen Sie eine Umgebung und die Django-Anwendung bereit

> **Note**
>
> Dieses Tutorial verwendet die EB-CLI als Bereitstellungsmechanismus. Sie können jedoch auch die Elastic Beanstalk-Konsole zum Bereitstellen einer ZIP-Datei mit den Inhalten des Projekts verwenden.

1. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

   ```
   ~/ebdjango$ eb init -p python-3.7 django-tutorial
   Application django-tutorial has been created.
   ```

Dieser Befehl erstellt eine Anwendung mit dem Namen `django-tutorial`. Außerdem wird Ihr lokales Repository konfiguriert, damit Sie Umgebungen mit der neuesten Python 3.7-Plattformversion erstellen können.

2. (Optional) Führen Sie den Vorgang eb init erneut aus, um ein Standardschlüsselpaar zu konfigurieren, sodass Sie SSH verwenden können, um eine Verbindung zu der EC2 Instance herzustellen, auf der Ihre Anwendung ausgeführt wird.

```
~/ebdjango$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein Schlüsselpaar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen.

```
~/ebdjango$ eb create django-env
```

> **Note**
>
> Wenn eine Fehlermeldung angezeigt wird, weil eine Servicerolle erforderlich ist, führen Sie `eb create` interaktiv aus (ohne Angabe eines Umgebungsnamens). Die EB CLI erstellt die Rolle dann für Sie.

Mit diesem Befehl wird eine lastenverteilte Elastic Beanstalk-Umgebung namens `django-env` erstellt. Das Erstellen einer Umgebung dauert ca. 5 Minuten. Bei der zur Anwendungsausführung erforderlichen Ressourcenerstellung durch Elastic Beanstalk werden die ausgegebenen Informationsmeldungen von der EB CLI an den Terminal weitergeleitet.

4. Nachdem die Umgebung erstellt wurde, können Sie den Domänennamen der neuen Umgebung ermitteln, indem Sie eb status ausführen.

```
~/ebdjango$ eb status
Environment details for: django-env
  Application name: django-tutorial
  ...
  CNAME: eb-django-app-dev.elasticbeanstalk.com
  ...
```

Der Domänenname der Umgebung ist der Wert der Eigenschaft `CNAME`.

5. Öffnen Sie die `settings.py`-Datei im Verzeichnis `ebdjango`. Suchen Sie die `ALLOWED_HOSTS`-Einstellung und fügen Sie dann den Domänennamen Ihrer Anwendung, den Sie im vorherigen Schritt gefunden haben, zum Wert der Einstellung hinzu. Wenn Sie diese Einstellung nicht in der Datei finden, fügen Sie sie in einer neuen Zeile hinzu.

```
...
ALLOWED_HOSTS = ['eb-django-app-dev.elasticbeanstalk.com']
```

6. Speichern Sie die Datei und stellen Sie die Anwendung dann bereit, indem Sie eb deploy ausführen . Bei Ausführung von eb deploy werden die Inhalte des Projektverzeichnisses von der EB CLI gebündelt und in der Umgebung bereitgestellt.

```
~/ebdjango$ eb deploy
```

> **Note**
>
> Wenn Sie bei Ihrem Projekt Git verwenden, lesen Sie Verwenden der EB CLI mit Git.

7. Wenn die Umgebungsaktualisierung abgeschlossen ist, öffnen Sie die Website mit eb open.

```
~/ebdjango$ eb open
```

Es wird ein Browserfenster mit dem für die Anwendung erstellten Domänennamen geöffnet. Sie sollten dieselbe Django-Website sehen, die Sie erstellt und lokal getestet haben.

Falls die Anwendung nicht ausgeführt wird oder Sie eine Fehlermeldung erhalten, finden Sie unter Troubleshooting deployments (Behebung von Bereitstellungsfehlern) weitere Informationen zur Ermittlung der Fehlerursache.

Wenn Sie die Anwendung ausführen, haben Sie Ihre erste Django-Anwendung mit Elastic Beanstalk bereitgestellt. Herzlichen Glückwunsch!

Aktualisieren der Anwendung

Ihre Anwendung wird nun in Elastic Beanstalk ausgeführt. Sie können die Anwendung oder deren Konfiguration aktualisieren und erneut bereitstellen. Elastic Beanstalk führt dabei die Aktualisierung der Instances aus und startet Ihre neue Anwendungsversion.

In diesem Beispiel aktivieren Sie die Django-Admin-Konsole und konfigurieren weitere Einstellungen.

Ändern der Einstellungen der Website

Standardmäßig erfolgt die Zeitangabe der Django-Website in koordinierter Weltzeit (UTC). Sie können diese Einstellung ändern, indem Sie eine Zeitzone in `settings.py` angeben.

So ändern Sie die Zeitzone der Website

1. Ändern Sie die `TIME_ZONE`-Einstellung in `settings.py`.

   Example~/ .py ebdjango/ebdjango/settings

   ```
   ...
   # Internationalization
   LANGUAGE_CODE = 'en-us'
   TIME_ZONE = 'US/Pacific'
   USE_I18N = True
   USE_L10N = True
   USE_TZ = True
   ```

   Eine Liste der Zeitzonen finden Sie auf dieser Seite.
2. Stellen Sie die Anwendung in Ihrer Elastic Beanstalk-Umgebung bereit.

   ```
   ~/ebdjango/$ eb deploy
   ```

Erstellen eines Websiteadministrators

Sie können einen Websiteadministrator für Ihre Django-Anwendung erstellen, um direkt von der Website auf die Administrationskonsole zuzugreifen. Die Anmeldeinformationen für den Administrator

werden sicher im lokalen Datenbank-Image gespeichert, das in das von Django generierte Standardprojekt eingebunden ist.

So erstellen Sie einen Websiteadministrator

1. Initialisieren Sie die lokale Datenbank der Django-Anwendung.

```
(eb-virt) ~/ebdjango$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```

2. Führen Sie zum Erstellen eines Administrators `manage.py createsuperuser` aus.

```
(eb-virt) ~/ebdjango$ python manage.py createsuperuser
Username: admin
Email address: me@mydomain.com
Password: ********
Password (again): ********
Superuser created successfully.
```

3. Um Django einen Speicherort für statische Dateien zuzuweisen, definieren Sie `STATIC_ROOT` in `settings.py`.

   Example~/ .py ebdjango/ebdjango/settings

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-files/
```

```
STATIC_URL = '/static/'
STATIC_ROOT = 'static'
```

4. Führen Sie aus`manage.py collectstatic`, um das `static` Verzeichnis mit statischen Elementen (JavaScript, CSS und Bildern) für die Admin-Site zu füllen.

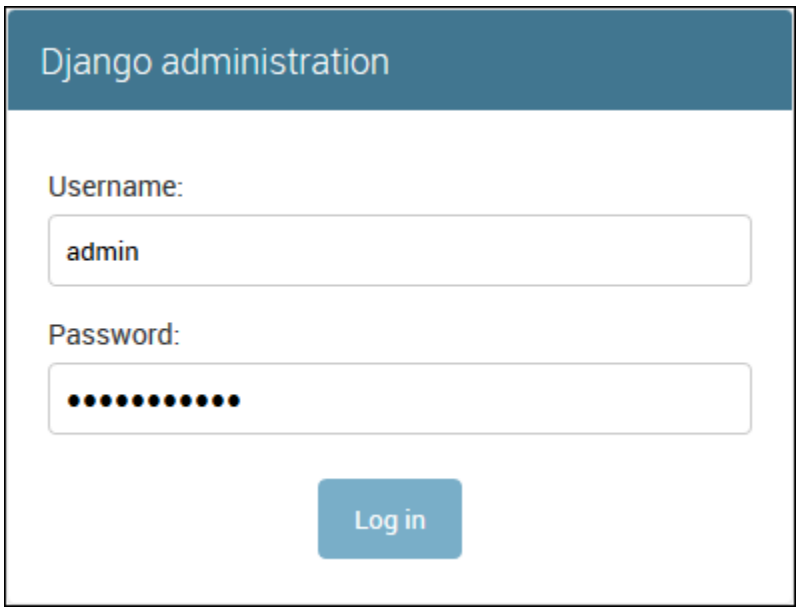
```
(eb-virt) ~/ebdjango$ python manage.py collectstatic
119 static files copied to ~/ebdjango/static
```

5. Stellen Sie die Anwendung bereit.

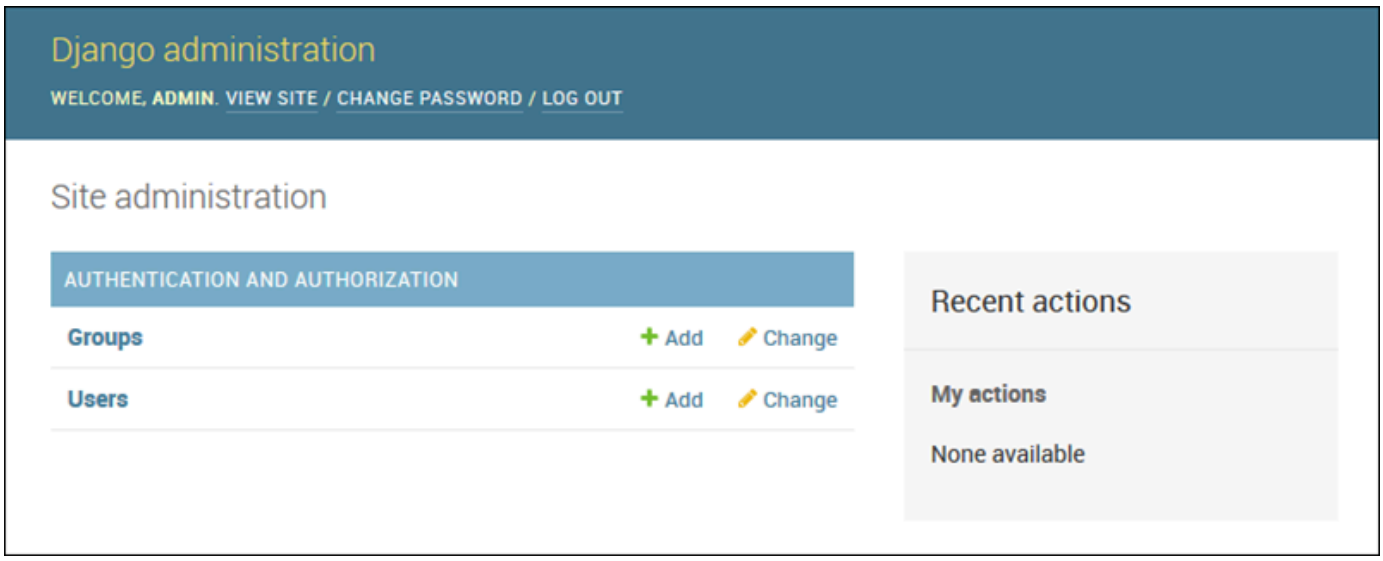
```
~/ebdjango$ eb deploy
```

6. Rufen Sie die Administrationskonsole auf, indem Sie die Website im Browser öffnen und `/admin/` an die Website-URL anfügen, z. B. im Folgenden.

```
http://djang-env.p33kq46sfh.us-west-2.elasticbeanstalk.com/admin/
```



7. Melden Sie sich mit dem in Schritt 2 konfigurierten Benutzernamen und dem zugehörigen Passwort an.



Sie können ein ähnliches Verfahren verwenden: local updating/testing gefolgt voneb deploy. Elastic Beanstalk führt die Aktualisierung der Live-Server aus, sodass Sie sich nicht um die Serveradministration kümmern müssen, sondern sich ganz auf die Anwendungsbereitstellung konzentrieren können.

Hinzufügen einer Konfigurationsdatei für die Datenbankmigration

Sie können Ihrem `.ebextensions`-Skript Befehle hinzufügen, die beim Aktualisieren der Website ausgeführt werden. Auf diese Weise können Sie automatisch Datenbankmigrationen generieren.

So fügen Sie bei der Anwendungsbereitstellung einen Migrationsschritt hinzu

1. Erstellen Sie eine Konfigurationsdatei namens `db-migrate.config` mit dem folgenden Inhalt.

   Example~/ebdjango/.ebextensions/db-migrate.config

   ```
   container_commands:
     01_migrate:
       command: "source /var/app/venv/*/bin/activate && python3 manage.py migrate"
       leader_only: true
   option_settings:
     aws:elasticbeanstalk:application:environment:
       DJANGO_SETTINGS_MODULE: ebdjango.settings
   ```

   Diese Konfigurationsdatei führt den `manage.py migrate`-Befehl während des Bereitstellungsprozesses aus, bevor Ihre Anwendung gestartet wird. Da die

Befehlsausführung vor dem Start der Anwendung erfolgt, muss die Umgebungsvariable `DJANGO_SETTINGS_MODULE` explizit konfiguriert werden (in der Regel wird dies von `wsgi.py` während des Starts übernommen). Geben Sie im Befehl `leader_only: true` an, damit dieser auch bei einer Bereitstellung auf mehreren Instances nur einmal ausgeführt wird.

2. Stellen Sie die Anwendung bereit.

```
~/ebdjango$ eb deploy
```

Bereinigen

Um Instance-Stunden und andere AWS Ressourcen zwischen den Entwicklungssitzungen zu sparen, beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung mit. eb terminate

```
~/ebdjango$ eb terminate django-env
```

Dieser Befehl beendet die Umgebung und alle AWS Ressourcen, die in ihr ausgeführt werden. Die Anwendung wird jedoch nicht gelöscht. Sie können durch die erneute Ausführung von eb create jederzeit zusätzliche Umgebungen mit der gleichen Konfiguration erstellen.

Sofern Sie die Beispielanwendung nicht mehr benötigen, können Sie auch den Projektordner sowie die virtuelle Umgebung entfernen.

```
~$ rm -rf ~/eb-virt
~$ rm -rf ~/ebdjango
```

Nächste Schritte

Weitere Informationen zu Django und ein ausführliches Tutorial finden Sie in der offiziellen Dokumentation.

Unter Bereitstellen einer Flask-Anwendung in Elastic Beanstalk steht Ihnen ein weiteres Python-Web-Framework zur Verfügung.

Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Python Elastic Beanstalk Beanstalk-Umgebung

Dieses Thema enthält Anweisungen zum Erstellen eines Amazon RDS mit der Elastic-Beanstalk-Konsole. Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann an Ihre Umgebung gekoppelt sein und von Elastic Beanstalk verwaltet werden oder sie kann als entkoppelte Datenbank erstellt und extern von einem anderen Service verwaltet werden. In dieser Anleitung wird die Datenbank an Ihre Umgebung gekoppelt und von Elastic Beanstalk verwaltet. Weitere Informationen zur Integration eines Amazon RDS mit Elastic Beanstalk finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Sections

- Hinzufügen einer DB-Instance zu Ihrer Umgebung
- Herunterladen eines Treibers
- Herstellen einer Verbindung zu einer Datenbank

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftenwert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.

Weitere Informationen zur Konfiguration einer Datenbank-Instance in Verbindung mit einer Elastic-Beanstalk-Umgebung finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Herunterladen eines Treibers

Fügen Sie den Datenbanktreiber zur requirements-Datei (Anforderungsdatei) des Projekts hinzu.

Example Beispieldatei "requirements.txt" – Django mit MySQL

```
Django==2.2
mysqlclient==2.0.3
```

Gängige Treiberpakete für Python

- MySQL – `mysqlclient`
- PostgreSQL – `psycopg2`
- Oracle – `cx_Oracle`
- SQL Server – `adodbapi`

Weitere Informationen finden Sie unter Python DatabaseInterfaces und Django 2.2 — unterstützte Datenbanken.

Herstellen einer Verbindung zu einer Datenbank

Elastic Beanstalk stellt Verbindungsinformationen für zugehörige DB-Instances in Umgebungseigenschaften bereit. Mit `os.environ['VARIABLE']` können Sie die Eigenschaften lesen und eine Datenbankverbindung konfigurieren.

Example Django-Einstellungsdatei – DATABASES Dictionary

```
import os

if 'RDS_HOSTNAME' in os.environ:
    DATABASES = {
        'default': {
            'ENGINE': 'django.db.backends.mysql',
            'NAME': os.environ['RDS_DB_NAME'],
            'USER': os.environ['RDS_USERNAME'],
            'PASSWORD': os.environ['RDS_PASSWORD'],
            'HOST': os.environ['RDS_HOSTNAME'],
            'PORT': os.environ['RDS_PORT'],
        }
    }
```

Python-Tools und -Ressourcen

Es gibt zahlreiche Möglichkeiten, wie Sie zusätzliche Unterstützung bei der Entwicklung von Python-Anwendungen erhalten können:

Ressource	Description
AWS SDK für Python (Boto3) auf GitHub	Installieren Sie Boto3, das von bezogen wurde. GitHub
AWS SDK für Python (Boto3) Homepage	Die AWS SDK für Python (Boto3) Homepage.
Python-Entwicklerzentrum	Umfangreiches Kit mit Beispiel-Code, Dokumentation, Tools und zusätzlichen Ressourcen.

Bereitstellung von Ruby-Anwendungen mit Elastic Beanstalk

Dieses Kapitel enthält Anweisungen zur Konfiguration und Bereitstellung Ihrer Ruby-Webanwendung für AWS Elastic Beanstalk. Elastic Beanstalk macht es einfach, Ihre Ruby-Webanwendungen mithilfe von Amazon Web Services bereitzustellen, zu verwalten und zu skalieren.

Sie können Ihre Anwendung in nur wenigen Minuten mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) oder der Elastic Beanstalk Beanstalk-Konsole bereitstellen. Nachdem Sie Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt haben, können Sie weiterhin die EB-CLI verwenden, um Ihre Anwendung und Umgebung zu verwalten, oder Sie können die Elastic Beanstalk Beanstalk-Konsole verwenden, AWS CLI oder die. APIs

Dieses Kapitel enthält auch step-by-step Anweisungen für die Bereitstellung einer Beispielanwendung auf Elastic Beanstalk mithilfe der EB-CLI und die anschließende Aktualisierung der Anwendung für die Verwendung der Webanwendungs-Frameworks Rails und Sinatra.

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse zu Elastic-Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im Tutorial „Erste Schritte“ mit den Grundlagen vertraut.

Themen

- Einrichtung Ihrer Ruby-Entwicklungsumgebung für Elastic Beanstalk
- Verwenden der Elastic Beanstalk-Ruby-Plattform
- Bereitstellen einer Rails-Anwendung in Elastic Beanstalk
- Bereitstellen einer Sinatra-Anwendung in Elastic Beanstalk
- Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Ruby Elastic Beanstalk Beanstalk-Umgebung

Einrichtung Ihrer Ruby-Entwicklungsumgebung für Elastic Beanstalk

Dieses Kapitel enthält Anweisungen zum Einrichten einer Ruby-Entwicklungsumgebung, in der Sie Ihre Anwendung lokal testen können, bevor Sie sie bereitstellen. AWS Elastic Beanstalk Es verweist auch auf Websites, die Installationsanweisungen für nützliche Tools bereitstellen.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem

vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

Sections

- Installieren von Ruby
- Das AWS SDK for Ruby installieren
- Installieren einer IDE oder eines Texteditors

Installieren von Ruby

Installieren Sie GCC, wenn Sie keinen C-Compiler haben. Unter Ubuntu verwenden Sie `apt`.

```
~$ sudo apt install gcc
```

Verwenden Sie auf Amazon Linux `yum`.

```
~$ sudo yum install gcc
```

Installieren Sie RVM, um Ruby-Sprachinstallationen auf Ihrem Rechner zu verwalten. Verwenden Sie die Befehle von rvm.io, um die Projektschlüssel zu erhalten und das Installationsskript auszuführen.

```
~$ gpg2 --recv-keys key1 key2
~$ curl -sSL https://get.rvm.io | bash -s stable
```

Dieses Skript installiert RVM in einem Ordner namens `.rvm` in Ihrem Benutzerverzeichnis und ändert Ihr Shell-Profil, um ein Setup-Skript zu laden, wenn Sie ein neues Terminal öffnen. Laden Sie die Skripts manuell, um beginnen zu können.

```
~$ source ~/.rvm/scripts/rvm
```

Verwenden Sie `rvm get head`, um die neueste Version zu erhalten.

```
~$ rvm get head
```

Zeigen Sie die verfügbaren Versionen von Ruby an.

```
~$ rvm list known
```

Lesen Sie Ruby im Dokument für AWS Elastic Beanstalk -Plattformen, um die neueste Version von Ruby zu finden, die für eine Elastic-Beanstalk-Plattform verfügbar ist. Installieren Sie diese Version.

```
~$ rvm install 3.2
```

Testen Sie Ihre Ruby-Installation.

```
~$ ruby --version
```

Das AWS SDK for Ruby installieren

Wenn Sie AWS Ressourcen von Ihrer Anwendung aus verwalten müssen, installieren Sie das AWS SDK für Ruby. Sie können mit dem SDK for Ruby beispielsweise Amazon DynamoDB (DynamoDB) zum Speichern von Benutzer- und Sitzungsinformationen verwenden, ohne eine relationale Datenbank zu erstellen.

Installieren Sie das SDK for Ruby und seine Abhängigkeiten mit dem Befehl `gem`.

```
$ gem install aws-sdk
```

Weitere Informationen und Installationsanleitungen finden Sie auf der AWS SDK für Ruby - Homepage.

Installieren einer IDE oder eines Texteditors

Integrierte Entwicklungsumgebungen (IDEs) bieten eine Vielzahl von Funktionen, die die Anwendungsentwicklung erleichtern. Wenn Sie noch keine IDE für die Ruby-Entwicklung verwendet haben, probieren Sie Aptana aus RubyMine und finden Sie heraus, welche für Sie am besten geeignet ist.

- Installieren von Aptana
- RubyMine

> **Note**
>
> Mit einer IDE werden dem Projektordner möglicherweise Dateien hinzugefügt, die Sie unter Umständen nicht an die Quellverwaltung übergeben möchten. Damit diese Dateien nicht an die Quellüberwachung übergeben werden, setzen Sie `.gitignore` oder die Entsprechung Ihres Quellüberwachungstools ein.

Wenn Sie gerade mit dem Programmieren beginnen und nicht alle Funktionen einer IDE benötigen, können Sie installing Sublime Text (Installieren von Sublime Text) auswählen.

Verwenden der Elastic Beanstalk-Ruby-Plattform

In diesem Thema wird beschrieben, wie Sie Ihre Ruby-Anwendungen auf Elastic Beanstalk konfigurieren, erstellen und ausführen.

AWS Elastic Beanstalk unterstützt eine Reihe von Plattformzweigen für verschiedene Versionen der Programmiersprache Ruby. Eine vollständige Liste finden Sie unter Ruby im Dokument AWS Elastic Beanstalk Plattformen.

Die Ruby-Webanwendung kann hinter einem NGINX-Proxyserver unter einem Puma-Anwendungsserver ausgeführt werden. Wenn Sie dies verwenden RubyGems, können Sie a `Gemfile`in Ihr Quellpaket aufnehmen, um Pakete während der Bereitstellung zu installieren.

Anwendungsserverkonfiguration

Elastic Beanstalk installiert den Puma-Anwendungsserver basierend auf dem Zweig der Ruby-Plattform, den Sie bei der Erstellung Ihrer Umgebung auswählen. Weitere Informationen zu den Komponenten der Ruby-Plattformversionen finden Sie unter Supported Platforms (Unterstützte Plattformen) in der AWS Elastic Beanstalk -Plattformen-Anleitung.

Sie können Ihre Anwendung mit Ihrem eigenen Puma-Server konfigurieren. Dies bietet die Möglichkeit, eine andere Version von Puma als die, die mit dem Ruby-Plattformzweig vorinstalliert ist, zu verwenden. Sie können Ihre Anwendung auch so konfigurieren, dass sie einen anderen Anwendungsserver wie Passenger verwendet. Dazu müssen Sie einen `Gemfile` in Ihrer Bereitstellung einbinden und einstellen. Sie müssen auch einen `Procfile` konfigurieren, um den Anwendungsserver zu starten. Weitere Informationen finden Sie unter Configuring the application process with a Procfile (Konfigurieren der Anwendung mit einer Procfile-Datei).

Andere Konfigurationsoptionen

Elastic Beanstalk bietet Konfigurationsoptionen, mit denen sich die auf den Amazon Elastic Compute Cloud (Amazon EC2)-Instances in der Elastic Beanstalk-Umgebung ausgeführte Software anpassen lässt. Sie können die für Ihre Anwendung erforderlichen Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren und die Ordner in der Anwendungsquelle, die statische Dateien enthalten, den vom Proxy-Server verwendeten Pfaden zuordnen. Die Plattform verfügt auch über einige allgemeine Umgebungsvariablen für Rails und Rack, die eine einfache Erkennung und Nutzung ermöglichen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das Ändern der Konfiguration einer ausgeführten Umgebung verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie gespeicherte Konfigurationen verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie Konfigurationsdateien einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe Konfigurationsoptionen.

Details zu den verschiedenen Möglichkeiten für das Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter the section called “Erweitern von Linux-Plattformen”.

Konfigurieren Ihrer Ruby-Umgebung

Mit der Elastic Beanstalk-Konsole können Sie die Protokollrotation für Amazon S3 aktivieren und Variablen konfigurieren, die die Anwendung in der Umgebung lesen kann.

So greifen Sie auf die Softwarekonfigurationseinstellungen für die Umgebung zu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

Protokolloptionen

Im Abschnitt Log options (Protokolloptionen) sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon-EC2-Instances der Anwendung in den der Anwendung zugeordneten Amazon-S3-Bucket kopiert werden.

Statische Dateien

Um die Leistung zu verbessern, können Sie den Abschnitt Static files (Statische Dateien) benutzen, um den Proxy-Server so zu konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zum Verzeichnis-Mapping fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, anstatt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter the section called “Statische Dateien”.

Standardmäßig ist der Proxy-Server in einer Ruby-Umgebung so konfiguriert, dass er statische Dateien wie folgt bereitstellt:

- Dateien im`public`-Ordner werden vom `/public`-Pfad und vom Domänenstamm bedient (`/`-Pfad).
- Dateien im`public/assets`Unterordner werden vom `/assets`-Pfad bedient.

Die Funktionsweise wird anhand der Funktionsweise der Standardkonfiguration deutlich:

- Wenn beispielsweise die Anwendungsquelle eine Datei mit dem Namen `logo.png` im Ordner `public` enthält, leitet der Proxy-Server sie von *`subdomain`*`.elasticbeanstalk.com/public/logo.png` und *`subdomain`*`.elasticbeanstalk.com/logo.png` an die Benutzer weiter.

- Wenn beispielsweise die Anwendungsquelle eine Datei mit dem Namen `logo.png` `assets` im `public`-Ordner enthält, leitet der Proxy-Server sie von *`subdomain`*`.elasticbeanstalk.com/assets/logo.png` weiter.

Sie können zusätzliche Zuordnungen für statische Dateien konfigurieren. Weitere Informationen finden Sie unter Namespaces der Ruby-Konfigurationen an späterer Stelle in diesem Thema.

> **Note**
>
> Für Plattformversionen vor Ruby 2.7 AL2 Version 3.3.7 unterstützt die standardmäßige Nginx-Proxy-Serverkonfiguration von Elastic Beanstalk die Bereitstellung statischer Dateien aus dem Domänenstamm () nicht. *`subdomain`*`.elasticbeanstalk.com/` Diese Plattformversion wurde am 21. Oktober 2021 veröffentlicht. Weitere Informationen finden Sie unter Neue Plattformversionen - Ruby in den AWS Elastic Beanstalk Versionshinweisen.

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon-EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel/Wert-Paare an die Anwendung weitergeleitet.

Die Ruby-Plattform definiert folgende Eigenschaften für die Umgebungskonfiguration:

- BUNDLE_WITHOUT – Eine durch Doppelpunkt getrennte Liste mit Gruppen, die bei der Installation von Abhängigkeiten aus einer Gemfile-Datei ignoriert werden sollen.
- BUNDLER_DEPLOYMENT_MODE – Auf `true` gesetzt (Standardeinstellung), um Abhängigkeiten im Bereitstellungsmodus mit Bundler zu installieren. Auf `false` gesetzt, um die `bundle install` im Entwicklungsmodus auszuführen.

  > **Note**
  >
  > Diese Umgebungseigenschaft ist in den Versionen der Amazon Linux AMI Ruby-Plattform (Vorgänger von Amazon Linux 2) nicht definiert.

- RAILS_SKIP_ASSET_COMPILATION – Setzen Sie den Wert auf `true`, damit `rake assets:precompile` während der Bereitstellung nicht ausgeführt wird.

- RAILS_SKIP_MIGRATIONS – Setzen Sie den Wert auf `true`, damit `rake db:migrate` während der Bereitstellung nicht ausgeführt wird.
- RACK_ENV – Geben Sie die Umgebungsphase für Rack an. Beispiel: `development`, `production` oder `test`.

Innerhalb der Ruby-Umgebung, die in Elastic Beanstalk ausgeführt wird, können Sie über das Objekt `ENV` auf Umgebungsvariablen zugreifen. Beispielsweise können Sie mit folgendem Code eine Eigenschaft mit dem Namen `API_ENDPOINT` als Variable lesen:

```
endpoint = ENV['API_ENDPOINT']
```

Weitere Informationen finden Sie unter Umgebungsvariablen und andere Softwareeinstellungen.

Namespaces der Ruby-Konfigurationen

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben durchführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

Sie können mithilfe des Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` den Umgebungsproxy so konfigurieren, dass statische Dateien bereitgestellt werden. Sie definieren Zuordnungen von virtuellen Pfaden zu Anwendungsverzeichnissen.

Die Ruby-Plattform definiert keine plattformspezifischen Namespaces. Stattdessen werden Umgebungseigenschaften für häufig verwendete Rails- und Rack-Optionen definiert.

Die folgende Konfigurationsdatei gibt eine Option für statische Dateien an, mit der das Verzeichnis `staticimages` dem Pfad `/images` zuordnet wird und die einzelnen plattformdefinierten Umgebungseigenschaften und eine zusätzliche Umgebungseigenschaft mit dem Namen `LOGGING` festlegt werden.

Example.ebextensions/ruby-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /images: staticimages
  aws:elasticbeanstalk:application:environment:
```

```
    BUNDLE_WITHOUT: test
    BUNDLER_DEPLOYMENT_MODE: true
    RACK_ENV: development
    RAILS_SKIP_ASSET_COMPILATION: true
    RAILS_SKIP_MIGRATIONS: true
    LOGGING: debug
```

> **Note**
>
> Die Umgebungseigenschaft `BUNDLER_DEPLOYMENT_MODE` und der Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` sind auf Amazon Linux AMI Ruby-Plattformvarianten (Vorgänger von Amazon Linux 2) nicht definiert.

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter Konfigurationsoptionen.

Pakete mit einem Gemfile auf Elastic Beanstalk installieren

Um Pakete RubyGems zu installieren, die Ihre Anwendung benötigt, fügen Sie eine `Gemfile` Datei in das Stammverzeichnis Ihrer Projektquelle ein.

Example Gemfile

```
source "https://rubygems.org"
gem 'sinatra'
gem 'json'
gem 'rack-parser'
```

Wenn eine `Gemfile`-Datei vorhanden ist, führt Elastic Beanstalk `bundle install` aus, um Abhängigkeiten zu installieren. Weitere Informationen finden Sie auf den Seiten Gemfiles und Bundle auf der Bundler.io-Website.

> **Note**
>
> Sie können neben der Standardversion, die auf der Ruby-Plattform vorinstalliert ist, eine andere Version von Puma verwenden. Hierfür machen Sie eine Eingabe in den `Gemfile`, die

die Version spezifiziert. Sie können auch einen anderen Anwendungsserver, z. B. Passenger, angeben, indem Sie einen `Gemfile` anpassen.
Für beide Fälle müssen Sie einen `Procfile` konfigurieren, um den Anwendungsserver zu starten.
Weitere Informationen finden Sie unter Configuring the application process with a Procfile (Konfigurieren der Anwendung mit einer Procfile-Datei).

Konfiguration des Bewerbungsprozesses mit einem Profil auf Elastic Beanstalk.

Um den Befehl anzugeben, mit dem Ihre Ruby-Anwendung gestartet wird, fügen Sie eine Datei namens `Procfile` in das Stammverzeichnis des Quell-Bundles ein.

> **Note**
>
> Elastic Beanstalk unterstützt diese Funktion nicht auf Amazon Linux AMI-Ruby-Plattformvarianten (vor Amazon Linux 2). Plattformzweige mit Namen, die mit Puma oder mit Passenger enthalten, haben unabhängig von ihrer Ruby-Version Vorrang vor Amazon Linux 2 und unterstützen die Funktion `Procfile` nicht.

Einzelheiten zum Schreiben und Verwenden von a finden Sie `Procfile` unter. Buildfile und Procfile

Wenn Sie keine angeben`Procfile`, generiert Elastic Beanstalk eine Standardeinstellung. `Procfile` Wenn Sie `Gemfile` Puma einbeziehen, geht Elastic Beanstalk davon aus, dass Sie Ihre bereitgestellte Version von Puma verwenden möchten, und generiert den folgenden Standard. `Procfile`

```
web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Wenn Sie Puma `Gemfile` nicht enthalten, geht Elastic Beanstalk davon aus, dass Sie den vorinstallierten Puma-Anwendungsserver verwenden, und generiert den folgenden Standard. `Procfile` Auf Amazon Linux 2 Ruby-Plattformbranches generiert Elastic Beanstalk immer den folgenden Standard, `Procfile` wenn Sie keinen angeben. `Procfile`

```
web: puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

> **Note**
>
> Am 10. Oktober 2024 wurden die letzten Zweige der Ruby Amazon Linux 2-Plattform eingestellt. Alle derzeit unterstützten Ruby-Plattformzweige basieren auf Amazon Linux 2023. Informationen zur Migration finden Sie unterMigration von Amazon Linux 2 zu Amazon Linux 2023.

Wenn Sie den Passenger-Anwendungsserver verwenden möchten, verwenden Sie die folgenden Beispieldateien, um Ihre Ruby-Umgebung für die Installation und Verwendung von Passenger zu konfigurieren.

1. Verwenden Sie diese Beispieldatei, um Passenger zu installieren.

   Example Gemfile

   ```
   source 'https://rubygems.org'
   gem 'passenger'
   ```

2. Verwenden Sie diese Beispieldatei, um Elastic Beanstalk anzuweisen, Passenger zu starten.

   Example Procfile

   ```
   web: bundle exec passenger start /var/app/current --socket /var/run/puma/my_app.sock
   ```

> **Note**
>
> Sie müssen in der Konfiguration des nginx-Proxy-Servers nichts ändern, um Passenger verwenden zu können. Um andere Anwendungsserver verwenden zu können, müssen Sie möglicherweise die nginx-Konfiguration anpassen, um Anforderungen ordnungsgemäß an Ihre Anwendung weiterzuleiten.

Bereitstellen einer Rails-Anwendung in Elastic Beanstalk

Rails ist ein Open-Source-Framework model-view-controller (MVC) für Ruby. Dieses Tutorial führt Sie durch den Prozess der Generierung einer Rails-Anwendung und deren Bereitstellung in einer AWS Elastic Beanstalk Umgebung.

Sections

- Voraussetzungen
- Grundkenntnisse zu Elastic Beanstalk
- Starten einer Elastic Beanstalk-Umgebung
- Installieren von Rails und Generieren einer Website
- Konfigurieren von Rails-Einstellungen
- Bereitstellen der Anwendung
- Bereinigen
- Nächste Schritte

Voraussetzungen

Grundkenntnisse zu Elastic Beanstalk

In diesem Tutorial wird davon ausgegangen, dass Sie die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole kennen. Sofern noch nicht geschehen, starten Sie anhand der Anleitung in Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können die erste Elastic-Beanstalk-Umgebung.

Befehlszeile

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine in Windows integrierte Version von Ubuntu und Bash zu erhalten.

Rails-Abhängigkeiten

Das Rails-Framework 6.1.4.1 hat die folgenden Abhängigkeiten. Stellen Sie sicher, dass Sie alle installiert haben.

- Ruby 2.5.0 oder höher – Installationsanweisungen finden Sie unter Einrichtung Ihrer Ruby-Entwicklungsumgebung für Elastic Beanstalk.

  In diesem Tutorial verwenden wir Ruby 3.0.2 und die entsprechende Elastic Beanstalk-Plattformversion.
- Node.js – Anweisungen zur Installation finden Sie unter Installieren von Node.js über den Package Manager.
- Yarn – Anweisungen zur Installation finden Sie unter Installation auf der Yarn-Website.

Starten einer Elastic Beanstalk-Umgebung

Erstellen Sie mit der Elastic Beanstalk-Konsole, eine Elastic Beanstalk-Umgebung. Wählen Sie die Plattform Ruby und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole über diesen vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication?Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus und wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder

NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise

schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

> **Note**
>
> Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon S3.

Installieren von Rails und Generieren einer Website

Installieren Sie Rails und seine Abhängigkeiten mit dem Befehl `gem`.

```
~$ gem install rails
Fetching: concurrent-ruby-1.1.9.gem
Successfully installed concurrent-ruby-1.1.9
Fetching: rack-2.2.3.gem
Successfully installed rack-2.2.3
...
```

Testen Sie Ihre Rails-Installation.

```
~$ rails --version
Rails 6.1.4.1
```

Verwenden Sie `rails new` mit dem Namen der Anwendung, um ein neues Rails-Projekt zu erstellen.

```
~$ rails new ~/eb-rails
```

Rails erstellt ein Verzeichnis mit dem angegebenen Namen, generiert alle Dateien, die für eine lokale Ausführung des Beispielprojekts benötigt werden, und führt anschließend Bundler aus, um alle Abhängigkeiten (Gems) zu installieren, die in der Gemfile-Datei des Projekts definiert sind.

> **Note**
>
> Dieser Prozess installiert die neueste Puma-Version für das Projekt. Diese Version kann sich von der Version unterscheiden, die Elastic Beanstalk in der Ruby-Plattformversion Ihrer Umgebung bereitstellt. Weitere Informationen zu Puma-Versionen von Elastic Beanstalk finden Sie unter Ruby-Plattformverlauf im Leitfaden für AWS Elastic Beanstalk -Plattformen. Weitere Informationen über die neueste Pumaversion finden Sie unter der Puma.io-Website. Wenn es eine Nichtübereinstimmung zwischen den beiden Puma-Versionen gibt, verwenden Sie eine der folgenden Optionen:
>
> - Verwenden Sie die Puma-Version, die mit dem vorherigen `rails new`-Befehl installiert wurde. In diesem Fall müssen Sie ein `Procfile` für die Plattform hinzufügen, um Ihre eigene bereitgestellte Puma-Serverversion zu verwenden. Weitere Informationen finden Sie unter Konfiguration des Bewerbungsprozesses mit einem Profil auf Elastic Beanstalk..
> - Aktualisieren Sie die Puma-Version so, dass sie mit der Version der Ruby-Platform Ihrer Umgebung übereinstimmt. Ändern Sie hierzu die Pumaversion im Gemfile, die sich im Stammverzeichnis des Projekts befindet. Führen Sie dann `bundle update` aus. Weitere Informationen finden Sie auf der Seite Bundle-Update auf der Bundler.io-Website.

Testen Sie die Rails-Installation, indem Sie das Standardprojekt lokal ausführen.

```
~$ cd eb-rails
~/eb-rails$ rails server
=> Booting Puma
=> Rails 6.1.4.1 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.5.2 (ruby 3.0.2-p107) ("Zawgyi")
*  Min threads: 5
*  Max threads: 5
*  Environment: development
*          PID: 77857
* Listening on http://127.0.0.1:3000
* Listening on http://[::1]:3000
Use Ctrl-C to stop
...
```

Öffnen Sie `http://localhost:3000` in einem Webbrowser, um das Standardprojekt in Aktion zu sehen.



Diese Seite ist nur im Entwicklungsmodus sichtbar. Fügen Sie Inhalte auf der Titelseite der Anwendung hinzu, um die Produktionsbereitstellung in Elastic Beanstalk zu unterstützen. Erstellen Sie mit `rails generate` einen Controller, eine Route und eine Ansicht für die Willkommensseite.

```
~/eb-rails$ rails generate controller WelcomePage welcome
      create  app/controllers/welcome_page_controller.rb
       route  get 'welcome_page/welcome'
      invoke  erb
      create    app/views/welcome_page
      create    app/views/welcome_page/welcome.html.erb
      invoke  test_unit
      create    test/controllers/welcome_page_controller_test.rb
      invoke  helper
      create    app/helpers/welcome_page_helper.rb
      invoke    test_unit
      invoke  assets
```

```
      invoke    coffee
      create      app/assets/javascripts/welcome_page.coffee
      invoke    scss
      create      app/assets/stylesheets/welcome_page.scss.
```

Dies gibt Ihnen alles, was Sie brauchen, um auf die Seite unter `/welcome_page/welcome` zuzugreifen. Bevor Sie die Änderungen veröffentlichen, sollten Sie die Inhalte der Ansicht ändern und eine Route hinzufügen, damit diese Seite als Top-Level-Objekt der Website angezeigt wird.

Ändern Sie die Inhalte in `app/views/welcome_page/welcome.html.erb` mit einem Texteditor. In diesem Beispiel werden die Inhalte der bestehenden Datei einfach mit `cat` überschrieben.

Example app/views/welcome_page/welcome.html.erb

```
<h1>Welcome!</h1>
<p>This is the front page of my first Rails application on Elastic Beanstalk.</p>
```

Abschließend fügen Sie die folgende Route zu `config/routes.rb` hinzu:

Example config/routes.rb

```
Rails.application.routes.draw do
  get 'welcome_page/welcome'
  root 'welcome_page#welcome'
```

Damit wird Rails angewiesen, an den Website-Stamm gerichtete Anforderungen an die Willkommen-Methode des Controllers der Willkommenseite zu leiten, von der die Inhalte in der Willkommen-Ansicht dargestellt werden (`welcome.html.erb`).

Damit Elastic Beanstalk die Anwendung erfolgreich auf der Ruby-Plattform bereitstellen kann, aktualisieren Sie `Gemfile.lock`. Einige Abhängigkeiten von `Gemfile.lock` können plattformspezifisch sein. Deshalb müssen Sie **`platform ruby`** zu `Gemfile.lock` hinzufügen, damit alle erforderlichen Abhängigkeiten mit der Bereitstellung installiert werden.

Example

```
~/eb-rails$ bundle lock --add-platform ruby
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies...
Writing lockfile to /Users/janedoe/EBDPT/RubyApps/eb-rails-doc-app/Gemfile.lock
```

Konfigurieren von Rails-Einstellungen

Konfigurieren Sie die Rails-Umgebungseigenschaften mit der Elastic Beanstalk-Konsole. Setzen Sie die `SECRET_KEY_BASE`-Umgebungseigenschaft auf eine Zeichenfolge mit bis zu 256 alphanumerischen Zeichen.

Rails verwendet diese Eigenschaft, um Schlüssel zu erstellen. Aus diesem Grund sollten Sie sie geheimhalten und nicht als Klartext in der Quellsteuerung speichern. Übergeben Sie sie stattdessen an Rails-Code in Ihrer Umgebung über eine Umgebungseigenschaft.

So konfigurieren Sie Umgebungsvariablen in der Elastic Beanstalk Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Scrollen Sie nach unten zu Runtime-Umgebungsvariablen.
6. Wählen Sie Umgebungsvariable hinzufügen aus.
7. Wählen Sie als Quelle die Option Klartext aus.

   > **Note**
   >
   > Die Werte Secrets Manager und SSM Parameter Store in der Drop-down-Liste dienen zur Konfiguration von Umgebungsvariablen als Secrets zum Speichern sensibler Daten wie Anmeldeinformationen und API-Schlüssel. Weitere Informationen finden Sie unter Elastic Beanstalk mit AWS Secrets Manager und AWS Systems Manager Parameter Store verwenden.

8. Geben Sie den Namen der Umgebungsvariablen und die Wertepaare der Umgebungsvariablen ein.
9. Wenn Sie weitere Variablen hinzufügen müssen, wiederholen Sie Schritt 6 bis Schritt 8.
10. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Jetzt können Sie die Website in Ihrer Umgebung bereitstellen.

Bereitstellen der Anwendung

Erstellen Sie ein Quell-Bundle mit den Dateien, die von Rails erstellt wurden. Der folgende Befehl erstellt ein Quell-Bundle mit dem Namen `rails-default.zip`.

```
~/eb-rails$ zip ../rails-default.zip -r * .[^.]*
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Rails in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung von der Konsole aus

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region

2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Nächste Schritte

Weitere Informationen über Rails finden Sie unter rubyonrails.org.

Im weiteren Verlauf der Anwendungsentwicklung können Sie Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die Elastic Beanstalk Command Line Interface (EB CLI) bietet easy-to-use Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk Beanstalk-Umgebungen von der Befehlszeile aus.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, konfigurieren Sie einen benutzerdefinierten Domainnamen für die Umgebung und aktivieren dann HTTPS für sichere Verbindungen.

Bereitstellen einer Sinatra-Anwendung in Elastic Beanstalk

Diese schrittweise Anleitung zeigt, wie eine einfache Sinatra-Webanwendung in AWS Elastic Beanstalk erstellt wird.

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole kennen. Sofern noch nicht geschehen, starten Sie anhand der Anleitung in Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können die erste Elastic-Beanstalk-Umgebung.

Für die Verfahren in diesem Leitfaden benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

Sinatra 2.1.0 erfordert Ruby 2.3.0 oder höher. In diesem Tutorial verwenden wir Ruby 3.0.2 und die entsprechende Elastic Beanstalk-Plattformversion. Installieren Sie Ruby mithilfe der Anleitungen unter Einrichtung Ihrer Ruby-Entwicklungsumgebung für Elastic Beanstalk.

Starten einer Elastic Beanstalk-Umgebung

Erstellen Sie mit der Elastic Beanstalk-Konsole, eine Elastic Beanstalk-Umgebung. Wählen Sie die Plattform Ruby und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole über diesen vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus und wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 Instanz — Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die für die Ausführung von Web-Apps auf der von Ihnen ausgewählten Plattform konfiguriert ist.

  Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverseproxy, der vor der Web-App platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Datenverkehr auf Port 80 zulässt. Diese Ressource ermöglicht es dem HTTP-Verkehr vom Load Balancer, die EC2 Instance zu erreichen, auf der Ihre Web-App ausgeführt wird. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer)– Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe — Eine EC2 Amazon-Sicherheitsgruppe, die so konfiguriert ist, dass sie eingehenden Verkehr auf Port 80 zulässt. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch CloudWatch Amazon-Alarme — Zwei Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.
- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *`subdomain region`*.elasticbeanstalk.com.

> **Domain-Sicherheit**
>
> Um die Sicherheit Ihrer Elastic-Beanstalk-Anwendungen zu erhöhen, ist die Domain elasticbeanstalk.com in der Public Suffix List (PSL) registriert.
> Wenn Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Elastic Beanstalk Beanstalk-Anwendungen einrichten müssen, empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, um die Sicherheit zu erhöhen. Diese Vorgehensweise schützt Ihre Domain vor Cross-Site-Request-Forgery-Versuchen (CSRF). Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

> **Note**
>
> Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter Verwenden von Elastic Beanstalk mit Amazon S3.

Schreiben einer einfachen Sinatra-Website

So können Sie eine Sinatra-Anwendung erstellen und bereitstellen:

1. Erstellen Sie eine Konfigurationsdatei mit dem Namen config.ru und folgendem Inhalt.

   Example config.ru

   ```
   require './helloworld'
   run Sinatra::Application
   ```

2. Erstellen Sie eine Ruby-Codedatei mit dem Namen helloworld.rb und folgendem Inhalt.

   Example helloworld.rb

   ```
   require 'sinatra'
   get '/' do
     "Hello World!"
   end
   ```

3. Erstellen Sie eine Gemfile-Datei mit folgendem Inhalt.

   Example Gemfile

   ```
   source 'https://rubygems.org'
   gem 'sinatra'
   gem 'puma'
   ```

4. Führen Sie die Bundle-Installation aus, um die `Gemfile.lock` zu generieren.

Example

```
~/eb-sinatra$ bundle install
Fetching gem metadata from https://rubygems.org/....
Resolving dependencies...
Using bundler 2.2.22
Using rack 2.2.3
...
```

5. Damit Elastic Beanstalk die Anwendung erfolgreich auf der Ruby-Plattform bereitstellen kann, aktualisieren Sie `Gemfile.lock`. Einige Abhängigkeiten von `Gemfile.lock` können plattformspezifisch sein. Deshalb müssen Sie **`platform ruby`** zu `Gemfile.lock` hinzufügen, damit alle erforderlichen Abhängigkeiten mit der Bereitstellung installiert werden.

   Example

   ```
   ~/eb-sinatra$ bundle lock --add-platform ruby
   Fetching gem metadata from https://rubygems.org/....
   Resolving dependencies...
   Writing lockfile to /Users/janedoe/EBDPT/RubyApps/eb-sinatra/Gemfile.lock
   ```

6. Erstellen Sie eine Procfile-Datei mit folgendem Inhalt.

   Example Procfile

   ```
   web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
   ```

Bereitstellen der Anwendung

Erstellen Sie ein Quell-Bundle mit Ihren Quelldateien. Der folgende Befehl erstellt ein Quell-Bundle mit dem Namen `sinatra-default.zip`.

```
~/eb-sinatra$ zip ../sinatra-default.zip -r * .[^.]*
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Sinatra in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Bereinigen

Nachdem Sie die Arbeit mit dem Demo-Code abgeschlossen haben, können Sie Ihre Umgebung beenden. Elastic Beanstalk löscht alle zugehörigen AWS Ressourcen wie EC2 Amazon-Instances, Datenbank-Instances, Load Balancer, Sicherheitsgruppen und Alarme.

Durch das Entfernen von Ressourcen wird die Elastic Beanstalk Beanstalk-Anwendung nicht gelöscht, sodass Sie jederzeit neue Umgebungen für Ihre Anwendung erstellen können.

So beenden Sie Ihre Elastic Beanstalk Beanstalk-Umgebung über die Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Nächste Schritte

Weitere Informationen über Sinatra finden Sie unter sinatrarb.com.

Im weiteren Verlauf der Anwendungsentwicklung können Sie Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die Elastic Beanstalk Command Line Interface (EB CLI) bietet easy-to-use Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk Beanstalk-Umgebungen von der Befehlszeile aus.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, konfigurieren Sie einen benutzerdefinierten Domainnamen für die Umgebung und aktivieren dann HTTPS für sichere Verbindungen.

Hinzufügen einer Amazon RDS-DB-Instance zu Ihrer Ruby Elastic Beanstalk Beanstalk-Umgebung

Dieses Thema enthält Anweisungen zum Erstellen eines Amazon RDS mit der Elastic-Beanstalk-Konsole. Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann an Ihre Umgebung gekoppelt sein und von Elastic Beanstalk verwaltet werden oder sie kann als entkoppelte Datenbank erstellt und extern von einem anderen Service verwaltet werden. In dieser Anleitung wird die Datenbank an Ihre Umgebung gekoppelt und von Elastic Beanstalk verwaltet. Weitere Informationen zur Integration eines Amazon RDS mit Elastic Beanstalk finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Sections

- Hinzufügen einer DB-Instance zu Ihrer Umgebung
- Herunterladen eines Adapters
- Herstellen einer Verbindung zu einer Datenbank

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftenwert
`RDS_HOSTNAME`	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
`RDS_PORT`	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
`RDS_DB_NAME`	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
`RDS_USERNAME`	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
`RDS_PASSWORD`	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.

Weitere Informationen zur Konfiguration einer Datenbank-Instance in Verbindung mit einer Elastic-Beanstalk-Umgebung finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Herunterladen eines Adapters

Fügen Sie den Datenbankadapter zur gem-Datei des Projekts hinzu.

Example Gemfile – Rails mit MySQL

```
source 'https://rubygems.org'
gem 'puma'
gem 'rails', '~> 6.1.4', '>= 6.1.4.1'
gem 'mysql2'
```

Gängige Adapter-Gems für Ruby

- MySQL – `mysql2`
- PostgreSQL – `pg`
- Oracle – `activerecord-oracle_enhanced-adapter`
- SQL Server – `activerecord-sqlserver-adapter`

Herstellen einer Verbindung zu einer Datenbank

Elastic Beanstalk stellt Verbindungsinformationen für zugehörige DB-Instances in Umgebungseigenschaften bereit. Mit `ENV['VARIABLE']` können Sie die Eigenschaften lesen und eine Datenbankverbindung konfigurieren.

Example config/database.yml – Ruby on Rails-Datenbankkonfiguration (MySQL)

```
production:
  adapter: mysql2
  encoding: utf8
  database: <%= ENV['RDS_DB_NAME'] %>
  username: <%= ENV['RDS_USERNAME'] %>
  password: <%= ENV['RDS_PASSWORD'] %>
  host: <%= ENV['RDS_HOSTNAME'] %>
  port: <%= ENV['RDS_PORT'] %>
```

Bereitstellung mit Docker-Containern auf Elastic Beanstalk

In diesem Kapitel wird erklärt, wie Sie Elastic Beanstalk verwenden können, um Webanwendungen aus Docker-Containern bereitzustellen. Docker-Container sind unabhängig und enthalten die Konfigurationsinformationen und Software, die Ihre Webanwendung für die Ausführung benötigt. Mit Docker-Containern können Sie Ihre eigene Laufzeitumgebung definieren. Sie können auch Ihre eigene Programmiersprache und Anwendungsabhängigkeiten wie Paketmanager oder Tools wählen, die normalerweise nicht von anderen Elastic Beanstalk-Plattformen unterstützt werden.

Folgen Sie den Schritten unterQuickStart für Docker, um eine Docker-Anwendung „Hello World“ zu erstellen und sie mithilfe der EB CLI in einer Elastic Beanstalk Beanstalk-Umgebung bereitzustellen.

Themen

- Filialen der Elastic Beanstalk Docker-Plattform
- Verwendung des Elastic Beanstalk Docker-Plattformzweigs
- Verwendung des von ECS verwalteten Docker-Plattformzweigs in Elastic Beanstalk
- Authentifizierung mit Image-Repositorys
- Konfiguration von Elastic Beanstalk Docker-Umgebungen
- Ältere Plattformen

Filialen der Elastic Beanstalk Docker-Plattform

Die Docker-Plattform für Elastic Beanstalk umfasst die folgenden Plattformzweige:

Docker läuft auf Amazon Linux 2 und Docker läuft AL2023

Elastic Beanstalk stellt Docker-Container und Quellcode auf EC2-Instances bereit und verwaltet sie. Diese Plattformzweige bieten Unterstützung für mehrere Container. Sie können auch das Docker-Compose-Tool nutzen, um das Konfigurieren, Testen und Bereitstellen von Anwendungen zu vereinfachen. Weitere Informationen zu diesem Plattformzweig finden Sie unter the section called “Docker-Plattformzweig”.

ECS läuft auf Amazon Linux 2 und ECS läuft auf AL2023

Wir bieten diesen Branch für Kunden an, die einen Migrationspfad AL2023 AL2 zur/von der ausgemusterten Plattformfiliale, auf der Multi-Container Docker läuft (Amazon Linux AMI), benötigen.

Die neuesten Plattformzweige unterstützen alle Funktionen aus dem eingestellten Plattformzweig. Es sind keine Änderungen am Quellcode erforderlich. Weitere Informationen finden Sie unter Migrieren Sie Ihre Elastic Beanstalk Beanstalk-Anwendung von ECS-verwaltetem Multi-Container-Docker auf ECS AL1 auf Amazon Linux 2023. Wenn Sie keine Elastic Beanstalk Beanstalk-Umgebung haben, die auf einem ECS-basierten Plattformzweig läuft, empfehlen wir Ihnen, den Plattformzweig Docker Running on 64bit zu verwenden. AL2023 Dies bietet einen einfacheren Ansatz und erfordert weniger Ressourcen.

Eine Liste der Softwarekomponentenversionen, die mit den einzelnen Plattformzweigen verknüpft sind, finden Sie unter Docker im Dokument Plattformen.AWS Elastic Beanstalk

Stillgelegte Plattformzweige, die auf Amazon Linux AMI (AL1) laufen

Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Erweitern Sie die folgenden Abschnitte, um mehr über die einzelnen eingestellten Plattformzweige und ihren Migrationspfad zum aktuellen Plattformzweig auf Amazon Linux 2 oder AL2023 (empfohlen) zu erfahren.

Docker (Amazon Linux AMI)

Dieser Plattformzweig kann ein Docker-Image bereitstellen, das in einer Dockerdatei oder einer `Dockerrun.aws.json`-v1-Definition beschrieben ist. Von diesem Docker-Plattformzweig wird pro Instance nur jeweils ein Container ausgeführt. Die nachfolgenden Plattformzweige, Docker läuft auf 64 Bit AL2023 und Docker läuft auf 64 Bit, Amazon Linux 2, unterstützen mehrere Docker-Container pro Instance.

Wir empfehlen Ihnen, Ihre Umgebungen mit dem neueren und unterstützten Plattformzweig Docker zu erstellen, der auf 64-Bit ausgeführt wird. AL2023 Anschließend können Sie Ihre Anwendung in die neu erstellte Umgebung migrieren. Weitere Informationen zum Erstellen dieser Umgebungen finden Sie unter the section called “Docker-Plattformzweig”. Weitere Informationen zur Migration finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Multi-Container-Docker (Amazon Linux AMI)

Dieser Plattformzweig verwendet Amazon ECS, um eine Bereitstellung mehrerer Docker-Container in einem Amazon-ECS-Cluster in einer Elastic-Beanstalk-Umgebung zu koordinieren. Wenn Sie derzeit diesen eingestellten Plattformzweig verwenden, sollten Sie zum neuesten Plattformzweig ECS auf Amazon Linux 2023 migrieren. Der neueste Plattformzweig unterstützt alle Funktionen aus diesem

außer Betrieb genommenen Plattformzweig. Es sind keine Änderungen am Quellcode erforderlich. Weitere Informationen finden Sie unter Migrieren Sie Ihre Elastic Beanstalk Beanstalk-Anwendung von ECS-verwaltetem Multi-Container-Docker auf ECS AL1 auf Amazon Linux 2023.

Vorkonfigurierte Docker-Container

Zusätzlich zu den zuvor genannten Docker-Plattformen gibt es auch den Zweig Preconfigured Docker GlassFish Platform, der auf dem Amazon Linux AMI-Betriebssystem () läuft. AL1

Dieser Plattformzweig wurde durch die Plattformzweige Docker auf 64-Bit AL2023 und Docker auf 64-Bit-Amazon Linux 2 ersetzt. Weitere Informationen finden Sie unter Bereitstellen einer GlassFish Anwendung auf der Docker-Plattform.

Verwendung des Elastic Beanstalk Docker-Plattformzweigs

In diesem Abschnitt wird beschrieben, wie Sie Ihr Docker-Image für den Start vorbereiten, wenn einer der Elastic Beanstalk-Plattformbranches (Docker) ausgeführt wird oder. AL2 AL2023

Folgen Sie den Schritten unterQuickStart für Docker, um eine Docker-Anwendung „Hello World“ zu erstellen und sie mithilfe der EB CLI in einer Elastic Beanstalk Beanstalk-Umgebung bereitzustellen.

Themen

- QuickStart: Stellen Sie eine Docker-Anwendung auf Elastic Beanstalk bereit
- QuickStart: Stellen Sie eine Docker Compose-Anwendung auf Elastic Beanstalk bereit
- Vorbereiten Ihres Docker-Images für die Bereitstellung auf Elastic Beanstalk

QuickStart: Stellen Sie eine Docker-Anwendung auf Elastic Beanstalk bereit

Dieses QuickStart Tutorial führt Sie durch den Prozess der Erstellung einer Docker-Anwendung und deren Bereitstellung in einer AWS Elastic Beanstalk Umgebung.

> ⚠ Nicht für Produktionszwecke
>
> Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine Docker-Anwendung und einen Container
- Schritt 2: Führen Sie Ihre Anwendung lokal aus
- Schritt 3: Stellen Sie Ihre Docker-Anwendung mit der EB CLI bereit
- Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus
- Schritt 5: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center -Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

EB-CLI

Dieses Tutorial verwendet die Befehlszeilenschnittstelle Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Docker

Um diesem Tutorial zu folgen, benötigen Sie eine funktionierende lokale Installation von Docker. Weitere Informationen finden Sie unter Docker erhalten auf der Docker-Docs-Website.

Stellen Sie sicher, dass der Docker-Daemon läuft, indem Sie den folgenden Befehl ausführen.

```
~$ docker info
```

Schritt 1: Erstellen Sie eine Docker-Anwendung und einen Container

In diesem Beispiel erstellen wir ein Docker-Image der Flask-Beispielanwendung, auf die auch in verwiesen wird. Bereitstellen einer Flask-Anwendung in Elastic Beanstalk

Die Anwendung besteht aus zwei Dateien:

- `app.py`— die Python-Datei, die den Code enthält, der im Container ausgeführt wird.
- `Dockerfile`— das Dockerfile zum Erstellen Ihres Images.

Platzieren Sie beide Dateien im Stammverzeichnis eines Verzeichnisses.

```
~/eb-docker-flask/
|-- Dockerfile
|-- app.py
```

Fügen Sie Ihrem die folgenden Inhalte hinzu`Dockerfile`.

Example~/eb-docker-flask/Dockerfile

```
FROM public.ecr.aws/docker/library/python:3.12
COPY . /app
WORKDIR /app
RUN pip install Flask==3.1.1
EXPOSE 5000
CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

Fügen Sie Ihrer `app.py` Datei die folgenden Inhalte hinzu.

Example~/eb-docker-flask/app.py

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello Elastic Beanstalk! This is a Docker application'
```

Verwenden Sie den Befehl docker build, um Ihr Container-Image lokal zu erstellen und das Image mit zu taggen. `eb-docker-flask` Der Punkt (`.`) am Ende des Befehls gibt an, dass der Pfad ein lokales Verzeichnis ist.

```
~/eb-docker-flask$ docker build -t eb-docker-flask .
```

Schritt 2: Führen Sie Ihre Anwendung lokal aus

Führen Sie Ihren Container mit dem Befehl docker run aus. Der Befehl gibt die ID des laufenden Containers aus. Die -d Option führt Docker im Hintergrundmodus aus. Die -p Option macht Ihre Anwendung auf Port 5000 verfügbar. Elastic Beanstalk stellt standardmäßig Traffic zu Port 5000 auf der Docker-Plattform bereit.

```
~/eb-docker-flask$ docker run -dp 127.0.0.1:5000:5000 eb-docker-flask
```

Navigieren Sie `http://127.0.0.1:5000/` in Ihrem Browser zu. Du solltest den Text „Hello Elastic Beanstalk! Dies ist eine Docker-Anwendung“.

Führen Sie den Befehl docker kill aus, um den Container zu beenden.

```
~/eb-docker-flask$ docker kill container-id
```

Schritt 3: Stellen Sie Ihre Docker-Anwendung mit der EB CLI bereit

Führen Sie die folgenden Befehle aus, um eine Elastic Beanstalk Beanstalk-Umgebung für diese Anwendung zu erstellen.

Um eine Umgebung zu erstellen und Ihre Docker-Anwendung bereitzustellen

1. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

```
~/eb-docker-flask$ eb init -p docker docker-tutorial --region us-east-2
Application docker-tutorial has been created.
```

Dieser Befehl erstellt eine Anwendung mit dem Namen `docker-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der neuesten Version der Docker-Plattform erstellt werden.

2. (Optional) Führen Sie eb init erneut aus, um ein Standardschlüsselpaar für die SSH-Verbindung zur EC2-Instance (auf der die Anwendung ausgeführt wird) zu konfigurieren.

```
~/eb-docker-flask$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein Schlüsselpaar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erstellt automatisch eine Zip-Datei für Ihre Anwendung und startet sie auf Port 5000.

```
~/eb-docker-flask$ eb create docker-tutorial
```

Es dauert etwa fünf Minuten, bis Elastic Beanstalk Ihre Umgebung erstellt hat.

Schritt 4: Führen Sie Ihre Anwendung auf Elastic Beanstalk aus

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, öffnen Sie Ihre Website mit. eb open

```
~/eb-docker-flask$ eb open
```

Herzlichen Glückwunsch! Sie haben eine Docker-Anwendung mit Elastic Beanstalk bereitgestellt! Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet.

Schritt 5: Bereinigen

Sie können Ihre Umgebung beenden, wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind. Elastic Beanstalk beendet alle AWS Ressourcen, die mit Ihrer Umgebung verknüpft sind.

Führen Sie den folgenden Befehl aus, um Ihre Elastic Beanstalk Beanstalk-Umgebung mit der EB-CLI zu beenden.

```
~/eb-docker-flask$ eb terminate
```

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt. Es dient als einfache Beispielanwendung mit einer einzigen EC2-Instance, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 instance (EC2-Instance) – Eine virtuelle Amazon EC2-Maschine zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 zugelassen wird. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Web-App erreichen. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.

- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen. Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

Wenn Sie eine oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, mit der lokalen Entwicklung und Ausführung von Docker-Anwendungen zu beginnen, finden Sie unterVorbereiten Ihres Docker-Images für die Bereitstellung auf Elastic Beanstalk.

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Beispielanwendung auch mit der Elastic Beanstalk Beanstalk-Konsole starten. Ausführliche Schritte finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

QuickStart: Stellen Sie eine Docker Compose-Anwendung auf Elastic Beanstalk bereit

Dieses QuickStart Tutorial führt Sie durch den Prozess der Erstellung einer Docker Compose-Anwendung mit mehreren Containern und deren Bereitstellung in einer Umgebung. AWS Elastic Beanstalk Sie erstellen eine Flask-Webanwendung mit einem Nginx-Reverse-Proxy, um zu demonstrieren, wie Docker Compose die Orchestrierung mehrerer Container vereinfacht.

> ⚠ Nicht für Produktionszwecke
>
> Beispiele dienen nur zur Veranschaulichung. Verwenden Sie keine Beispielanwendungen in der Produktion.

Sections

- Ihr AWS Konto
- Voraussetzungen
- Schritt 1: Erstellen Sie eine Docker Compose-Anwendung
- Schritt 2: Führen Sie Ihre Anwendung lokal aus
- Schritt 3: Stellen Sie Ihre Docker Compose-Anwendung mit der EB CLI bereit
- Schritt 4: Testen Sie Ihre Anwendung auf Elastic Beanstalk
- Schritt 5: Bereinigen
- AWS Ressourcen für Ihre Anwendung
- Nächste Schritte
- Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Ihr AWS Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie ein AWS Konto erstellen. Wenn Sie sich registrieren, können Sie auf Elastic Beanstalk und andere AWS Dienste zugreifen, die Sie benötigen.

Wenn Sie bereits ein AWS Konto haben, können Sie weitermachen zu. Voraussetzungen

Erstellen Sie ein AWS Konto

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung -Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center -Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center - Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center - Benutzerhandbuch.

Voraussetzungen

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine Shell für die Ausführung der Befehle. Befehle werden ggf. in Auflistungen mit einem vorangestellten Eingabeaufforderungssymbol ($) und dem Namen des aktuellen Verzeichnisses angezeigt.

```
~/eb-project$ this is a command
this is output
```

In Linux und macOS können Sie Ihren bevorzugten Shell- und Paket-Manager verwenden. Unter Windows können Sie das Windows-Subsystem für Linux installieren, um eine Windows-integrierte Version von Ubuntu und Bash zu erhalten.

EB-CLI

Dieses Tutorial verwendet die Befehlszeilenschnittstelle Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen) und Konfigurieren der EB CLI.

Docker und Docker Compose

Um diesem Tutorial zu folgen, benötigen Sie eine funktionierende lokale Installation von Docker und Docker Compose. Weitere Informationen finden Sie unter Docker herunterladen und Docker Compose installieren auf der Docker-Dokumentationswebsite.

Stellen Sie sicher, dass Docker und Docker Compose installiert sind und ausgeführt werden, indem Sie die folgenden Befehle ausführen.

```
~$ docker info
~$ docker compose version
```

Schritt 1: Erstellen Sie eine Docker Compose-Anwendung

In diesem Beispiel erstellen wir mit Docker Compose eine Multicontainer-Anwendung, die aus einer Flask-Webanwendung und einem Nginx-Reverse-Proxy besteht. Dies zeigt, wie Docker Compose die Orchestrierung mehrerer Container vereinfacht, die zusammenarbeiten.

Die Anwendung beinhaltet eine Konfiguration zur Gesundheitsüberwachung, die es Elastic Beanstalk ermöglicht, detaillierte Anwendungsmetriken von Ihrem Nginx-Proxy zu sammeln.

Die Anwendung besteht aus der folgenden Struktur:

```
~/eb-docker-compose-flask/
|-- docker-compose.yml
|-- web/
|   |-- Dockerfile
|   |-- app.py
|   `-- requirements.txt
|-- proxy/
|   |-- Dockerfile
|   `-- nginx.conf
`-- .platform/
    `-- hooks/
        `-- postdeploy/
            `-- 01_setup_healthd_permissions.sh
```

Erstellen Sie die Verzeichnisstruktur und fügen Sie die folgenden Dateien hinzu:

Erstellen Sie zunächst die `docker-compose.yml` Hauptdatei, die die Dienste und ihre Beziehungen definiert.

Example~/eb-docker-compose-flask/docker-compose.yml

```
services:
  web:
    build: ./web
    expose:
      - "5000"

  nginx-proxy:
    build: ./proxy
    ports:
      - "80:80"
    volumes:
      - "/var/log/nginx:/var/log/nginx"
    depends_on:
      - web
```

Erstellen Sie die Flask-Webanwendung im `web` Verzeichnis. Fügen Sie Ihrer `app.py` Datei den folgenden Inhalt hinzu.

Example~/eb-docker-compose-flask/web/app.py

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello Elastic Beanstalk! This is a Docker Compose application'
```

Fügen Sie Ihrem Webservice die folgenden Inhalte hinzu`Dockerfile`.

Example~/eb-docker-compose-flask/web/Dockerfile

```
FROM public.ecr.aws/docker/library/python:3.12
COPY . /app
WORKDIR /app
RUN pip install Flask==3.1.1
EXPOSE 5000
```

```
CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

Erstellen Sie den Nginx-Reverse-Proxy im `proxy` Verzeichnis. Fügen Sie Ihrer Datei den folgenden Inhalt hinzu`nginx.conf`.

Diese Konfiguration beinhaltet eine Einrichtung zur Gesundheitsüberwachung, die es Elastic Beanstalk ermöglicht, detaillierte Anwendungsmetriken zu sammeln. Weitere Informationen zur Anpassung der Protokollformate für die Gesundheitsüberwachung finden Sie unter. Format der Protokolle der erweiterten Zustandsberichte

Example~/eb-docker-compose-flask/proxy/nginx.conf

```
events {
    worker_connections 1024;
}

http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    map $http_upgrade $connection_upgrade {
        default       "upgrade";
    }

    # Health monitoring log format for Elastic Beanstalk
    log_format healthd
 '$msec"$uri"$status"$request_time"$upstream_response_time"$http_x_forwarded_for';

    upstream flask_app {
        server web:5000;
    }

    server {
        listen 80 default_server;
        root /usr/share/nginx/html;

        # Standard access log
        access_log /var/log/nginx/access.log;

        # Health monitoring log for Elastic Beanstalk
        if ($time_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})T(\d{2})") {
            set $year $1;
            set $month $2;
```

```
            set $day $3;
            set $hour $4;
        }
        access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
 healthd;

        location / {
            proxy_pass http://flask_app;
            proxy_http_version    1.1;

            proxy_set_header    Connection            $connection_upgrade;
            proxy_set_header    Upgrade               $http_upgrade;
            proxy_set_header    Host                  $host;
            proxy_set_header    X-Real-IP             $remote_addr;
            proxy_set_header    X-Forwarded-For       $proxy_add_x_forwarded_for;
        }
    }
}
```

Fügen Sie Ihrem Proxydienst `Dockerfile` die folgenden Inhalte hinzu.

Example~/eb-docker-compose-flask/proxy/Dockerfile

```
FROM public.ecr.aws/nginx/nginx:alpine
COPY nginx.conf /etc/nginx/nginx.conf
EXPOSE 80
```

Erstellen Sie abschließend ein Plattform-Hook-Skript, um die erforderlichen Protokollverzeichnisse und Berechtigungen für die Zustandsüberwachung einzurichten. Plattform-Hooks ermöglichen es Ihnen, während des Bereitstellungsprozesses benutzerdefinierte Skripts auszuführen. Weitere Informationen zu Plattform-Hooks finden Sie unterPlattform-Hooks.

Example~/eb-docker-compose-flask/.platform/hooks/postdeploy/01_setup_healthd_permissions.sh

```
#!/bin/bash
set -ex

NGINX_CONTAINER=$(docker ps --filter "name=nginx-proxy" -q)

if [ -z "$NGINX_CONTAINER" ]; then
    echo "Error: No nginx-proxy container found running"
    exit 1
fi
```

```
NGINX_UID=$(docker exec ${NGINX_CONTAINER} id -u nginx)
NGINX_GID=$(docker exec ${NGINX_CONTAINER} id -g nginx)

mkdir -p /var/log/nginx/healthd
chown -R ${NGINX_UID}:${NGINX_GID} /var/log/nginx
```

Schritt 2: Führen Sie Ihre Anwendung lokal aus

Verwenden Sie den Befehl docker compose up, um Ihre Multicontainer-Anwendung lokal zu erstellen und auszuführen. Docker Compose erstellt beide Container-Images und startet die in Ihrer Datei definierten Dienste. `docker-compose.yml`

```
~/eb-docker-compose-flask$ docker compose up --build
```

Die --build Option stellt sicher, dass Docker Compose die Container-Images erstellt, bevor die Dienste gestartet werden. Sie sollten eine Ausgabe sehen, die zeigt, dass sowohl der Webdienst als auch der Nginx-Proxy-Service gestartet werden.

Navigieren Sie `http://localhost` in Ihrem Browser zu. Du solltest den Text „Hello Elastic Beanstalk! Dies ist eine Docker Compose-Anwendung“. Der Nginx-Proxy empfängt Ihre Anfrage auf Port 80 und leitet sie an die Flask-Anwendung weiter, die auf Port 5000 läuft.

Wenn Sie mit dem Testen fertig sind, beenden Sie die Anwendung, indem Sie Ctrl+C im Terminal drücken, oder führen Sie den folgenden Befehl in einem separaten Terminal aus:

```
~/eb-docker-compose-flask$ docker compose down
```

Schritt 3: Stellen Sie Ihre Docker Compose-Anwendung mit der EB CLI bereit

Führen Sie die folgenden Befehle aus, um eine Elastic Beanstalk Beanstalk-Umgebung für diese Anwendung zu erstellen.

Um eine Umgebung zu erstellen und Ihre Docker Compose-Anwendung bereitzustellen

1. Initialisieren Sie mit dem Befehl eb init das EB CLI-Repository.

   ```
   ~/eb-docker-compose-flask$ eb init -p docker docker-compose-tutorial --region us-east-2
   Application docker-compose-tutorial has been created.
   ```

Dieser Befehl erstellt eine Anwendung mit dem Namen `docker-compose-tutorial` und konfiguriert Ihr lokales Repository so, dass Umgebungen mit der neuesten Version der Docker-Plattform erstellt werden.

2. (Optional) Führen Sie eb init erneut aus, um ein Standardschlüsselpaar für die SSH-Verbindung zur EC2-Instance (auf der die Anwendung ausgeführt wird) zu konfigurieren.

```
~/eb-docker-compose-flask$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein Schlüsselpaar zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erkennt Ihre `docker-compose.yml` Datei automatisch und stellt Ihre Multi-Container-Anwendung bereit.

```
~/eb-docker-compose-flask$ eb create docker-compose-tutorial
```

Elastic Beanstalk benötigt etwa fünf Minuten, um Ihre Umgebung zu erstellen und Ihre Multi-Container-Anwendung bereitzustellen.

Schritt 4: Testen Sie Ihre Anwendung auf Elastic Beanstalk

Wenn der Prozess zur Erstellung Ihrer Umgebung abgeschlossen ist, öffnen Sie Ihre Website mit. eb open

```
~/eb-docker-compose-flask$ eb open
```

Großartig! Sie haben eine Docker Compose-Anwendung mit mehreren Containern mit Elastic Beanstalk bereitgestellt! Es wird ein Browserfenster mit dem für die Anwendung erstellten Domainnamen geöffnet. Sie sollten die Nachricht von Ihrer Flask-Anwendung sehen, die über den Nginx-Reverse-Proxy bereitgestellt wird.

Schritt 5: Bereinigen

Sie können Ihre Umgebung beenden, wenn Sie mit der Arbeit an Ihrer Anwendung fertig sind. Elastic Beanstalk beendet alle AWS Ressourcen, die mit Ihrer Umgebung verknüpft sind.

Führen Sie den folgenden Befehl aus, um Ihre Elastic Beanstalk Beanstalk-Umgebung mit der EB-CLI zu beenden.

```
~/eb-docker-compose-flask$ eb terminate
```

AWS Ressourcen für Ihre Anwendung

Sie haben gerade eine Einzelinstanzanwendung erstellt, auf der mehrere Container ausgeführt werden. Es dient als einfache Beispielanwendung mit einer einzigen EC2-Instance, sodass kein Lastenausgleich oder auto Skalierung erforderlich sind. Für Einzelinstanzanwendungen erstellt Elastic Beanstalk die folgenden Ressourcen: AWS

- EC2 instance (EC2-Instance) – Eine virtuelle Amazon EC2-Maschine zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

  Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 zugelassen wird. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Web-App erreichen. Standardmäßig ist Datenverkehr ist auf anderen Ports nicht zulässig.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- CloudWatch Amazon-Alarme — Zwei CloudWatch Alarme, die die Auslastung der Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- CloudFormation stack — Elastic Beanstalk verwendet CloudFormation , um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu propagieren. Die Ressourcen werden in einer Vorlage definiert, die Sie in der CloudFormation -Konsole anzeigen können.

- Domainname — Ein Domainname, der in der Form an Ihre Web-App weitergeleitet wird. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet. Ihre Docker Compose-Anwendung führt mehrere Container auf einer einzigen EC2-Instance aus, wobei Elastic Beanstalk die Orchestrierung automatisch übernimmt.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen. Sie können Ihre neue Umgebung auch mit der Elastic Beanstalk Beanstalk-Konsole erkunden. Ausführliche Schritte finden Sie unter Erkunden Sie Ihre Umgebung im Kapitel Erste Schritte dieses Handbuchs.

Nachdem Sie ein oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, mit der lokalen Entwicklung und Ausführung von Docker Compose-Anwendungen zu beginnen, finden Sie unterVorbereiten Ihres Docker-Images für die Bereitstellung auf Elastic Beanstalk.

Bereitstellung mit der Elastic Beanstalk Beanstalk-Konsole

Sie können auch die Elastic Beanstalk Beanstalk-Konsole verwenden, um eine Docker Compose-Anwendung zu starten. Erstellen Sie eine ZIP-Datei, die Ihre `docker-compose.yml` Datei und alle zugehörigen Verzeichnisse und Dateien enthält, und laden Sie sie dann hoch, wenn Sie eine neue Anwendung erstellen. Eine ausführliche Anleitung dazu finden Sie unter Erstellen einer Beispielanwendung im Kapitel Erste Schritte dieses Handbuchs.

Vorbereiten Ihres Docker-Images für die Bereitstellung auf Elastic Beanstalk

In diesem Abschnitt wird beschrieben, wie Sie Ihr Docker-Image für die Bereitstellung auf Elastic Beanstalk vorbereiten, wobei entweder der Docker Running oder der Platform Branches läuft AL2 . AL2023 Welche Konfigurationsdateien Sie benötigen, hängt davon ab, ob Ihre Images lokal oder remote sind und ob Sie Docker Compose verwenden.

> **Note**
>
> Ein Beispiel für ein Verfahren, mit dem eine Docker-Umgebung gestartet wird, finden Sie im QuickStart für Docker Thema.

Themen

- Verwaltung Ihrer Bilder mit Docker Compose in Elastic Beanstalk
- Verwaltung von Bildern ohne Docker Compose in Elastic Beanstalk
- Erstellen benutzerdefinierter Images mit einer Dockerfile-Datei

Verwaltung Ihrer Bilder mit Docker Compose in Elastic Beanstalk

Sie können Docker Compose verwenden, um verschiedene Dienste in einer YAML-Datei zu verwalten. Weitere Informationen zu Docker Compose finden Sie unter Warum Compose verwenden? auf der Docker-Website.

- Erstellen Sie einen `docker-compose.yml`. Diese Datei ist erforderlich, wenn Sie Docker Compose verwenden, um Ihre Anwendung mit Elastic Beanstalk zu verwalten. Wenn all Ihre Deployments aus Images in öffentlichen Repositorys stammen, sind keine weiteren Konfigurationsdateien erforderlich. Wenn sich die Quell-Images Ihrer Bereitstellung in einem privaten Repository befinden, müssen Sie einige zusätzliche Konfigurationen vornehmen. Weitere Informationen finden Sie unter Verwenden von Bildern aus einem privaten Repository. Weitere Informationen zur `docker-compose.yml`-Datei finden Sie auf der Docker-Website in der Referenz zu Compose-Dateien.
- Die `Dockerfile` ist optional. Erstellen Sie eines, wenn Sie Elastic Beanstalk benötigen, um ein lokales benutzerdefiniertes Image zu erstellen und auszuführen. Weitere Informationen zur `Dockerfile`-Datei finden Sie auf der Docker-Website in der Dockerfile-Referenz.
- Möglicherweise müssen Sie eine `.zip` Datei erstellen. Wenn Sie nur eine `Dockerfile` Datei für die Bereitstellung Ihrer Anwendung verwenden, müssen Sie keine erstellen. Wenn Sie zusätzliche Konfigurationsdateien verwenden, muss die ZIP-Datei die `docker-compose.yml` Datei`Dockerfile`, Ihre Anwendungsdateien und alle Abhängigkeiten von Anwendungsdateien enthalten. Die `Dockerfile` und die `docker-compose.yml` müssen sich im Stammverzeichnis oder auf der obersten Ebene des ZIP-Archivs befinden. Wenn Sie die EB-CLI verwenden, um Ihre Anwendung bereitzustellen, erstellt sie eine ZIP-Datei für Sie.

Weitere Informationen zu Docker Compose und zur Installation von Docker Compose finden Sie auf den Docker-Websites Overview of Docker Compose und Install Docker Compose.

Verwaltung von Bildern ohne Docker Compose in Elastic Beanstalk

Wenn Sie Docker Compose nicht zur Verwaltung Ihrer Docker-Images verwenden, müssen Sie eine`Dockerfile`, eine Datei oder beides konfigurieren. `Dockerrun.aws.json`

- Erstellen Sie ein`Dockerfile`, um Elastic Beanstalk ein benutzerdefiniertes Image erstellen und lokal ausführen zu lassen.
- Erstellen Sie eine `Dockerrun.aws.json` v1-Datei, um ein Docker-Image aus einem gehosteten Repository für Elastic Beanstalk bereitzustellen.
- Möglicherweise müssen Sie eine `.zip` Datei erstellen. Wenn Sie nur eine der beiden Dateien, die `Dockerfile` oder die`Dockerrun.aws.json`, verwenden, müssen Sie keine ZIP-Datei erstellen. Wenn Sie beide Dateien verwenden, benötigen Sie eine ZIP-Datei. Die ZIP-Datei muss sowohl das als auch das enthalten`Dockerrun.aws.json`, zusammen mit der Datei, die Ihre Anwendungsdateien `Dockerfile` und alle Abhängigkeiten von Anwendungsdateien enthält. Wenn Sie die EB CLI verwenden, um Ihre Anwendung bereitzustellen, wird automatisch eine `.zip`-Datei erstellt.

`Dockerrun.aws.json`v1-Konfigurationsdatei

Eine `Dockerrun.aws.json`-Datei beschreibt, wie ein Remote-Docker-Image als Elastic Beanstalk-Anwendung bereitgestellt wird. Diese JSON-Datei ist spezifisch für Elastic Beanstalk. Wenn Ihre Anwendung auf einem Bild ausgeführt, das in einem gehosteten Repository verfügbar ist, können Sie das Image in einer `Dockerrun.aws.json` v1-Datei angeben und die `Dockerfile` auslassen.

> **`Dockerrun.aws.json`**-Versionen
>
> Der `AWSEBDockerrunVersion`-Parameter gibt die Version der `Dockerrun.aws.json`-Datei an.
>
> - Der Docker AL2 und die AL2023 Plattformen verwenden die folgenden Versionen der Datei.
>   - `Dockerrun.aws.json` v3— Umgebungen, die Docker Compose verwenden.
>   - `Dockerrun.aws.json` v1— Umgebungen, die Docker Compose nicht verwenden.

- ECS, das auf Amazon Linux 2 ausgeführt wird, und ECS, das auf läuft, AL2023 verwendet die `Dockerrun.aws.json v2` Datei. Die ausgemusterte Plattform ECS-The Multicontainer Docker Amazon Linux AMI (AL1) verwendete ebenfalls dieselbe Version.

Dockerrun.aws.json-v1

Zu den gültigen Schlüsseln und Werten für die `Dockerrun.aws.json v1`-Datei zählen unter anderem folgende Vorgänge:

AWSEBDockerrunVersion

(Erforderlich) Geben Sie die Versionsnummer an, 1 wenn Sie Docker Compose nicht zur Verwaltung Ihres Images verwenden.

Authentifizierung

(Nur erforderlich für private Repositorys) Gibt das Amazon S3-Objekt an, in dem die `.dockercfg`-Datei gespeichert ist.

Weitere Informationen finden Sie Authentifizierung mit Image-Repositorys unter Verwenden von Bildern aus einem privaten Repository weiter unten in diesem Kapitel.

Image

Gibt das Docker-Basis-Image auf einem vorhandenen Docker-Repository an, aus dem Sie einen Docker-Container erstellt haben. Geben Sie den Wert des Namensschlüssels im Format *`<organization>/<image name>`* für Bilder auf Docker Hub oder *`<site>/<organization name>/<image name>`* für andere Websites an.

Wenn Sie ein Image in der `Dockerrun.aws.json`-Datei angeben, wird `docker pull` von jeder Instance Ihrer Elastic Beanstalk-Umgebung ausgeführt, um das Image auszuführen. Sie können optional den Update-Schlüssel einschließen. Der Standardwert ist `true`. Er weist Elastic Beanstalk an, das Repository zu prüfen, Updates auf das Image abzurufen und alle zwischengespeicherten Images zu überschreiben.

Geben Sie, wenn Sie eine `Dockerfile` verwenden, den Image-Schlüssel in der `Dockerrun.aws.json`-Datei nicht an. Elastic Beanstalk erstellt und verwendet immer das Image, das in `Dockerfile` Beschrieben ist, wenn eines vorhanden ist.

Ports

(Erforderlich, wenn Sie den Image-Schlüssel angeben) Listet die Ports auf, die auf dem Docker-Container bereitgestellt werden sollen. Elastic Beanstalk verwendet den ContainerPortWert, um den Docker-Container mit dem Reverse-Proxy zu verbinden, der auf dem Host ausgeführt wird.

Sie können mehrere Container-Ports angeben, von Elastic Beanstalk wird aber nur der erste Port verwendet. Dieser Port wird verwendet, um Ihren Container mit dem Reverse-Proxy des Hosts zu verbinden und Anforderungen aus dem öffentlichen Internet weiterzuleiten. Wenn Sie a verwenden**`Dockerfile`**, sollte der erste ContainerPortWert mit dem ersten Eintrag in der 's-EXPOSE-Liste **`Dockerfile`** übereinstimmen.

Optional können Sie eine Liste von Anschlüssen in angeben HostPort. HostPortEinträge geben die Host-Ports an, denen ContainerPortWerte zugeordnet sind. Wenn Sie keinen HostPortWert angeben, wird standardmäßig der ContainerPortWert verwendet.

```
{
  "Image": {
    "Name": "image-name"
  },
  "Ports": [
    {
      "ContainerPort": 8080,
      "HostPort": 8000
    }
  ]
}
```

Datenträger

Ordnen Sie Volumes von einer EC2-Instance Ihrem Docker-Container zu. Geben Sie ein oder mehrere Volume-Arrays für die Zuordnung an.

```
{
  "Volumes": [
    {
      "HostDirectory": "/path/inside/host",
      "ContainerDirectory": "/path/inside/container"
    }
  ]
```

```
...
```

Protokollierung

Geben Sie das Verzeichnis im Container an, in das Ihre Anwendung Protokolle schreibt. Elastic Beanstalk lädt alle Protokolle in dieses Verzeichnis in Amazon S3 hoch, wenn Sie Protokollfragmente oder Bundle-Protokolle anfordern. Wenn Sie Protokolle zu einem Ordner mit dem Namen `rotated` in diesem Verzeichnis rotieren, können Sie auch Elastic Beanstalk so konfigurieren, dass rotierte Protokolle für eine dauerhafte Speicherung in Amazon S3 hochgeladen werden. Weitere Informationen finden Sie unter Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen.

Befehl

Geben Sie einen Befehl an, der im Container ausgeführt werden soll. Wenn Sie einen Entrypoint (Eintrittspunkt) angeben, dann wird Command (Befehl) als Argument für Entrypoint (Eintrittspunkt) hinzugefügt. Weitere Informationen finden Sie unter CMD in der Docker-Dokumentation.

Eintrittspunkt

Geben Sie einen Standard-Befehl an, der ausgeführt werden soll, wenn der Container gestartet wird. Weitere Informationen finden Sie unter EINTRITTSPUNKT in der Docker-Dokumentation.

Der folgende Codeausschnitt ist ein Beispiel, das die Syntax der `Dockerrun.aws.json`-Datei für einen einzelnen Container.

```
{
  "AWSEBDockerrunVersion": "1",
  "Image": {
    "Name": "janedoe/image",
    "Update": "true"
  },
  "Ports": [
    {
      "ContainerPort": "1234"
    }
  ],
  "Volumes": [
    {
      "HostDirectory": "/var/app/mydb",
      "ContainerDirectory": "/etc/mysql"
```

```
    }
  ],
  "Logging": "/var/log/nginx",
  "Entrypoint": "/app/bin/myapp",
  "Command": "--argument"
}>
```

Sie können Elastic Beanstalk nur mit der `Dockerrun.aws.json`-Datei oder mit einem `.zip`-Archiv bereitstellen, das sowohl die `Dockerrun.aws.json`-Datei als auch die `Dockerfile`-Datei enthält. Wenn Sie beide Dateien bereitstellen, beschreibt die `Dockerfile` das Docker-Image und die Datei `Dockerrun.aws.json` bietet zusätzliche Informationen für die Bereitstellung. Einzelheiten dazu finden Sie weiter unten in diesem Abschnitt.

> **Note**
>
> Die beiden Dateien müssen sich im Stammverzeichnis oder im oberste Level des `.zip`-Archivs befinden. Erstellen Sie kein Archiv aus einem Verzeichnis, das die Dateien enthält. Navigieren Sie stattdessen zu dem Verzeichnis und erstellen Sie das Archiv dort.
> Wenn Sie die beide Dateien bereitstellen, geben Sie kein Image in der Datei `Dockerrun.aws.json` an. Elastic Beanstalk erstellt und verwendet das Image, das in `Dockerfile` beschrieben wird, und ignoriert das in der Datei `Dockerrun.aws.json` angegebene Image.

Erstellen benutzerdefinierter Images mit einer Dockerfile-Datei

Falls Sie noch nicht über ein in einem Repository gehostetes Image verfügen, müssen Sie eine `Dockerfile` erstellen.

Der folgende Codeausschnitt ist ein Beispiel für die `Dockerfile`. Wenn Sie die Anweisungen in QuickStart für Docker befolgen, können Sie die `Dockerfile` wie beschrieben hochladen. Elastic Beanstalk führt das Spiel 2048 aus, wenn Sie diese verwende `Dockerfile`.

Weitere Informationen zu den Anweisungen, die Sie in die `Dockerfile` einschließen können, finden Sie unter Dockerfile-Referenz auf der Docker-Website.

```
FROM ubuntu:12.04

RUN apt-get update
```

```
RUN apt-get install -y nginx zip curl

RUN echo "daemon off;" >> /etc/nginx/nginx.conf
RUN curl -o /usr/share/nginx/www/master.zip -L https://codeload.github.com/
gabrielecirulli/2048/zip/master
RUN cd /usr/share/nginx/www/ && unzip master.zip && mv 2048-master/* . && rm -rf 2048-
master master.zip

EXPOSE 80

CMD ["/usr/sbin/nginx", "-c", "/etc/nginx/nginx.conf"]
```

> **Note**
>
> Sie können mehrstufige Builds aus einer einzigen Dockerfile ausführen, um kleinere Images mit einer signifikanten Verringerung der Komplexität zu erzeugen. Weitere Informationen finden Sie unter Verwenden mehrstufiger Builds auf der Docker-Dokumentationswebsite.

Verwendung des von ECS verwalteten Docker-Plattformzweigs in Elastic Beanstalk

Dieses Thema bietet einen Überblick über die von Elastic Beanstalk ECS verwalteten Docker-Plattformbranches für Amazon Linux 2 und Amazon Linux 2023. Es enthält auch Konfigurationsinformationen, die für die von Docker ECS verwaltete Plattform spezifisch sind.

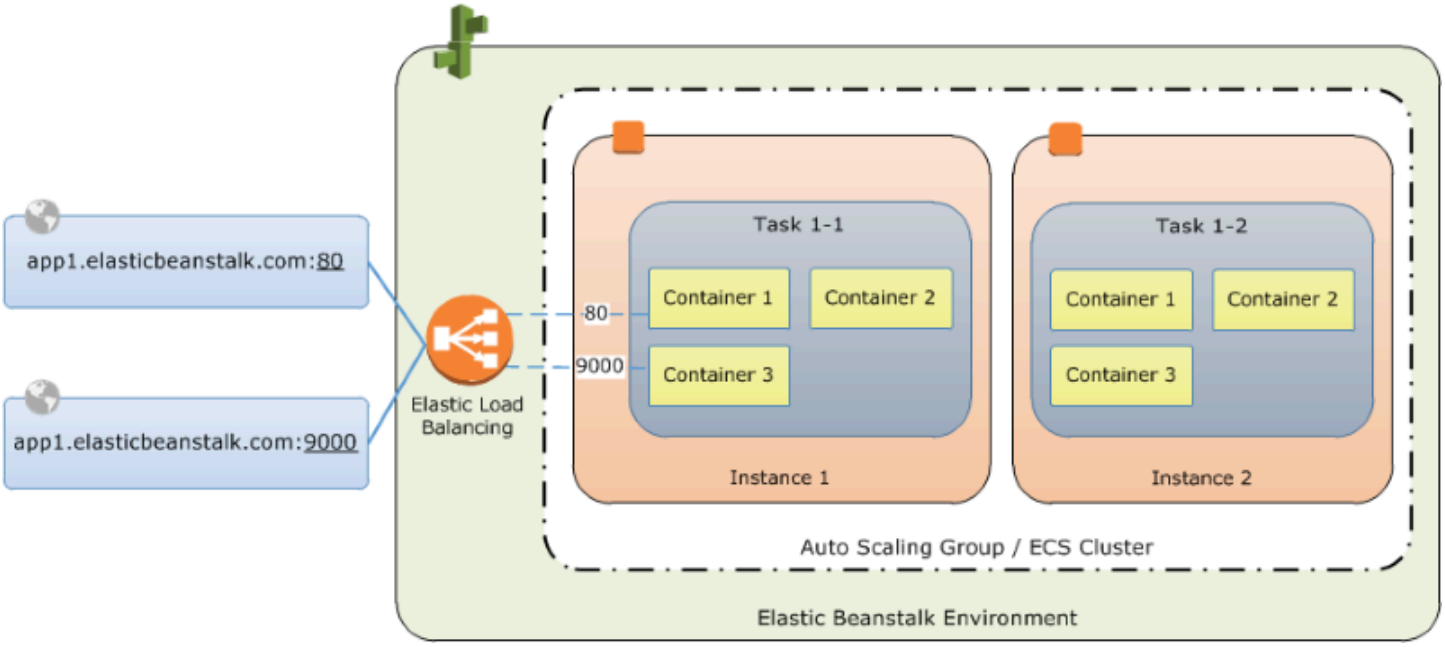
Migration von Multi-Container-Docker auf AL1

Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Dieses Kapitel enthält zwar Informationen zur Konfiguration dieser ausgemusterten Plattform, wir empfehlen jedoch dringend, auf den neuesten unterstützten Plattformzweig zu migrieren. Wenn Sie derzeit den ausgemusterten Multi-Container-Docker verwenden, der auf AL1 Platform Branch läuft, können Sie auf die neueste Version von ECS Running on AL2023 Platform Branch migrieren. Der neueste Plattformzweig unterstützt alle Funktionen dieses außer Betrieb genommenen Plattformzweigs. Es sind keine Änderungen am Quellcode erforderlich. Weitere Informationen finden Sie unter Migrieren Sie Ihre Elastic Beanstalk Beanstalk-Anwendung von ECS-verwaltetem Multi-Container-Docker auf ECS AL1 auf Amazon Linux 2023.

Überblick über die von ECS verwaltete Docker-Plattform

Elastic Beanstalk verwendet Amazon Elastic Container Service (Amazon ECS), um Container-Bereitstellungen in ECS-verwalteten Docker-Umgebungen zu koordinieren. Amazon ECS bietet Tools zum Verwalten eines Instance-Clusters mit Docker-Containern. Elastic Beanstalk übernimmt Amazon ECS-Aufgaben wie Clustererstellung, Aufgabendefinition und -ausführung. Alle Instances in der Umgebung werden im gleichen Satz von Containern ausgeführt, die in einer `Dockerrun.aws.json`-v2-Datei definiert sind. Um Docker bestmöglich zu nutzen, können Sie mit Elastic Beanstalk eine Umgebung erstellen, in der Ihre Amazon EC2-Instances mehrere Docker-Container parallel ausführen.

Das folgende Diagramm zeigt ein Beispiel für eine Elastic Beanstalk-Umgebung, die mit drei Docker-Containern konfiguriert ist, die auf jeder Amazon EC2-Instance in einer Auto Scaling-Gruppe ausgeführt werden:



Von Elastic Beanstalk erstellte Amazon ECS-Ressourcen

Wenn Sie eine Umgebung mit der ECS-verwalteten Docker-Plattform erstellen, erstellt und konfiguriert Elastic Beanstalk automatisch verschiedene Ressourcen von Amazon Elastic Container Service beim Entwickeln der Umgebung. Dabei werden die erforderlichen Container auf jeder Amazon-EC2-Instance erstellt.

- Amazon ECS-Cluster – Container-Instances in Amazon ECS sind in Cluster organisiert. Bei Verwendung mit Elastic Beanstalk wird ein Cluster immer für jede ECS-verwaltete Docker-

Umgebung erstellt. Ein ECS-Cluster enthält auch Auto Scaling Scaling-Gruppenkapazitätsanbieter und andere Ressourcen.

- Amazon-ECS-Aufgabendefinition – Elastic Beanstalk nutzt `Dockerrun.aws.json` v2 in Ihrem Projekt zum Generieren der Amazon-ECS-Aufgabendefinition, die für die Konfiguration der Container-Instances in der Umgebung verwendet wird.
- Amazon ECS-Aufgabe – Elastic Beanstalk kommuniziert mit Amazon ECS über die Ausführung einer Aufgabe auf allen Instances in der Umgebung zum Koordinieren der Container-Bereitstellung. In einer skalierbaren Umgebung initiiert Elastic Beanstalk eine neue Aufgabe, wenn eine Instance zum Cluster hinzugefügt wird.
- Amazon ECS Container-Agent – Der Agent wird in einem Docker-Container auf den Instances in Ihrer Umgebung ausgeführt. Der Agent ruft den Amazon ECS Service ab und wartet, bis eine Aufgabe ausgeführt wird.
- Amazon ECS Data Volumes — Zusätzlich zu den Volumes, die Sie in `Dockerrun.aws.json` Version 2 definieren, fügt Elastic Beanstalk Volume-Definitionen in die Aufgabendefinition ein, um die Protokollerfassung zu erleichtern.

  Elastic Beanstalk erstellt Protokoll-Volumes auf der Container-Instance, eines für jeden Container, unter `/var/log/containers/`*`containername`*. Diese Volumes haben den Namen `awseb-logs-`*`containername`* und werden für Container für das Mounting bereitgestellt. Details zum Mounting finden Sie unter Format der Containerdefinition.

Weitere Informationen zu Amazon ECS-Ressourcen finden Sie im Amazon Elastic Container Service Developer Guide.

`Dockerrun.aws.json`v2-Datei

Die Container-Instances benötigen eine Konfigurationsdatei mit dem Namen`Dockerrun.aws.json`. Container-Instances beziehen sich auf Amazon EC2 EC2-Instances, auf denen ECS-verwaltetes Docker in einer Elastic Beanstalk Beanstalk-Umgebung ausgeführt wird. Diese Datei gilt nur für Elastic Beanstalk und kann eigenständig oder in Kombination mit Quellcode und Inhalten in einem Quell-Bundle verwendet werden,um eine Umgebung auf einer Docker-Plattform zu erstellen.

> **Note**
>
> Version 2 von `Dockerrun.aws.json` fügt Support für mehrere Container pro Amazon-EC2-Instance hinzu und kann nur mit der ECS-verwalteten Docker-Plattform verwendet werden. Das Format unterscheidet sich erheblich von den anderen Versionen der

Konfigurationsdateien, die die Zweige der Docker-Plattform unterstützen, die nicht von ECS verwaltet werden.

Einzelheiten `Dockerrun.aws.json v2` zum aktualisierten Format und eine Beispieldatei finden Sie im.

Docker-Images

Die von ECS verwaltete Docker-Plattform für Elastic Beanstalk erfordert, dass Images vorkonfiguriert und in einem öffentlichen oder privaten Online-Image-Repository gespeichert werden, bevor eine Elastic Beanstalk Beanstalk-Umgebung erstellt werden kann.

> **Note**
>
> Das Erstellen benutzerdefinierter Images während der Bereitstellung mit einer `Dockerfile` wird von der ECS-verwalteten Docker-Plattform auf Elastic Beanstalk nicht unterstützt. Erstellen Sie Ihre Images und stellen Sie sie in einem Online-Repository bereit, bevor Sie eine Elastic Beanstalk-Umgebung erstellen.

Geben Sie Images nach Namen in `Dockerrun.aws.json` v2 an.

Zum Konfigurieren von Elastic Beanstalk für die Authentifizierung an ein privates Repository fügen Sie den `authentication`-Parameter in die `Dockerrun.aws.json`-v2-Datei ein.

Fehlgeschlagene Container-Bereitstellungen

Wenn eine Amazon ECS-Aufgabe fehlschlägt, werden ein oder mehrere Container in Ihrer Elastic Beanstalk-Umgebung nicht gestartet. Elastic Beanstalk setzt Multi-Container-Umgebungen aufgrund einer fehlgeschlagenen Amazon-ECS-Aufgabe nicht zurück. Wenn ein Container in der Umgebung nicht startet, stellen Sie die aktuelle Version oder eine frühere funktionierende Version mit der Elastic Beanstalk-Konsole erneut bereit.

So stellen Sie eine vorhandene Version bereit

1. Öffnen Sie die Elastic Beanstalk-Konsole in der Region Ihrer Umgebung.
2. Klicken Sie auf Actions (Aktionen) rechts neben Ihrem Anwendungsnamen und klicken Sie dann auf Anwendungsversionen anzeigen.

3. Wählen Sie eine Version Ihrer Anwendung und klicken Sie auf Deploy (Bereitstellen).

Erweiterung der ECS-basierten Docker-Plattformen für Elastic Beanstalk

Elastic Beanstalk bietet Erweiterungsfunktionen, mit denen Sie Ihre eigenen Befehle, Skripts, Software und Konfigurationen auf Ihre Anwendungsbereitstellungen anwenden können. Der Bereitstellungsablauf für die ECS AL2 - und AL2023 Plattformbranchen unterscheidet sich geringfügig von den anderen Linux-basierten Plattformen. Weitere Informationen finden Sie unter Workflow zur Instance-Bereitstellung für ECS auf Amazon Linux 2 und höher.

Von ECS verwaltete Docker-Konfiguration für Elastic Beanstalk

In diesem Kapitel wird erklärt, wie Sie Ihre von ECS verwaltete Docker-Umgebung konfigurieren. In der folgenden Liste werden die in diesem Kapitel erläuterten Konfigurationselemente zusammengefasst.

- `Dockerrun.aws.jsonv2` — Diese Konfigurationsdatei spezifiziert unter anderem Ihr Image-Repository und den Namen Ihrer Docker-Images.
- EC2-Instanzprofilrolle — Wenn Sie ein benutzerdefiniertes Instanzprofil haben, erklären wir Ihnen, wie Sie es so konfigurieren, dass die für ECS zur Verwaltung Ihrer Container erforderlichen Berechtigungen aktuell bleiben.
- Elastic Load Balancing Listener — Sie müssen mehrere Elastic Load Balancing Listener konfigurieren, wenn Ihre Umgebung eingehenden Traffic für Proxys oder andere Dienste unterstützen soll, die nicht auf dem Standard-HTTP-Port laufen.

Themen

- Konfiguration der Datei DockerRun.aws.json v2
- Von Containern verwaltete Richtlinie und EC2-Instanzrolle
- Verwenden mehrerer Elastic Load Balancing-Listener

Konfiguration der Datei DockerRun.aws.json v2

`Dockerrun.aws.json v2` ist eine Elastic-Beanstalk-Konfigurationsdatei, die beschreibt, wie eine Reihe von Docker-Containern bereitgestellt wird, die in einem ECS-Cluster in einer Elastic-Beanstalk-Umgebung bereitgestellt wird. Die Elastic Beanstalk-Plattform erstellt eine

ECS-Aufgabendefinition, die eine ECS-Container-Definition beinhaltet. Diese Definitionen werden in der `Dockerrun.aws.json`-Konfigurationsdatei beschrieben.

Die Container-Definition in der `Dockerrun.aws.json`-Datei beschreibt die Container, die auf jeder Amazon-EC2-Instance im ECS-Cluster bereitgestellt werden sollen. In diesem Fall wird eine Amazon-EC2-Instance auch als Host-Container-Instance bezeichnet, da sie die Docker-Container hostet. Die Konfigurationsdatei beschreibt auch die Datenvolumen, die auf der Host-Container-Instance erstellt werden müssen, damit die Docker-Container gemountet werden können. Weitere Informationen und ein Diagramm der Komponenten in einer von ECS verwalteten Docker-Umgebung auf Elastic Beanstalk finden Sie unter Überblick über die von ECS verwaltete Docker-Plattform weiter oben in diesem Kapitel.

Eine `Dockerrun.aws.json`-Datei kann alleine verwendet oder mit einem zusätzlichen Quell-Code in einem einzigen Archiv zu einer ZIP-Datei gepackt werden. Quellcode, der mit einer `Dockerrun.aws.json`-Datei archiviert wird, wird in Amazon EC2-Container-Instances bereitgestellt und ist im `/var/app/current/`-Verzeichnis zugänglich.

Themen

- Dockerrun.aws.json v2
- Volume-Format
- ARN-Format der Ausführungsrolle
- Format der Containerdefinition
- Authentifizierungsformat – Verwenden von Images aus einem privaten Repository
- Beispiel Dockerrun.aws.json v2

`Dockerrun.aws.json` v2

Der Abschnitt `Dockerrun.aws.json` enthält die folgenden Bereiche:

AWSEBDockerrunVersion

Gibt die Versionsnummer als Wert 2 für ECS-verwaltete Docker-Umgebungen an.

executionRoleArn

Gibt die IAM-Rollen zur Aufgabenausführung für verschiedene Zwecke und Dienste an, die mit Ihrem Konto verknüpft sind. Damit Ihre Anwendung Elastic Beanstalk Beanstalk-Umgebungsvariablen verwenden kann, die als Secrets gespeichert sind, müssen Sie den ARN einer Aufgabenausführungsrolle angeben, die die erforderlichen Berechtigungen gewährt. In

anderen häufigen Anwendungsfällen ist dieser Parameter möglicherweise ebenfalls erforderlich. Weitere Informationen finden Sie unter ARN-Format der Ausführungsrolle.

volumes

Erstellt Volumes aus Ordnern in der Amazon EC2-Container-Instance oder aus Ihrem Quell-Bundle (bereitgestellt in `/var/app/current`). Mounten Sie diese Volumen auf Pfade innerhalb Ihrer Docker-Container mit `mountPoints` im `containerDefinitions`-Abschnitt.

containerDefinitions

Ein Array von Containerdefinitionen.

Authentifizierung (optional)

Der Speicherort in Amazon S3 für eine `.dockercfg`-Datei, die Authentifizierungsdaten für ein privates Repository enthält.

Die Bereiche für die Containerdefinition und Volumes von `Dockerrun.aws.json` verwenden die gleiche Formatierung wie die entsprechenden Bereiche einer Amazon-ECS-Aufgabendefinition. Weitere Informationen zum Aufgabendefinitionsformat und eine vollständige Liste der Aufgabendefinitionsparameter finden Sie in den Amazon-ECS-Aufgabendefinitionen im Entwicklerhandbuch für Amazon Elastic Container Service.

Volume-Format

Der Volume-Parameter erstellt Volumes aus Ordnern in der Amazon-EC2-Container-Instance oder aus Ihrem Quell-Bündel (bereitgestellt in `/var/app/current`).

Volumes werden in folgendem Format angegeben:

```
"volumes": [
    {
      "name": "volumename",
      "host": {
        "sourcePath": "/path/on/host/instance"
      }
    }
  ],
```

Mounten Sie diese Volumes auf Pfade innerhalb Ihrer Docker-Container mit `mountPoints` in der Containerdefinition.

Elastic Beanstalk konfiguriert zusätzliche Volumes für Protokolle, eines für jeden Container. Diese sollten von Ihren Docker-Containern bereitgestellt werden, um Protokolle an die Host-Instance zu schreiben.

Weitere Informationen finden Sie in dem `mountPoints`-Feld im folgenden Abschnitt Container-Definitionsformat.

ARN-Format der Ausführungsrolle

Damit Ihre Anwendung Elastic Beanstalk Beanstalk-Umgebungsvariablen verwenden kann, die als Secrets gespeichert sind, müssen Sie eine IAM-Rolle für die Aufgabenausführung angeben. Die Rolle muss dem Amazon ECS-Container die Erlaubnis erteilen, AWS API-Aufrufe in Ihrem Namen durchzuführen, wobei AWS Secrets Manager Secrets oder AWS Systems Manager Parameter Store-Parameter verwendet werden, um sensible Daten zu referenzieren. Anweisungen zum Erstellen einer IAM-Rolle für die Aufgabenausführung mit den erforderlichen Berechtigungen für Ihr Konto finden Sie unter Amazon ECS-IAM-Rolle zur Aufgabenausführung im Amazon Elastic Container Service Developer Guide.

```
{
"AWSEBDockerrunVersion": 2,
  "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
```

Zusätzliche Berechtigungen sind für die von Amazon ECS verwaltete Docker-Plattform erforderlich

ECS-Zuschüsse **`iam:PassRole`** für das EC2-Instanzprofil

Damit Ihr EC2-Instance-Profil dem ECS-Container diese Rolle zuweisen kann, müssen Sie die im folgenden Beispiel demonstrierte `iam:PassRole` Berechtigung angeben. Das `iam:PassRole` ermöglicht den EC2-Instances die Erlaubnis, die Aufgabenausführungsrolle an den ECS-Container zu übergeben.

In diesem Beispiel beschränken wir die EC2-Instance darauf, die Rolle nur an den ECS-Service zu übergeben. Diese Bedingung ist zwar nicht erforderlich, wir fügen sie jedoch hinzu, um den Umfang der geteilten Berechtigungen gemäß den bewährten Methoden zu reduzieren. Das erreichen wir mit dem `Condition` Element.

> **Note**
>
> Für jede Verwendung der ECS IAM-Aufgabenausführungsrolle ist die `iam:PassRole` entsprechende Genehmigung erforderlich. Es gibt andere häufige Anwendungsfälle, für die

die ECS-Rolle „Managed Service“ für die Ausführung von Aufgaben erforderlich ist. Weitere Informationen finden Sie unter IAM-Rolle für die Ausführung von Amazon ECS-Aufgaben im Amazon Elastic Container Service Developer Guide.

Example Richtlinie mit Genehmigung **`iam:PassRole`**

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/ecs-task-execution-role"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
                }
            }
        }
    ]
}
```

Zugriff auf Geheimnisse und Parameter für den Amazon ECS-Container-Agenten gewähren

Die IAM-Rolle für die Amazon ECS-Aufgabenausführung benötigt außerdem Berechtigungen für den Zugriff auf die Secrets und Parameterspeicher. Ähnlich wie bei den Anforderungen der EC2-Instance-Profilrolle benötigt der ECS-Container-Agent die Erlaubnis, die erforderlichen Secrets Manager- oder Systems Manager Manager-Ressourcen abzurufen. Weitere Informationen finden Sie unter Secrets Manager- oder Systems Manager Manager-Berechtigungen im Amazon Elastic Container Service Developer Guide

Zugriff auf Geheimnisse und Parameter auf die Elastic Beanstalk EC2-Instances gewähren

Um als Umgebungsvariablen konfigurierte Geheimnisse zu unterstützen, müssen Sie Ihrem EC2-Instance-Profil auch Berechtigungen hinzufügen. Weitere Informationen erhalten Sie unter Abrufen von Geheimnissen und Parametern für Elastic Beanstalk Beanstalk-Umgebungsvariablen und Erforderliche IAM-Berechtigungen für Secrets Manager.

Die folgenden Beispiele kombinieren das vorherige `iam:PassRole` Beispiel mit den Beispielen, auf die verwiesen Erforderliche IAM-Berechtigungen für Secrets Manager wird. Sie fügen die Berechtigungen hinzu, die die EC2-Instances für den Zugriff auf die AWS Secrets Manager und die AWS Systems Manager Speicher zum Abrufen der Secrets und Parameterdaten benötigen, um die Elastic Beanstalk Beanstalk-Umgebungsvariablen zu initialisieren, die für Secrets konfiguriert wurden.

Example Secrets Manager Manager-Richtlinie kombiniert mit **`iam:PassRole`** Berechtigungen

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/ecs-task-execution-role"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:my-secret",
                "arn:aws:kms:us-east-1:111122223333:key/my-key"
            ]
        }
    ]
```

```
}
```

Example Systems Manager Manager-Richtlinie kombiniert mit **`iam:PassRole`** Berechtigungen

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/ecs-task-execution-role"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:parameter/my-parameter",
                "arn:aws:kms:us-east-1:111122223333:key/my-key"
            ]
        }
    ]
}
```

Format der Containerdefinition

Die folgenden Beispiele zeigen eine Teilmenge von Parametern, die allgemein im Abschnitt containerDefinitions verwendet werden. Weitere optionale Parameter sind verfügbar.

Die Elastic-Beanstalk-Plattform erstellt eine ECS-Aufgabendefinition, die eine ECS-Container-Definition beinhaltet. Beanstalk unterstützt eine Teilmenge von Parametern für die ECS-Container-Definition. Weitere Informationen finden Sie unter Container-Definitionen im Entwicklerhandbuch für Amazon Elastic Container Service.

Eine `Dockerrun.aws.json`-Datei enthält ein Array aus einem oder mehreren Containerdefinitionsobjekten mit den folgenden Feldern:

Name

Name des Containers. Weitere Informationen über die maximale Länge und die zulässigen Zeichen finden Sie unter Standardparameter für Containerdefinition.

Abbild

Der Name eines Docker-Image in einem Online-Docker-Repository, aus dem Sie einen Docker-Container erstellt haben. Beachten Sie diese Konventionen:

- Abbilder in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).
- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B, `amazon/amazon-ecs-agent`.
- Abbilder in anderen Online-Repositorys sind durch einen Domänennamen zusätzlich qualifiziert (z. B, `quay.io/assemblyline/ubuntu`).

Umgebung

Ein Array an Umgebungsvariablen, die an den Container übergeben werden.

Der folgende Eintrag definiert beispielsweise eine Umgebungsvariable mit dem Namen **`Container`** und dem Wert **`PHP`**:

```
"environment": [
  {
    "name": "Container",
    "value": "PHP"
  }
],
```

essential

„true“, wenn die Aufgabe stoppen soll, wenn der Container fehlschlägt. Unwesentliche Container können abschließen oder abstürzen, ohne dass dies Auswirkungen auf die restlichen Container in der Instance hat.

memory

Größe des Speichers auf der Container-Instance, die für den Container reserviert werden soll. Geben Sie eine Ganzzahl ungleich null für einen oder beide der Parameter `memory` oder `memoryReservation` in Containerdefinitionen an.

memoryReservation

Die weiche Arbeitsspeichergrenze (in MiB) für die Reservierung für den Container. Geben Sie eine Ganzzahl ungleich null für einen oder beide der Parameter `memory` oder `memoryReservation` in Containerdefinitionen an.

mountPoints

Volumes aus der Amazon EC2-Container-Instance, die gemountet werden soll und der Speicherort im Docker-Container-Dateisystem, an dem sie gemountet werden sollen. Wenn Sie Volumes mounten, die mit Anwendungs-Inhalte enthalten, kann Ihr Container die Daten lesen, die Sie in Ihrem Quell-Bundle hochladen. Wenn Sie Protokoll-Volumes für das Schreiben von Protokolldaten mounten, kann Elastic Beanstalk Protokolldaten von diese Volumes erfassen.

Elastic Beanstalk erstellt Protokoll-Volumes auf der Container-Instance, eines für jeden Docker-Container, unter `/var/log/containers/`*`containername`*. Diese Volumes haben die Bezeichnung `awseb-logs-`*`containername`* und sollten in das Verzeichnis innerhalb der Container-Dateistruktur bereitgestellt werden, in dem Protokolle geschrieben werden.

Beispiel: Der folgende Mounting-Punkt weist den nginx-Protokoll-Standort im Container dem von Elastic Beanstalk generierten Volume für den `nginx-proxy`-Container zu.

```
{
  "sourceVolume": "awseb-logs-nginx-proxy",
  "containerPath": "/var/log/nginx"
}
```

portMappings

Ordnet Netzwerk-Ports auf dem Container Ports auf dem Host zu.

links

Liste der Container, zu denen Link erstellt werden soll. Verknüpfte Container können sich gegenseitig erkennen und sicher kommunizieren.

volumesFrom

Mounten aller Volumes aus einem anderen Container. Zum Beispiel zum Mounten von Volumes aus einem Container mit dem Namen `web`:

```
"volumesFrom": [
  {
    "sourceContainer": "web"
  }
],
```

Authentifizierungsformat – Verwenden von Images aus einem privaten Repository

Der `authentication`-Abschnitt enthält Authentifizierungsdaten für ein privates Repository. Dieser Eintrag ist optional.

Fügen Sie die Informationen über den Amazon S3-Bucket hinzu, der die Authentifizierungsdatei im `authentication`-Parameter der `Dockerrun.aws.json`-Datei enthält. Stellen Sie sicher, dass der `authentication`-Parameter einen gültigen Amazon S3-Bucket und -Schlüssel enthält. Der Amazon S3-Bucket muss in derselben Region gehostet sein wie die Umgebung, die ihn verwendet. Elastic Beanstalk lädt keine Dateien von Amazon S3-Buckets herunter, die in anderen Regionen gehostet werden.

Verwendet das folgende Format:

```
"authentication": {
    "bucket": "amzn-s3-demo-bucket",
    "key": "mydockercfg"
  },
```

Weitere Informationen zum Generieren und Hochladen der Authentifizierungsdatei finden Sie unter Authentifizierung mit Image-Repositorys.

Beispiel Dockerrun.aws.json v2

Der folgende Codeausschnitt ist ein Beispiel, das die Syntax der `Dockerrun.aws.json`-Datei für eine Instance mit zwei Containern zeigt.

```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/current/php-app"
      }
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
      "image": "php:fpm",
      "environment": [
        {
          "name": "Container",
          "value": "PHP"
        }
      ],
      "essential": true,
      "memory": 128,
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        }
      ]
    },
    {
      "name": "nginx-proxy",
      "image": "nginx",
```

```
      "essential": true,
      "memory": 128,
      "portMappings": [
        {
          "hostPort": 80,
          "containerPort": 80
        }
      ],
      "links": [
        "php-app"
      ],
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        },
        {
          "sourceVolume": "nginx-proxy-conf",
          "containerPath": "/etc/nginx/conf.d",
          "readOnly": true
        },
        {
          "sourceVolume": "awseb-logs-nginx-proxy",
          "containerPath": "/var/log/nginx"
        }
      ]
    }
  ]
}
```

Von Containern verwaltete Richtlinie und EC2-Instanzrolle

Wenn Sie eine Umgebung in der Elastic Beanstalk Beanstalk-Konsole erstellen, werden Sie aufgefordert, ein Standard-Instanzprofil zu erstellen, das die verwaltete Richtlinie enthält. `AWSElasticBeanstalkMulticontainerDocker` Daher sollte Ihr standardmäßiges EC2-Instance-Profil zunächst diese verwaltete Richtlinie enthalten. Wenn Ihre Umgebung eine benutzerdefinierte EC2-Instance-Profilrolle anstelle der Standardrolle verwendet, stellen Sie sicher, dass die verwaltete Richtlinie angehängt `AWSElasticBeanstalkMulticontainerDocker` ist, damit die erforderlichen Berechtigungen für die Containerverwaltung erhalten bleiben. up-to-date

Elastic Beanstalk verwendet ein Amazon ECS-optimiertes AMI mit einem Amazon ECS-Container-Agenten, der in einem Docker-Container ausgeführt wird. Der Agent kommuniziert mit Amazon ECS zum Koordinieren von Container-Bereitstellungen. Um mit Amazon ECS kommunizieren zu können, muss jede Amazon EC2 EC2-Instance über die entsprechenden IAM-Berechtigungen verfügen, die in dieser verwalteten Richtlinie angegeben sind. Diese Berechtigungen finden Sie AWSElasticBeanstalkMulticontainerDockerim Referenzhandbuch für AWS verwaltete Richtlinien.

Wenn Sie Elastic Beanstalk Beanstalk-Umgebungsvariablen verwenden, die für den Zugriff auf Geheimnisse oder Parameter konfiguriert sind, die in AWS Secrets Manager oder AWS Systems Manager Parameter Store gespeichert sind, müssen Sie Ihr EC2-Instance-Profil mit zusätzlichen Berechtigungen anpassen. Weitere Informationen finden Sie unter ARN-Format der Ausführungsrolle.

Verwenden mehrerer Elastic Load Balancing-Listener

Sie können mehrere Listener für Elastic Load Balancing auf einer von ECS-verwalteten Docker-Umgebung konfigurieren, um eingehenden Datenverkehr für Proxys oder andere Services zu unterstützen, die nicht auf dem Standard-HTTP-Port ausgeführt werden.

Erstellen Sie einen `.ebextensions`-Ordner in Ihrem Quell-Bundle und fügen Sie eine Datei mit einer `.config`-Dateierweiterung hinzu. Das folgende Beispiel zeigt eine Konfigurationsdatei, die einen Elastic Load Balancing-Listener auf Port 8080 erstellt.

`.ebextensions/elb-listener.config`

```
option_settings:
  aws:elb:listener:8080:
    ListenerProtocol: HTTP
    InstanceProtocol: HTTP
    InstancePort: 8080
```

Wenn Ihre Umgebung in einer von Ihnen erstellten benutzerdefinierten Amazon Virtual Private Cloud (Amazon VPC) läuft, kümmert sich Elastic Beanstalk um den Rest. In einer Standard-VPC müssen Sie die Sicherheitsgruppe Ihrer Instance so konfigurieren, dass Sie eingehenden Datenverkehr vom Load Balancer erlaubt. Fügen Sie eine zweite Konfigurationsdatei hinzu, die eine Ingress-Regel zur Sicherheitsgruppe hinzufügt:

`.ebextensions/elb-ingress.config`

```
Resources:
  port8080SecurityGroupIngress:
```

```
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 8080
      FromPort: 8080
      SourceSecurityGroupName: { "Fn::GetAtt": ["AWSEBLoadBalancer",
 "SourceSecurityGroup.GroupName"] }
```

Weitere Informationen zum Konfigurationsdateiformat finden Sie unter Hinzufügen und Anpassen von Elastic Beanstalk-Umgebungsressourcen und Optionseinstellungen.

Zusätzlich zum Hinzufügen eines Listeners zur Konfiguration von Elastic Load Balancing und dem Öffnen eines Port in der Sicherheitsgruppe müssen Sie den Port auf der Host-Instance einem Port auf dem Docker-Container im `containerDefinitions`-Abschnitt der `Dockerrun.aws.json`-v2-Datei zuordnen. Der folgende Auszug zeigt ein Beispiel:

```
"portMappings": [
  {
    "hostPort": 8080,
    "containerPort": 8080
  }
]
```

Einzelheiten `Dockerrun.aws.json v2` zum v2-Dateiformat finden Sie unter. `Dockerrun.aws.json`

Erstellen einer von ECS verwalteten Docker-Umgebung mit der Elastic Beanstalk Beanstalk-Konsole

Dieses Tutorial beschreibt die Container-Konfiguration und die Vorbereitung des Quellcodes für eine ECS-verwaltete Docker-Umgebung, die zwei Container verwendet.

Die Container, eine PHP-Anwendung und ein Nginx-Proxy, werden auf jeder der Amazon Elastic Compute Cloud (Amazon EC2) -Instances in einer Elastic Beanstalk Beanstalk-Umgebung nebeneinander ausgeführt. Nach dem Erstellen der Umgebung und der Überprüfung, ob die Anwendungen ausgeführt werden, müssen Sie eine Verbindung zu einer Container-Instance herstellen, um zu sehen, ob alles zusammenpasst.

Sections

- Definieren von ECS-verwalteten Docker-Containern
- Inhalte hinzufügen
- Bereitstellen in Elastic Beanstalk
- Verbinden mit einer Container-Instance
- Überprüfen des Amazon ECS-Container-Agenten

Definieren von ECS-verwalteten Docker-Containern

Der erste Schritt beim Erstellen einer neuen Docker-Umgebung ist das Erstellen eines Verzeichnisses für Ihre Anwendungsdaten. Dieser Ordner kann sich in einem beliebigen Verzeichnis auf Ihrem lokalen Computer befinden und einen beliebigen Namen haben. Zusätzlich zu einer Container-Konfigurationsdatei enthält dieser Ordner den Inhalt, den Sie in Elastic Beanstalk hochladen und für Ihre Umgebung bereitstellen.

> **Note**
>
> Der gesamte Code für dieses Tutorial ist im awslabs-Repository unter verfügbar. GitHub https://github.com/awslabs/eb-docker-nginx-proxy

Die Datei, die Elastic Beanstalk zur Konfiguration der Container auf einer EC2 Amazon-Instance verwendet, ist eine JSON-formatierte Textdatei mit dem Namen v2. `Dockerrun.aws.json` Die Versionen der von ECS verwalteten Docker-Plattformen verwenden ein Version 2-Format dieser Datei. Dieses Format kann nur mit der von ECS verwalteten Docker-Plattform verwendet werden, da es sich erheblich von den anderen Versionen der Konfigurationsdateien unterscheidet, die die Docker-Plattformzweige unterstützen, die nicht von ECS verwaltet werden.

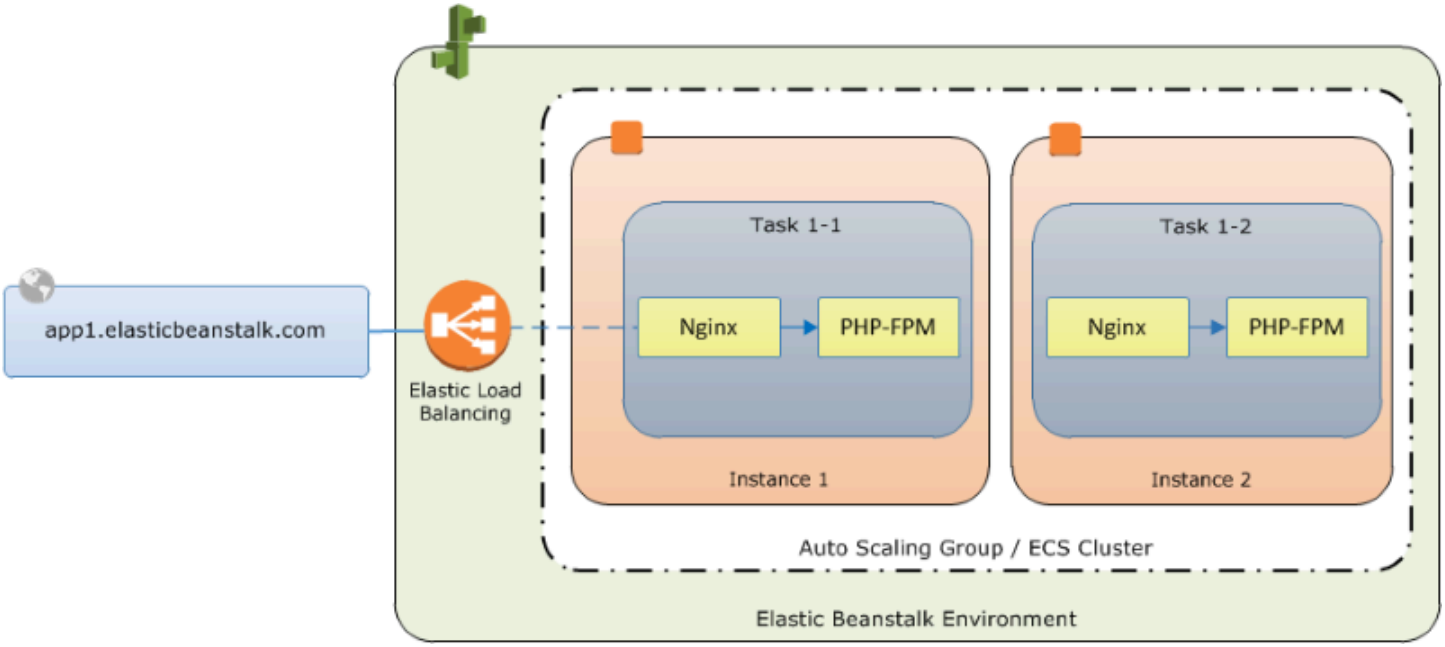
Erstellen Sie eine `Dockerrun.aws.json` v2-Textdatei mit diesem Namen im Stammverzeichnis Ihrer Anwendung und fügen Sie den folgenden Text hinzu:

```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/current/php-app"
      }
```

```
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
      "image": "php:fpm",
      "essential": true,
      "memory": 128,
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        }
      ]
    },
    {
      "name": "nginx-proxy",
      "image": "nginx",
      "essential": true,
      "memory": 128,
      "portMappings": [
        {
          "hostPort": 80,
          "containerPort": 80
        }
      ],
      "links": [
        "php-app"
      ],
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        },
        {
          "sourceVolume": "nginx-proxy-conf",
```

```
          "containerPath": "/etc/nginx/conf.d",
          "readOnly": true
        },
        {
          "sourceVolume": "awseb-logs-nginx-proxy",
          "containerPath": "/var/log/nginx"
        }
      ]
    }
  ]
}
```

In dieser Beispielkonfiguration werden zwei Container definiert, eine PHP-Website mit einem nginx-Proxy davor. Diese beiden Containern werden parallel in Docker-Containern auf jeder Instance in Ihrer Elastic Beanstalk-Umgebung ausgeführt. Dabei greifen sie auf gemeinsame Inhalte (die Inhalte der Website) aus Volumes auf der Host-Instance zu, die auch in dieser Datei definiert sind. Die Container selbst werden aus Images erstellt, die in offiziellen Repositorys auf Docker Hub gehostet werden. Die sich ergebende Umgebung sieht wie folgt aus:



Die Volumes in der Konfiguration entsprechen dem Inhalt, den Sie als Nächstes erstellen und als Teil Ihres Anwendungs-Quell-Bundle hochladen. Die Container greifen auf Inhalt auf dem Host zu, indem Volumes im `mountPoints`-Abschnitt der Container-Definitionen gemountet werden.

Weitere Informationen zum Format von `Dockerrun.aws.json` v2 und seinen Parametern finden Sie unterFormat der Containerdefinition.

Inhalte hinzufügen

Als Nächstes fügen Sie einige Inhalte für Ihre PHP-Website hinzu, um sie für Besucher anzuzeigen, und eine Konfigurationsdatei für den nginx-Proxy.

php-app/index.php

```
<h1>Hello World!!!</h1>
<h3>PHP Version <pre><?= phpversion()?></pre></h3>
```

php-app/static.html

```
<h1>Hello World!</h1>
<h3>This is a static HTML page.</h3>
```

proxy/conf.d/default.conf

```
server {
  listen 80;
  server_name localhost;
  root /var/www/html;

  index index.php;

  location ~ [^/]\.php(/|$) {
    fastcgi_split_path_info ^(.+?\.php)(/.*)$;
    if (!-f $document_root$fastcgi_script_name) {
      return 404;
    }

    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_path_info;
    fastcgi_param PATH_TRANSLATED $document_root$fastcgi_path_info;

    fastcgi_pass php-app:9000;
    fastcgi_index index.php;
  }
}
```

Bereitstellen in Elastic Beanstalk

Ihr Anwendungsordner enthält nun die folgenden Dateien:

```
### Dockerrun.aws.json
### php-app
#   ### index.php
#   ### static.html
### proxy
    ### conf.d
        ### default.conf
```

Dies ist alles, was Sie benötigen, um die Elastic Beanstalk-Umgebung zu erstellen. Erstellen Sie ein `.zip`-Archiv der obigen Dateien und Ordner (nicht im Projektordner der obersten Ebene enthalten). Zum Erstellen des Archivs in Windows Explorer wählen Sie den Inhalt des Projektordners, klicken Sie mit der rechten Maustaste, wählen Sie Send To (Senden an) und klicken Sie dann auf Compressed (zipped) Folder (Komprimierter (gezippter) Ordner).

> **Note**
>
> Weitere Informationen über die erforderliche Dateistruktur und Anweisungen für die Erstellung von Archiven in anderen Umgebungen finden Sie unter Ein Elastic Beanstalk Beanstalk-Anwendungsquellpaket erstellen

Laden Sie als Nächstes das Quell-Bundle in Elastic Beanstalk hoch und erstellen Sie Ihre Umgebung. Wählen Sie für Platform (Plattform) die Option Docker aus. Wählen Sie für Platform Branch ECS auf 64-Bit-Amazon Linux 2023 aus.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
2. Wählen Sie für Platform (Plattform) die Plattform und den Plattformzweig aus, die der von Ihrer Anwendung verwendeten Sprache entsprechen, oder die Docker-Plattform für containerbasierte Anwendungen.
3. Wählen Sie für Application code (Anwendungscode) die Option Upload your code (Eigenen Code hochladen).

4. Wählen Sie Local file (Lokale Datei), wählen Sie Choose file (Datei wählen) und öffnen Sie das Quell-Bundle.
5. Klicken Sie auf Review and launch (Überprüfen und starten).
6. Prüfen Sie die verfügbaren Einstellungen und wählen Sie Create app (Anwendung erstellen) aus.

Die Elastic Beanstalk-Konsole leitet Sie zum Verwaltungs-Dashboard für die neue Umgebung um. Dieser Bildschirm zeigt den Status der Umgebung und Ereignisse, die vom Elastic Beanstalk-Service ausgegeben werden. Wenn der Status grün ist, klicken Sie auf die URL neben dem Umgebungsnamen, um Ihre neue Website anzuzeigen.

Verbinden mit einer Container-Instance

Als Nächstes stellen Sie eine Verbindung zu einer EC2 Amazon-Instance in Ihrer Elastic Beanstalk Beanstalk-Umgebung her, um einige der beweglichen Teile in Aktion zu sehen.

Am einfachsten ist es, in Ihrer Umgebung mithilfe der EB CLI eine Verbindung mit einer Instance herzustellen. Sofern noch nicht geschehen, müssen Sie die EB CLI zuerst installieren, um sie nutzen zu können. Sie müssen Ihre Umgebung auch mit einem Amazon EC2 SSH-Schlüsselpaar konfigurieren. Verwenden Sie hierzu entweder die Sicherheitskonfigurationsseite der Konsole oder den EB CLI-Befehl eb init. Um eine Verbindung mit einer Umgebungs-Instance herzustellen, verwenden Sie den EB CLI-Befehl eb ssh.

Nachdem Sie nun mit einer EC2 Amazon-Instance verbunden sind, die Ihre Docker-Container hostet, können Sie sehen, wie die Dinge eingerichtet sind. Führen Sie `ls` auf `/var/app/current` aus:

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/app/current
Dockerrun.aws.json  php-app  proxy
```

Dieses Verzeichnis enthält die Dateien aus dem Quell-Bundle, die Sie während der Umgebungserstellung in Elastic Beanstalk hochgeladen haben.

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/log/containers
nginx-proxy    nginx-proxy-4ba868dbb7f3-stdouterr.log
php-app        php-app-dcc3b3c8522c-stdouterr.log       rotated
```

Hier werden Protokolle auf der Container-Instance erstellt und von Elastic Beanstalk erfasst. Elastic Beanstalk erstellt ein Volume in diesem Verzeichnis für jeden Container, den Sie auf den Container-Standort mounten, in dem Protokolle geschrieben werden.

Sie können auch einen Blick auf Docker werfen, um die laufenden Containern mit anzuzeige `docker ps`.

```
[ec2-user@ip-10-0-0-117 ~]$ sudo docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED
  STATUS                  PORTS                                 NAMES

4ba868dbb7f3   nginx                              "/docker-entrypoint.…"   4 minutes ago
   Up 4 minutes            0.0.0.0:80->80/tcp, :::80->80/tcp   ecs-awseb-Tutorials-env-
dc2aywfjwg-1-nginx-proxy-acca84ef87c4aca15400
dcc3b3c8522c   php:fpm                            "docker-php-entrypoi…"   4 minutes ago
   Up 4 minutes            9000/tcp                               ecs-awseb-Tutorials-env-
dc2aywfjwg-1-php-app-b8d38ae288b7b09e8101
d9367c0baad6   amazon/amazon-ecs-agent:latest   "/agent"                 5 minutes ago
  Up 5 minutes (healthy)                                  ecs-agent
```

Dies zeigt die zwei laufenden Container, die Sie bereitgestellt haben, sowie den Amazon ECS-Container-Agenten, der die Bereitstellung koordiniert hat.

Überprüfen des Amazon ECS-Container-Agenten

EC2 Amazon-Instances in einer von ECS verwalteten Docker-Umgebung auf Elastic Beanstalk führen einen Agentenprozess in einem Docker-Container aus. Dieser Agent stellt eine Verbindung mit dem Amazon ECS Service her, um die Container-Bereitstellungen zu koordinieren. Diese Bereitstellungen werden als Aufgaben in Amazon ECS ausgeführt, die in Aufgabendefinitionsdateien konfiguriert sind. Elastic Beanstalk erstellt diese Aufgabendefinitionsdateien basierend auf der `Dockerrun.aws.json`, die Sie in ein Quell-Bundle hochladen.

Überprüfen Sie den Status des Container-Agenten mit einer HTTP-Get-Anforderung an `http://localhost:51678/v1/metadata`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/metadata
{
  "Cluster":"awseb-Tutorials-env-dc2aywfjwg",
  "ContainerInstanceArn":"arn:aws:ecs:us-west-2:123456789012:container-instance/awseb-
Tutorials-env-dc2aywfjwg/db7be5215cd74658aacfcb292a6b944f",
  "Version":"Amazon ECS Agent - v1.57.1 (089b7b64)"
}
```

Diese Struktur zeigt den Namen des Amazon ECS-Clusters und den ARN (Amazon Resource Name) der Cluster-Instance (die EC2 Amazon-Instance, mit der Sie verbunden sind).

Für weitere Informationen senden Sie eine HTTP-Get-Anforderung an `http://localhost:51678/v1/tasks`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/tasks
{
   "Tasks":[
      {
         "Arn":"arn:aws:ecs:us-west-2:123456789012:task/awseb-Tutorials-env-dc2aywfjwg/
bbde7ebe1d4e4537ab1336340150a6d6",
         "DesiredStatus":"RUNNING",
         "KnownStatus":"RUNNING",
         "Family":"awseb-Tutorials-env-dc2aywfjwg",
         "Version":"1",
         "Containers":[
            {

 "DockerId":"dcc3b3c8522cb9510b7359689163814c0f1453b36b237204a3fd7a0b445d2ea6",
               "DockerName":"ecs-awseb-Tutorials-env-dc2aywfjwg-1-php-app-
b8d38ae288b7b09e8101",
               "Name":"php-app",
               "Volumes":[
                  {
                     "Source":"/var/app/current/php-app",
                     "Destination":"/var/www/html"
                  }
               ]
            },
            {

 "DockerId":"4ba868dbb7f3fb3328b8afeb2cb6cf03e3cb1cdd5b109e470f767d50b2c3e303",
               "DockerName":"ecs-awseb-Tutorials-env-dc2aywfjwg-1-nginx-proxy-
acca84ef87c4aca15400",
               "Name":"nginx-proxy",
               "Ports":[
                  {
                     "ContainerPort":80,
                     "Protocol":"tcp",
                     "HostPort":80
                  },
                  {
                     "ContainerPort":80,
                     "Protocol":"tcp",
                     "HostPort":80
```

```
                }
            ],
            "Volumes":[
                {
                    "Source":"/var/app/current/php-app",
                    "Destination":"/var/www/html"
                },
                {
                    "Source":"/var/log/containers/nginx-proxy",
                    "Destination":"/var/log/nginx"
                },
                {
                    "Source":"/var/app/current/proxy/conf.d",
                    "Destination":"/etc/nginx/conf.d"
                }
            ]
        }
    ]
  }
}
```

Diese Struktur beschreibt die Aufgabe, die zur Bereitstellung der beiden Docker-Container aus dem Beispielprojekt dieses Tutorials ausgeführt wird. Die folgenden Informationen werden angezeigt:

- KnownStatus— Der `RUNNING` Status gibt an, dass die Container noch aktiv sind.
- Family (Familie) – Der Name der Aufgabendefinition, die Elastic Beanstalk aus `Dockerrun.aws.json` erstellt hat.
- Version – Die Version der Aufgabendefinition. Diese wird mit jeder Aktualisierung der Aufgabendefinitionsdatei erhöht.
- Containers (Container) – Informationen über die Container, die in der Instance ausgeführt werden.

Noch mehr Informationen finden Sie im Amazon ECS-Service selbst, den Sie mit der aufrufe AWS Command Line Interface. Anweisungen zur Verwendung von AWS CLI mit Amazon ECS und allgemeine Informationen zu Amazon ECS finden Sie im Amazon ECS-Benutzerhandbuch.

Migrieren Sie Ihre Elastic Beanstalk Beanstalk-Anwendung von ECS-verwaltetem Multi-Container-Docker auf ECS AL1 auf Amazon Linux 2023

> **Note**
>
> Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. .

Dieses Thema führt Sie bei der Migration Ihrer Anwendungen vom ausgemusterten Plattformzweig Multi-Container Docker auf 64-Bit-Amazon Linux zu ECS, das auf 64-Bit 023 ausgeführt wird. AL2 Diese Verzweigung der Zielplattform ist aktuell und wird unterstützt. Wie der vorherige AL1Multi-Container-Docker-Zweig verwendet auch der neuere ECS AL2 023-Plattformzweig Amazon ECS, um die Bereitstellung mehrerer Docker-Container in einem Amazon ECS-Cluster in einer Elastic Beanstalk Beanstalk-Umgebung zu koordinieren. Der neue ECS AL2 023-Plattformzweig unterstützt alle Funktionen des vorherigen Multi-Container-Docker-Plattformzweigs. AL1 Außerdem wird dieselbe `Dockerrun.aws.json`-v2-Datei unterstützt.

Sections

- Migrieren mit der Elastic-Beanstalk-Konsole
- Migrieren Sie mit dem AWS CLI

Migrieren mit der Elastic-Beanstalk-Konsole

Um mithilfe der Elastic Beanstalk Beanstalk-Konsole zu migrieren, stellen Sie denselben Quellcode in einer neuen Umgebung bereit, die auf dem Plattformzweig ECS Running on AL2 023 basiert. Es sind keine Änderungen am Quellcode erforderlich.

So migrieren Sie zum Plattformzweig ECS auf Amazon Linux 2023

1. Erstellen Sie mit der Anwendungsquelle, die bereits in der alten Umgebung bereitgestellt wurde, ein Anwendungsquellpaket. Sie können dasselbe Anwendungsquellpaket und dieselbe `Dockerrun.aws.json`-v2-Datei verwenden.
2. Erstellen Sie eine neue Umgebung mit dem Plattformzweig ECS auf Amazon Linux 2023. Verwenden Sie das Quellpaket aus dem vorherigen Schritt für den Anwendungscode. Ausführlichere Schritte finden Sie weiter oben in diesem Kapitel unter Bereitstellen in Elastic Beanstalk im Tutorial zu ECS-verwaltetem Docker.

Migrieren Sie mit dem AWS CLI

Sie haben auch die Möglichkeit, die AWS Command Line Interface (AWS CLI) zu verwenden, um Ihre bestehende Multi-Container-Docker-Umgebung von Amazon Linux auf den neueren ECS AL2 023-Plattformzweig zu migrieren. In diesem Fall müssen Sie keine neue Umgebung erstellen oder Ihren Quellcode erneut bereitstellen. Sie müssen nur den Befehl update-environment ausführen. AWS CLI Er führt ein Plattform-Update durch, um die vorhandene Umgebung zum Plattformzweig ECS Amazon Linux 2023 zu migrieren.

Verwenden Sie die folgende Syntax, um Ihre Umgebung auf den neuen Plattformzweig zu migrieren.

```
aws elasticbeanstalk update-environment \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 version running ECS" \
--region my-region
```

Im Folgenden finden Sie ein Beispiel für den Befehl zum Migrieren der Umgebung beta-101 zur Version 3.0.0 des Plattformzweigs ECS Amazon Linux 2023 in der Region us-east-1.

```
aws elasticbeanstalk update-environment \
--environment-name beta-101 \
--solution-stack-name "64bit Amazon Linux 2023 v4.0.0 running ECS" \
--region us-east-1
```

Der `solution-stack-name`-Parameter stellt den Plattformzweig und seine Version bereit. Verwenden Sie die neueste Plattformzweig-Version, indem Sie den richtigen Namen des Lösungsstacks angeben. Die Version jedes Plattformzweigs ist im Namen des Lösungsstacks enthalten, wie im obigen Beispiel gezeigt. Eine Liste der aktuellsten Lösungsstacks für die Docker-Plattform finden Sie unter Unterstützte Plattformen im Leitfaden zu AWS Elastic Beanstalk - Plattformen.

> **Note**
>
> Der list-available-solution-stacksBefehl enthält eine Liste der Plattformversionen, die für Ihr Konto in einer AWS Region verfügbar sind.
>
> ```
> aws elasticbeanstalk list-available-solution-stacks --region us-east-1 --query
>  SolutionStacks
> ```

Weitere Informationen zu AWS CLI finden Sie im AWS Command Line Interface Benutzerhandbuch. Weitere Informationen zu AWS CLI Befehlen für Elastic Beanstalk finden Sie in der AWS CLI Befehlsreferenz für Elastic Beanstalk.

Authentifizierung mit Image-Repositorys

In diesem Thema wird beschrieben, wie Sie sich mit Elastic Beanstalk bei Online-Image-Repositorys authentifizieren. Bei privaten Repositorys muss Elastic Beanstalk sich authentifizieren, bevor es Ihre Images abrufen und bereitstellen kann. Für Amazon ECR Public ist die Authentifizierung optional, bietet jedoch höhere Ratenlimits und eine verbesserte Zuverlässigkeit.

Verwenden von Images aus einem Amazon ECR-Repository

Sie können Ihre benutzerdefinierten Docker-Images in AWS Amazon Elastic Container Registry (Amazon ECR) speichern.

Wenn Sie Ihre Docker-Images in Amazon ECR speichern, authentifiziert sich Elastic Beanstalk automatisch mit dem Instance-Profil Ihrer Umgebung bei der Amazon ECR-Registry. Daher müssen Sie Ihren Instances die Erlaubnis erteilen, auf die Bilder in Ihrem Amazon ECR-Repository zuzugreifen. Fügen Sie dazu dem Instance-Profil Ihrer Umgebung Berechtigungen hinzu, indem Sie die von Amazon EC2 ContainerRegistryReadOnly verwaltete Richtlinie an das Instance-Profil anhängen. Dadurch erhalten Sie schreibgeschützten Zugriff auf alle Amazon ECR-Repositorys in Ihrem Konto. Sie haben auch die Möglichkeit, nur auf ein einzelnes Repository zuzugreifen, indem Sie die folgende Vorlage verwenden, um eine benutzerdefinierte Richtlinie zu erstellen:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEbAuth",
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken"
            ],
            "Resource": [
                "*"
            ]
```

```
        },
        {
            "Sid": "AllowPull",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ecr:us-east-2:111122223333:repository/repository-name"
            ],
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:GetRepositoryPolicy",
                "ecr:DescribeRepositories",
                "ecr:ListImages",
                "ecr:BatchGetImage"
            ]
        }
    ]
}
```

Ersetzen Sie den Amazon-Ressourcennamen (ARN) in der oben genannten Policy durch den ARN Ihres Repository.

Sie müssen die Bildinformationen in Ihrer `Dockerrun.aws.json` Datei angeben. Die Konfiguration ist je nachdem, welche Plattform Sie verwenden, unterschiedlich.

Verwenden Sie für die von ECS verwaltete Docker-Plattform den `image` Schlüssel in einem Container-Definitionsobjekt:

```
"containerDefinitions": [
        {
        "name": "my-image",
        "image": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",
```

Für die Docker-Plattform verweisen Sie auf das Bild per URL. Die URL gehört zur `Image` Definition Ihrer `Dockerrun.aws.json` Datei:

```
  "Image": {
      "Name": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",
      "Update": "true"
    },
```

Benutzen AWS Secrets Manager

Konfigurieren Sie Elastic Beanstalk so, dass es sich vor der Bereitstellung bei Ihrem privaten Repository authentifiziert, um den Zugriff auf Ihre Container-Images zu ermöglichen.

Dieser Ansatz verwendet die Prebuild-Phase des Elastic Beanstalk Beanstalk-Bereitstellungsprozesses mit zwei Komponenten:

- ebextensions zur Definition von Umgebungsvariablen, die Repository-Anmeldeinformationen speichern
- Plattform-Hook-Skripte, die docker login vor dem Abrufen von Bildern ausgeführt werden

Die Hook-Skripte rufen einen Benutzernamen und ein Passwort aus Umgebungsvariablen ab, die aus einem einzigen AWS Secrets Manager Geheimnis im JSON-Format aufgefüllt werden. Für diese Funktion sind Elastic Beanstalk Docker und von ECS verwaltete Docker-Plattformen erforderlich, die am oder nach dem 13. Januar 2026 veröffentlicht wurden. Weitere Informationen finden Sie unter Environment Secrets.

So konfigurieren Sie Elastic Beanstalk für die Authentifizierung bei Ihrem privaten Repository mit AWS Secrets Manager

> **Note**
>
> Bevor Sie fortfahren, stellen Sie sicher, dass Sie Ihre Anmeldeinformationen in eingerichtet AWS Secrets Manager und die erforderlichen IAM-Berechtigungen konfiguriert haben. Einzelheiten finden Sie unter Voraussetzungen für die Konfiguration von Geheimnissen als Umgebungsvariablen.

1. Erstellen Sie die folgende Verzeichnisstruktur für Ihr Projekt:

```
### .ebextensions
#   ### env.config
### .platform
#   ### confighooks
#   #   ### prebuild
#   #       ### 01login.sh
#   ### hooks
```

```
#       ### prebuild
#           ### 01login.sh
### Dockerfile
```

2. Wird verwendet AWS Secrets Manager, um die Anmeldeinformationen Ihres privaten Repositorys als geheimen JSON-formatierten Schlüssel zu speichern.

```
aws secretsmanager create-secret --name repo-credentials \
    --secret-string '{"username":"myuser","password":"mypassword"}'
```

3. Erstellen Sie die folgende Datei `env.config` und platzieren Sie sie im Verzeichnis `.ebextensions`, wie in der obigen Verzeichnisstruktur gezeigt. Diese Konfiguration verwendet den aws:elasticbeanstalk:application:environmentsecrets Namespace mit JSON-Schlüsselextraktion, um die Umgebungsvariablen `USER` und `PASSWD` Elastic Beanstalk aus einzelnen Feldern im Secret zu initialisieren.

```
option_settings:
  aws:elasticbeanstalk:application:environmentsecrets:
    USER: arn:aws:secretsmanager:us-east-1:111122223333:secret:repo-credentials-
AbCd12:username
    PASSWD: arn:aws:secretsmanager:us-east-1:111122223333:secret:repo-credentials-
AbCd12:password
```

4. Erstellen Sie die folgende `01login.sh` Skriptdatei und platzieren Sie sie an den folgenden Speicherorten (auch in der vorherigen Verzeichnisstruktur dargestellt):

   - `.platform/confighooks/prebuild/01login.sh`
   - `.platform/hooks/prebuild/01login.sh`

```
#!/bin/bash
echo $PASSWD | docker login -u $USER --password-stdin
```

   Das `01login.sh` Skript verwendet die in Schritt 3 konfigurierten Umgebungsvariablen und übergibt das Passwort an docker login via`stdin`. Weitere Informationen zur Docker-Authentifizierung finden Sie unter docker login in der Docker-Dokumentation.

> **Hinweise**
>
> - Die von ECS verwaltete Docker-Plattform verwendet die native ECS-Syntax für die Referenzierung von Geheimnissen. Weitere Informationen finden Sie unter Weitergabe von Secrets Manager Manager-Geheimnissen durch Amazon ECS-Umgebungsvariablen im Amazon Elastic Container Service Developer Guide.
> - Weitere Informationen zu Plattform-Hooks finden Sie unter the section called "Plattform-Hooks" Extending Elastic Beanstalk Linux platforms.

Sobald die Authentifizierung konfiguriert ist, kann Elastic Beanstalk Images aus Ihrem privaten Repository abrufen und bereitstellen.

Verwenden der Datei **`Dockerrun.aws.json`**

In diesem Abschnitt wird eine andere Methode für die Authentifizierung von Elastic Beanstalk bei einem privaten Repository beschrieben. Dabei wird eine Authentifizierungsdatei mit dem Docker-Befehl erstellt und in einen Amazon S3-Bucket hochgeladen. In der Datei `Dockerrun.aws.json` müssen auch die Bucket-Informationen eingeschlossen werden.

So generieren Sie eine Authentifizierungsdatei und stellen sie in Elastic Beanstalk bereit

1. Erstellen Sie eine Authentifizierungsdatei mit dem docker login-Befehl. Für Repositorys auf Docker Hub führen Sie au docker login:

   ```
   $ docker login
   ```

   Für andere Registrys fügen Sie die URL des Registry-Servers ein:

   ```
   $ docker login registry-server-url
   ```

   > **Note**
   >
   > Wenn Ihre Elastic Beanstalk-Umgebung die Amazon Linux AMI-Docker-Plattformversion verwendet (vor Amazon Linux 2), lesen Sie die entsprechenden Informationen unter

the section called “Docker-Konfiguration auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)”.

Weitere Informationen zur Authentifizierungsdatei finden Sie unter Store images on Docker Hub und docker login auf der Docker-Website.

2. Laden Sie eine Kopie der Authentifizierungsdatei mit dem Namen `.dockercfg` in einen sicheren Amazon S3-Bucket hoch.

   - Der Amazon S3 S3-Bucket muss in derselben AWS-Region Umgebung gehostet werden, in der er verwendet wird. Elastic Beanstalk kann keine Dateien von einem Amazon S3-Bucket herunterladen, der in anderen Regionen gehostet wird.
   - Erteilen Sie Berechtigungen für den Vorgang `s3:GetObject` für die IAM-Rolle im Instance-Profil. Weitere Informationen finden Sie unter Elastic Beanstalk Instance-Profile verwalten.

3. Schließen Sie die Amazon S3-Bucket-Informationen in den `Authentication`-Parameter der Datei `Dockerrun.aws.json` ein.

   Das folgende Beispiel zeigt die Verwendung einer Authentifizierungsdatei mit dem Namen `mydockercfg` in einem Bucket namens `amzn-s3-demo-bucket`, um ein privates Image in einem Drittanbieter-Registry zu verwenden. Die richtige Versionsnummer für `AWSEBDockerrunVersion` finden Sie in der Anmerkung, die dem Beispiel folgt.

```
{
  "AWSEBDockerrunVersion": "version-no",
  "Authentication": {
    "Bucket": "amzn-s3-demo-bucket",
    "Key": "mydockercfg"
  },
  "Image": {
    "Name": "quay.io/johndoe/private-image",
    "Update": "true"
  },
  "Ports": [
    {
      "ContainerPort": "1234"
    }
  ],
  "Volumes": [
    {
```

```
      "HostDirectory": "/var/app/mydb",
      "ContainerDirectory": "/etc/mysql"
    }
  ],
  "Logging": "/var/log/nginx"
}
```

`Dockerrun.aws.json`-Versionen

Der `AWSEBDockerrunVersion`-Parameter gibt die Version der `Dockerrun.aws.json`-Datei an.

- Der Docker AL2 und die AL2023 Plattformen verwenden die folgenden Versionen der Datei.
  - `Dockerrun.aws.json v3`— Umgebungen, die Docker Compose verwenden.
  - `Dockerrun.aws.json v1`— Umgebungen, die Docker Compose nicht verwenden.
- ECS, das auf Amazon Linux 2 ausgeführt wird, und ECS, das auf läuft, AL2023 verwendet die `Dockerrun.aws.json v2` Datei. Die ausgemusterte Plattform ECS-The Multicontainer Docker Amazon Linux AMI (AL1) verwendete ebenfalls dieselbe Version.

Wenn sich Elastic Beanstalk bei der Onlineregistrierung authentifizieren kann, die als Host für das private Repository fungiert, können Ihre Images bereitgestellt und abgerufen werden.

Verwenden von Bildern aus Amazon ECR Public

Amazon ECR Public ist eine öffentliche Container-Registry, die Docker-Images hostet. Öffentliche Amazon ECR Repositorys sind zwar öffentlich zugänglich, aber die Authentifizierung bietet höhere Ratenlimits und eine höhere Zuverlässigkeit für Ihre Bereitstellungen.

Note

Die öffentliche Amazon ECR-Authentifizierung wird in den Regionen (`cn-*`) und AWS GovCloud Regionen (`us-gov-*`) China nicht unterstützt. In diesen Regionen verwendet Elastic Beanstalk nicht authentifizierte Pulls.

Um die öffentliche Amazon ECR-Authentifizierung zu aktivieren, fügen Sie dem Instance-Profil Ihrer Umgebung die folgenden Berechtigungen hinzu. Weitere Informationen zur Amazon ECR Public Authentication finden Sie unter Registry authentication in Amazon ECR public im Amazon Elastic Container Registry Public User Guide:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
       {
          "Effect": "Allow",
          "Action": [
             "ecr-public:GetAuthorizationToken",
             "sts:GetServiceBearerToken"
          ],
          "Resource": "*"
       }
    ]
}
```

Sobald diese Berechtigungen mit Ihrem Instance-Profil verknüpft sind, authentifiziert sich Elastic Beanstalk automatisch bei den öffentlichen Registern von Amazon ECR. Sie können Amazon ECR Public Images im `public.ecr.aws/`*`registry-alias`*`/`*`repository-name:tag`* Standardformat in Ihrer `Dockerrun.aws.json` Datei oder Dockerfile referenzieren.

Konfiguration von Elastic Beanstalk Docker-Umgebungen

In diesem Kapitel werden zusätzliche Konfigurationsinformationen für alle unterstützten Docker-Plattformzweige erläutert, einschließlich des von ECS verwalteten Docker-Plattformzweigs. Sofern in einem Abschnitt kein bestimmter Plattformzweig oder eine Komponente für Plattformzweige angegeben ist, gilt dies für alle Umgebungen, in denen unterstützte Docker- und ECS-verwaltete Docker-Plattformen ausgeführt werden.

> **Note**
>
> Wenn Ihre Elastic Beanstalk-Umgebung eine Amazon Linux AMI Docker-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie unbedingt die zusätzlichen

Informationen unter the section called “Docker-Konfiguration auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)”.

Abschnitte

- Konfigurieren von Software in Docker-Umgebungen
- Verweisen auf Umgebungsvariablen in Containern
- Verwenden der Interpolationsfunktion für Umgebungsvariablen mit Docker Compose
- Generieren von Protokollen für verbesserte Gesundheitsberichte mit Docker Compose
- Benutzerdefinierte Protokollierung für Docker-Container mit Docker Compose
- Docker-Images
- Konfigurieren von verwalteten Aktualisierungen für Docker-Umgebungen
- Docker-Konfigurations-Namespaces
- Docker-Konfiguration auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)

Konfigurieren von Software in Docker-Umgebungen

Sie können die Elastic Beanstalk-Konsole zum Konfigurieren der Software verwenden, die in den Instances der Umgebung ausgeführt wird.

So konfigurieren Sie Ihre Docker-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Nehmen Sie die notwendigen Konfigurationsänderungen vor.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Hinweise zum Konfigurieren von Software-Einstellungen in einer beliebigen Umgebung finden Sie unter the section called “Umgebungsvariablen und Softwareeinstellungen”. Die folgenden Abschnitte behandeln Docker-spezifische Informationen.

Containeroptionen

Der Abschnitt Container options (Containeroptionen) enthält plattformspezifische Optionen. Für Docker-Umgebungen können Sie auswählen, ob Ihre Umgebung den NGINX-Proxy-Server enthält.

Umgebungen mit Docker Compose

Wenn Sie Ihre Docker-Umgebung mit Docker Compose verwalten, geht Elastic Beanstalk davon aus, dass Sie einen Proxy-Server als Container ausführen. Daher wird die Einstellung Proxy-Server standardmäßig auf Keine festgelegt und von Elastic Beanstalk wird keine NGINX-Konfiguration bereitgestellt.

> **Note**
>
> Selbst wenn Sie NGINX als Proxy-Server auswählen, wird diese Einstellung in einer Umgebung mit Docker Compose ignoriert. Die Einstellung Proxy-Server wird trotzdem standardmäßig auf Keine festgelegt.

Da der NGINX-Webserver-Proxy für die Amazon Linux 2-basierte Docker-Plattform mit Docker Compose deaktiviert ist, müssen Sie die Schritte zum Generieren von Protokollen für erweiterte Zustandsberichte ausführen. Weitere Informationen finden Sie unter Generieren von Protokollen für verbesserte Gesundheitsberichte mit Docker Compose.

Umgebungseigenschaften (Umgebungsvariablen)

Sie können Umgebungseigenschaften (auch als Umgebungsvariablen bezeichnet) verwenden, um Werte wie Endpunkte, Debug-Einstellungen und andere Informationen an Ihre Anwendung zu übergeben. Im Bereich Umgebungsvariablen der Konsole können Sie Umgebungsvariablen für die EC2-Instances angeben, auf denen Ihre Anwendung ausgeführt wird. Umgebungsvariablen werden als Schlüssel-Wert-Paare an die Anwendung übergeben.

Ihr Anwendungscode, der in einem Container ausgeführt wird, kann nach Namen auf eine Umgebungsvariable verweisen und dessen Wert lesen. Der Quellcode, der diese Umgebungsvariablen liest, variiert je nach Programmiersprache. Anweisungen zum Lesen von

Werten der Umgebungsvariablen in den Programmiersprachen, die von Elastic Beanstalk verwaltete Plattformen unterstützen, finden Sie im jeweiligen Plattformthema. Eine Liste der Links zu diesen Themen finden Sie unter the section called “Umgebungsvariablen und Softwareeinstellungen”.

Geheimnisse und Parameter in Elastic Beanstalk Beanstalk-Umgebungsvariablen

Elastic Beanstalk bietet die Möglichkeit, Daten in Umgebungsvariablen zu referenzieren AWS Secrets Manager und AWS Systems Manager Parameter zu speichern. Dies ist eine sichere Option für Ihre Anwendung, um nativ auf Geheimnisse und Parameter zuzugreifen, die von diesen Diensten gespeichert werden, ohne API-Aufrufe an sie verwalten zu müssen. Ihre Elastic Beanstalk Docker- und ECS-verwalteten Docker-Plattformen müssen eine Version sein, die am oder nach dem 26. März 2025 veröffentlicht wurde, um diese Funktion zu unterstützen. Weitere Hinweise zur Verwendung von Umgebungsvariablen zur Referenzierung von Geheimnissen finden Sie unter. Abrufen von Geheimnissen und Parametern für Elastic Beanstalk Beanstalk-Umgebungsvariablen

Umgebungen mit Docker Compose

Wenn Sie Ihre Docker-Umgebung mit Docker Compose verwalten, sind einige zusätzliche Konfigurationsschritte erforderlich, um die Umgebungsvariablen in den Containern abzurufen. Damit die ausführbaren Dateien, die in Ihrem Container ausgeführt werden, auf diese Umgebungsvariablen zugreifen können, muss in der `docker-compose.yml` auf sie verwiesen werde. Weitere Informationen finden Sie unter Verweisen auf Umgebungsvariablen in Containern.

Verweisen auf Umgebungsvariablen in Containern

Wenn Sie das Docker Compose-Tool auf der Amazon Linux 2-Docker-Plattform verwenden, wird von Elastic Beanstalk im Stammverzeichnis Ihres Anwendungsprojekts eine Docker Compose-Umgebungsdatei mit der Bezeichnung `.env` generiert. In dieser Datei werden die für Elastic Beanstalk konfigurierten Umgebungsvariablen gespeichert.

> **Note**
>
> Wenn Sie eine `.env`-Datei in Ihr Anwendungspaket aufnehmen, wird von Elastic Beanstalk keine `.env`-Datei generiert.

Damit ein Container auf die in Elastic Beanstalk definierten Umgebungsvariablen verweisen kann, muss mindestens einer der folgenden Konfigurationsansätze verwendet werden.

- Fügen Sie die von Elastic Beanstalk generierte `.env`-Datei zur Konfigurationsoption `env_file` in der `docker-compose.yml`-Datei hinzu.
- Definieren Sie die Umgebungsvariablen direkt in der `docker-compose.yml`-Datei.

Im Anschluss finden Sie entsprechende Beispieldateien. Die `docker-compose.yml`-Beispieldatei veranschaulicht beide Ansätze.

- Wenn Sie die Umgebungseigenschaften `DEBUG_LEVEL=1` und `LOG_LEVEL=error` definieren, erstellt Elastic Beanstalk die folgende `.env`-Datei für Sie:

```
DEBUG_LEVEL=1
LOG_LEVEL=error
```

- In dieser `docker-compose.yml`-Datei verweist die Konfigurationsoption `env_file` auf die `.env`-Datei. Außerdem wird die Umgebungsvariable `DEBUG=1` direkt in der `docker-compose.yml`-Datei definiert.

```
services:
  web:
    build: .
    environment:
      - DEBUG=1
    env_file:
      - .env
```

Hinweise

- Wenn Sie die gleiche Umgebungsvariable in beiden Dateien festlegen, hat die in der `docker-compose.yml`-Datei definierte Variable Vorrang vor der Variablen in der `.env`-Datei.
- Achten Sie darauf, dass sich zwischen dem Gleichheitszeichen (=) und dem zugewiesenen Wert Ihrer Variablen keine Leerzeichen befinden. Andernfalls werden der Zeichenfolge Leerzeichen hinzugefügt.

Weitere Informationen zu Umgebungsvariablen in Docker Compose finden Sie unter Environment variables in Compose.

Verwenden der Interpolationsfunktion für Umgebungsvariablen mit Docker Compose

Ab der Plattformversion vom 28. Juli 2023 bietet der Plattformzweig von Docker Amazon Linux 2 das Docker-Compose-Interpolations-Feature. Mit diesem Feature können Werte in einer Compose-Datei durch Variablen festgelegt und zur Laufzeit interpoliert werden. Weitere Informationen über dieses Feature finden Sie unter Interpolation auf der Docker-Dokumentationswebsite.

> **Important**
>
> Wenn Sie dieses Feature mit Ihren Anwendungen verwenden möchten, beachten Sie, dass Sie einen Ansatz implementieren müssen, der Plattform-Hooks verwendet.
> Dies ist aufgrund einer Maßnahme erforderlich, die wir in der Plattform-Engine implementiert haben. Diese Maßnahme stellt die Abwärtskompatibilität für Kunden sicher, die das neue Interpolations-Feature nicht kennen und bestehende Anwendungen haben, die Umgebungsvariablen mit dem Zeichen $ verwenden. Die aktualisierte Plattform-Engine umgeht die Interpolation standardmäßig, indem sie das $ Zeichen durch $$ Zeichen ersetzt.

Im Folgenden finden Sie ein Beispiel für ein Plattform-Hook-Skript, das Sie einrichten können, um die Verwendung des Interpolations-Features zu ermöglichen.

```
#!/bin/bash

: '
example data format in .env file
key1=value1
key2=value2
'
envfile="/var/app/staging/.env"
tempfile=$(mktemp)

while IFS= read -r line; do
  # split each env var string at '='
  split_str=(${line//=/ })
  if [ ${#split_str[@]} -eq 2 ]; then
    # replace '$$' with '$'
    replaced_str=${split_str[1]//\$\$/\$}
    # update the value of env var using ${replaced_str}
    line="${split_str[0]}=${replaced_str}"
```

```
  fi
  # append the updated env var to the tempfile
  echo "${line}" #"${tempfile}"
done < "${envfile}"
# replace the original .env file with the tempfile
mv "${tempfile}" "${envfile}"
```

Platzieren Sie die Plattform-Hooks unter diesen beiden Verzeichnissen:

- `.platform/confighooks/predeploy/`
- `.platform/hooks/predeploy/`

Weitere Informationen finden Sie unter Plattform-Hooks im Thema Erweiterung von Linux-Plattformen in diesem Handbuch.

Generieren von Protokollen für verbesserte Gesundheitsberichte mit Docker Compose

Der Elastic-Beanstalk-Zustandsagent stellt Betriebssystem- und Anwendungszustandsmetriken für Elastic-Beanstalk-Umgebungen bereit. Es nutzt Webserver-Protokollformate, die Informationen in einem bestimmten Format weiterleiten.

Elastic Beanstalk nimmt an, dass Sie einen Webserver-Proxy als Container ausführen. Daher ist der NGINX-Webserver-Proxy für Docker-Umgebungen deaktiviert, in denen Docker Compose ausgeführt wird. Sie müssen Ihren Server so konfigurieren, dass Protokolle an den vom Elastic Beanstalk-Zustandsagenten verwendeten Speicherort und im entsprechenden Format geschrieben werden. Dies ermöglicht die uneingeschränkte Nutzung der erweiterten Zustandsberichte, auch wenn der Webserver-Proxy deaktiviert ist.

Anweisungen dazu finden Sie unter Konfiguration von Webserver-Protokollen

Benutzerdefinierte Protokollierung für Docker-Container mit Docker Compose

Zur effizienten Behandlung von Problemen sowie zur effizienten Überwachung Ihrer containerisierten Services können Sie von Elastic Beanstalk über die Umgebungsverwaltungskonsole oder über die EB CLI Instance-Protokolle anfordern. Instance-Protokolle umfassen kombinierte und verpackte Bundle-Protokolle und Protokollfragmente, um eine effiziente und unkomplizierte Betrachtung von Protokollen und aktuellen Ereignissen zu ermöglichen.

Für jeden in der `docker-compose.yml`-Datei definierten Service wird von Elastic Beanstalk unter `/var/log/eb-docker/containers/`*`<service name>`* in der Container-Instance ein Protokollverzeichnis erstellt. Wenn Sie das Docker Compose-Feature auf der Amazon Linux 2-Docker-Plattform verwenden, können Sie diese Verzeichnisse an dem Speicherort innerhalb der Containerdateistruktur mounten, an dem Protokolle geschrieben werden. Wenn Sie Protokollverzeichnisse zum Schreiben von Protokolldaten mounten, kann Elastic Beanstalk Protokolldaten aus diesen Verzeichnissen erfassen.

Um die Protokolldateien Ihres Dienstes so zu konfigurieren, dass sie abrufbare Tail-Dateien und Bundle-Logs sind

1. Bearbeiten Sie die `docker-compose.yml`-Datei.
2. Fügen Sie unter dem Schlüssel `volumes` für Ihren Service ein Bind-Mount hinzu:

   `"${EB_LOG_BASE_DIR}/`*`<service name>`*`:`*`<log directory inside container>`*

   In der folgenden `docker-compose.yml`-Beispieldatei gilt Folgendes:

   - `nginx-proxy` ist *`<service name>`*
   - `/var/log/nginx` ist *`<log directory inside container>`*

```
services:
  nginx-proxy:
    image: "nginx"
    volumes:
      - "${EB_LOG_BASE_DIR}/nginx-proxy:/var/log/nginx"
```

- Das Verzeichnis `var/log/nginx` enthält die Protokolle für den Dienst nginx-proxy im Container und wird dem Verzeichnis `/var/log/eb-docker/containers/nginx-proxy` auf dem Host zugeordnet.
- Alle Protokolle in diesem Verzeichnis sind nun als Bundle und als Protokollfragmente über die Elastic Beanstalk-Funktion zum Anfordern von Instance-Protokollen abrufbar.

Hinweise

- ${EB_LOG_BASE_DIR} ist eine Umgebungsvariable, die von Elastic Beanstalk mit dem Wert `/var/log/eb-docker/containers` versehen wird
- Elastic Beanstalk erstellt das `/var/log/eb-docker/containers/<service name>`-Verzeichnis für jeden Service in der `docker-compose.yml`-Datei.

Docker-Images

Die von Docker und ECS verwalteten Docker-Plattformzweige für Elastic Beanstalk unterstützen die Verwendung von Docker-Images, die in einem öffentlichen oder privaten Online-Image-Repository gespeichert sind.

Geben Sie Images mit Namen in `Dockerrun.aws.json` an. Beachten Sie diese Konventionen:

- Abbilder in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).
- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B, `amazon/amazon-ecs-agent`).
- Images in anderen Online-Repositorys sind durch einen Domainnamen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu` oder *`account-id`*`.dkr.ecr.us-east-2.amazonaws.com/ubuntu:trusty`).

Für Umgebungen, die nur die Docker-Plattform verwenden, können Sie auch während der Erstellung der Umgebung mit einer Docker-Datei ein eigenes Image erstellen. Details dazu finden Sie unter Erstellen benutzerdefinierter Images mit einer Dockerfile-Datei. Die von ECS verwaltete Docker-Plattform unterstützt diese Funktionalität nicht.

Konfigurieren von verwalteten Aktualisierungen für Docker-Umgebungen

Mit verwalteten Plattformaktualisierungen können Sie die Umgebung so konfigurieren, dass automatisch nach einem Zeitplan eine Aktualisierung auf die neueste Version einer Plattform durchgeführt wird.

Im Fall von Docker-Umgebungen können Sie entscheiden, ob eine automatische Plattformaktualisierung innerhalb von Docker-Versionen stattfinden soll – falls die neue Version

der Plattform eine neue Docker-Version enthält. Elastic Beanstalk unterstützt verwaltete Plattform-Updates für alle Docker-Versionen, wenn von einer Umgebung aus aktualisiert wird, auf der eine Docker-Plattformversion neuer als 2.9.0 läuft. Wenn eine neue Plattformversion eine neue Version von Docker enthält, inkrementiert Elastic Beanstalk die Nebenversionsnummer der Aktualisierung. Um verwaltete Plattformaktualisierungen für Docker-Versionen zuzulassen, aktivieren Sie deshalb verwaltete Plattformaktualisierungen für Nebenversions- und Patch-Versionsaktualisierungen. Um verwaltete Plattformaktualisierungen innerhalb von Docker-Versionen zu verhindern, aktivieren Sie verwaltete Plattformaktualisierungen nur für Patch-Versionsaktualisierungen.

Die folgende Konfigurationsdatei beispielsweise ermöglicht verwaltete Plattformaktualisierungen jeden Dienstag um 9:00 Uhr UTC für Neben- und Patch-Versionsaktualisierungen und lässt damit verwaltete Aktualisierungen innerhalb von Docker-Versionen zu:

Example.ebeextensions/ .config managed-platform-update

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: minor
```

Für Umgebungen mit Docker-Plattformversionen 2.9.0 oder früher führt Elastic Beanstalk nie verwaltete Plattformaktualisierungen durch, wenn die neue Plattformversion eine neue Docker-Version enthält.

Docker-Konfigurations-Namespaces

Mithilfe einer Konfigurationsdatei können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Die Konfigurationsoptionen können plattformspezifisch sein oder für alle Plattformen im Elastic Beanstalk-Service als Ganzes gelten. Die Konfigurationsoptionen sind in Namespaces organisiert.

> **Note**
>
> Diese Informationen gelten nur für Docker-Umgebungen, in denen Docker Compose nicht ausgeführt wird. In Docker-Umgebungen, in denen Docker Compose ausgeführt wird, weist diese Option ein anderes Verhalten auf. Weitere Informationen zu Proxy-Services mit Docker Compose finden Sie unter Containeroptionen.

Die Docker-Plattform unterstützt neben den unterstützten Optionen für alle Elastic Beanstalk-Umgebungen auch Optionen in den folgenden Namespaces:

- `aws:elasticbeanstalk:environment:proxy` – Wählen Sie den Proxy-Server für Ihre Umgebung aus. Docker unterstützt entweder die Ausführung von Nginx oder keinen Proxy-Server.

In der folgenden Beispielkonfigurationsdatei wird eine Docker-Umgebung so konfiguriert, dass kein Proxy-Server ausgeführt wird.

Example.ebextensions/docker-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: none
```

Docker-Konfiguration auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)

Wenn Ihre Elastic Beanstalk Docker-Umgebung eine Amazon Linux AMI-Plattformversion (Vorgängerversion von Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Verwenden einer Authentifizierungsdatei für ein privates Repository

Diese Informationen sind für Sie relevant, wenn Sie Bilder aus einem privaten Repository verwenden. Beginnend mit der Docker-Version 1.7 hat der Befehl docker login den Namen der Authentifizierungsdatei und das Format der Datei geändert. Amazon Linux AMI Docker-Plattformversionen (Vorgängerversion von Amazon Linux 2) erfordern die Konfigurationsdatei für ein älteres `~/.dockercfg` Format.

Ab Docker-Version 1.7 und höher erstellt der docker login-Befehl die Authentifizierungsdatei in `~/.docker/config.json` im folgenden Format:

```
{
    "auths":{
      "server":{
        "auth":"key"
      }
    }
  }
```

Ab Docker-Version 1.6.2 und früher erstellt der docker login-Befehl die Authentifizierungsdatei in `~/.dockercfg` im folgenden Format:

```
{
    "server" :
    {
      "auth" : "auth_token",
      "email" : "email"
    }
  }
```

Wenn Sie eine `config.json`-Datei konvertieren möchten, entfernen Sie den äußeren `auths`-Schlüssel, fügen Sie einen `email`-Schlüssel hinzu und reduzieren Sie das JSON-Dokument, um dem alten Format zu entsprechen.

Bei Amazon Linux 2 Docker-Plattformversionen verwendet Elastic Beanstalk den Namen und das Format der neueren Authentifizierungsdatei. Wenn Sie eine Amazon Linux 2-Docker-Plattformversion verwenden, können Sie die vom docker login-Befehl erstellte Authentifizierungsdatei ohne Konvertierung nutzen.

Konfigurieren von zusätzlichen Speicher-Volumes

Zur Verbesserung der Amazon Linux AMI-Leistung konfiguriert Elastic Beanstalk zwei Amazon EBS-Speicher-Volumes für die Amazon EC2-Instances Ihrer Docker-Umgebung. Zusätzlich zum Stamm-Volume, das für alle Elastic Beanstalk-Umgebungen bereitgestellt wird, wird ein zweites 12-GB-Volume mit dem Namen `xvdcz` für die Image-Speicherung auf Docker-Umgebungen verfügbar gemacht.

Wenn Sie zusätzlichen Speicherplatz oder mehr IOPS für Docker-Images benötigen, können Sie das Image-Speicher-Volume mit der `BlockDeviceMapping`-Konfigurationsoption im aws:autoscaling:launchconfiguration-Namespace anpassen.

Die folgende Konfigurationsdatei erhöht die Größe des Speicher-Volumes beispielsweise auf 100 GB mit 500 bereitgestellten IOPS:

Example.ebextensions/blockdevice-xvdcz.config

```
option_settings:
  aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: /dev/xvdcz=:100::io1:500
```

Wenn Sie die `BlockDeviceMappings`-Option zur Konfiguration zusätzlicher Volumes für Ihre Anwendung verwenden, sollten Sie ein Mapping für `xvdcz` einschließen, um sicherzustellen, dass es erstellt wurde. Im folgenden Beispiel werden zwei Volumes konfiguriert, das Image-Speicher-Volume `xvdcz` mit Standardeinstellungen und ein zusätzliches 24-GB-Anwendungs-Volume mit dem Namen `sdh`:

Example.ebextensions/blockdevice-sdh.config

```
option_settings:
  aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: /dev/xvdcz=:12:true:gp2,/dev/sdh=:24
```

> **Note**
>
> Wenn Sie die Einstellungen in diesem Namespace ändern, ersetzt Elastic Beanstalk alle Instances in Ihrer Umgebung durch Instances, die mit der neuen Konfiguration ausgeführt werden. Details dazu finden Sie unter Konfigurationsänderungen.

Ältere Plattformen

In diesem Kapitel werden Inhalte aufgeführt, die sich auf frühere Docker-Plattformen beziehen, die von nicht mehr unterstützt werden. AWS Elastic Beanstalk Die hier aufgeführten Themen werden weiterhin in diesem Dokument als Referenz für alle Kunden verwendet, die diese Funktionen oder Komponenten vor ihrer Einstellung verwendet haben.

Themen

- Migration von Multi-Container-Docker auf Amazon Linux zu Elastic Beanstalk Docker auf Amazon Linux 2
- Vorkonfigurierte GlassFish Docker-Container auf Elastic Beanstalk

Migration von Multi-Container-Docker auf Amazon Linux zu Elastic Beanstalk Docker auf Amazon Linux 2

Vor der Veröffentlichung des Plattformzweigs ECS auf Amazon Linux 2 mit 64 Bit bot Elastic Beanstalk einen alternativen Migrationspfad zu Amazon Linux 2 für Kunden mit Umgebungen an, die auf dem Plattformzweig Multi-Container-Docker auf Amazon Linux mit 64 Bit basierten. Dieses

Thema beschreibt diesen Migrationspfad und bleibt in diesem Dokument als Referenz für alle Kunden, die diesen Migrationspfad abgeschlossen haben.

Wir empfehlen jetzt Kunden mit Umgebungen, die auf dem Plattformzweig Multi-Container-Docker auf Amazon Linux mit 64 Bit basieren, zum Plattformzweig ECS auf Amazon Linux 2 mit 64 Bit zu migrieren. Im Gegensatz zum alternativen Migrationspfad verwendet dieser Ansatz weiterhin Amazon ECS, um Container-Bereitstellungen in ECS-verwalteten Docker-Umgebungen zu koordinieren. Dieser Aspekt erlaubt einen einfacheren Ansatz. Es sind keine Änderungen am Quellcode erforderlich und dasselbe `Dockerrun.aws.json` v2 wird unterstützt. Weitere Informationen finden Sie unter Migrieren Sie Ihre Elastic Beanstalk Beanstalk-Anwendung von ECS-verwaltetem Multi-Container-Docker auf ECS AL1 auf Amazon Linux 2023.

Legacy-Migration von Multi-Container-Docker auf Amazon Linux zum Docker-Plattformzweig Amazon Linux 2

Sie können Ihre Anwendungen, die auf der Multicontainer-Docker-Plattform unter Amazon Linux AMI ausgeführt werden, zur Amazon Linux 2 Docker-Plattform migrieren. Bei der Multicontainer-Docker-Plattform unter Amazon Linux AMI müssen vordefinierte Anwendungsimages für die Ausführung als Container angegeben werden. Nach der Migration besteht diese Einschränkung nicht mehr, da die Amazon Linux 2 Docker-Plattform die Erstellung Ihrer Container-Images mit Elastic Beanstalk auch während der Bereitstellung ermöglicht. Ihre Anwendungen werden weiterhin in Umgebungen mit mehreren Containern ausgeführt – mit den zusätzlichen Vorteilen des Docker Compose-Tools.

Docker Compose ist ein Tool zum Definieren und Ausführen von Multi-Container-Docker-Anwendungen. Weitere Informationen zu Docker Compose und zur Installation von Docker Compose finden Sie auf den Docker-Websites Overview of Docker Compose und Install Docker Compose.

Die Datei **`docker-compose.yml`**

Das Docker Compose-Tool verwendet die `docker-compose.yml`-Datei für die Konfiguration Ihrer Anwendungsservices. Diese Datei ersetzt die `Dockerrun.aws.json` v2-Datei in Ihrem Anwendungsprojektverzeichnis und Anwendungsquellpaket. Die `docker-compose.yml`-Datei muss zwar manuell erstellt werden, die meisten der Parameterwerte können Sie jedoch Ihrer `Dockerrun.aws.json` v2-Datei entnehmen.

Im Anschluss finden Sie ein Beispiel für eine `docker-compose.yml`-Datei und die entsprechende `Dockerrun.aws.json` v2-Datei für die gleiche Anwendung. Weitere Informationen zur `docker-compose.yml`-Datei finden Sie in der Referenz zu Compose-Dateien. Weitere Informationen zur `Dockerrun.aws.json` v2-Datei finden Sie unter `Dockerrun.aws.json` v2.

docker-compose.yml

```
version: '2.4'
services:
  php-app:
    image: "php:fpm"
    volumes:
      - "./php-app:/var/www/html:ro
"
      - "${EB_LOG_BASE_DIR}/php-app
:/var/log/sample-app"
    mem_limit: 128m
    environment:
      Container: PHP
  nginx-proxy:
    image: "nginx"
    ports:
      - "80:80"
    volumes:
      - "./php-app:/var/www/html:ro
"
      - "./proxy/conf.d:/etc/nginx/
conf.d:ro"
      - "${EB_LOG_BASE_DIR}/nginx-p
roxy:/var/log/nginx"
    mem_limit: 128m
    links:
      - php-app
```

Dockerrun.aws.json v2

```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/
current/php-app"
      }
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/
current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
      "image": "php:fpm",
      "environment": [
        {
          "name": "Container",
          "value": "PHP"
        }
      ],
      "essential": true,
      "memory": 128,
      "mountPoints": [
        {
          "sourceVolume": "php-app"
,
          "containerPath": "/var/www
/html",
          "readOnly": true
        }
      ]
    },
```

docker-compose.yml	Dockerrun.aws.json v2

```
    {
      "name": "nginx-proxy",
      "image": "nginx",
      "essential": true,
      "memory": 128,
      "portMappings": [
        {
          "hostPort": 80,
          "containerPort": 80
        }
      ],
      "links": [
        "php-app"
      ],
      "mountPoints": [
        {
          "sourceVolume": "php-app"
,
          "containerPath": "/var/www
/html",
          "readOnly": true
        },
        {
          "sourceVolume": "nginx-pr
oxy-conf",
          "containerPath": "/etc/ngi
nx/conf.d",
          "readOnly": true
        },
        {
          "sourceVolume": "awseb-lo
gs-nginx-proxy",
          "containerPath": "/var/log
/nginx"
        }
      ]
    }
  ]
}
```

Weitere Überlegungen zur Migration

Die Docker-Amazon-Linux-2-Plattform und die Multi-Container-Docker-Amazon-Linux-AMI-Plattform implementieren Umgebungseigenschaften unterschiedlich. Diese beiden Plattformen haben auch unterschiedliche Protokollverzeichnisse, die Elastic Beanstalk für jeden ihrer Container erstellt. Nachdem Sie von der Amazon-Linux-AMI-Multi-Container-Docker-Plattform migriert haben, müssen Sie sich dieser verschiedenen Implementierungen für Ihre neue Amazon-Linux-2-Docker-Plattformumgebung bewusst sein.

Area	Docker-Plattform auf Amazon Linux 2 mit Docker Compose	Multi-Container-Docker-Plattform unter Amazon Linux AMI
Umgebungseigenschaften	Damit Ihre Container auf Umgebungseigenschaften zugreifen können, müssen Sie einen Verweis auf die `.env`-Datei in der `docker-compose.yml` -Datei hinzufügen. Elastic Beanstalk generiert die `.env`-Datei und listet jede der Eigenschaften als Umgebungsvariable auf. Weitere Informationen finden Sie unter Verweisen auf Umgebungsvariablen in Containern.	Elastic Beanstalk kann Umgebungseigenschaften direkt an den Container übergeben. Der im Container ausgeführte Code kann auf diese Eigenschaften ohne zusätzliche Konfiguration in Form von Umgebungsvariablen zugreifen.
Protokollverzeichnisse	Für jeden Container erstellt Elastic Beanstalk ein Protokollverzeichnis namens `/var/log/eb-docker/containers/` *`<service name>`* (oder `${EB_LOG_BASE_DIR}/<service name>`). Weitere Informationen finden Sie unter Benutzerdefinierte Protokollierung für Docker-Container mit Docker Compose.	Für jeden Container erstellt Elastic Beanstalk ein Protokollverzeichnis namens `/var/log/containers/` *`<containername>`* . Weitere Informationen finden Sie im Feld `mountPoints` unter Format der Containerdefinition.

Migrationsschritte

So migrieren Sie zur Amazon Linux 2 Docker-Plattform

1. Erstellen Sie auf der Grundlage der vorhandenen `docker-compose.yml` -Datei die `Dockerrun.aws.json` v2-Datei für Ihre Anwendung. Weitere Informationen finden Sie weiter oben im Abschnitt Die Datei `docker-compose.yml`.
2. Ersetzen Sie im Stammverzeichnis Ihres Anwendungsprojektordners die `Dockerrun.aws.json` v2-Datei durch die soeben erstellte `docker-compose.yml`-Datei.

   Ihre Verzeichnisstruktur sollte wie folgt aussehen.

   ```
   ~/myApplication
   |-- docker-compose.yml
   |-- .ebextensions
   |-- php-app
   |-- proxy
   ```

3. Verwenden Sie den Befehl eb init, um Ihr lokales Verzeichnis für die Bereitstellung in Elastic Beanstalk zu konfigurieren.

   ```
   ~/myApplication$ eb init -p docker application-name
   ```

4. Verwenden Sie den Befehl eb create, um eine Umgebung zu erstellen und Ihr Docker-Image bereitzustellen.

   ```
   ~/myApplication$ eb create environment-name
   ```

5. Wenn es sich bei Ihrer App um eine Webanwendung handelt, verwenden Sie nach dem Start der Umgebung den Befehl eb open, um sie in einem Webbrowser anzuzeigen.

   ```
   ~/myApplication$ eb open environment-name
   ```

6. Den Status Ihrer neu erstellten Umgebung können Sie mithilfe des Befehls eb status anzeigen.

   ```
   ~/myApplication$ eb status environment-name
   ```

Vorkonfigurierte GlassFish Docker-Container auf Elastic Beanstalk

> **Note**
>
> Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

Der vorkonfigurierte GlassFish Docker-Plattformzweig, der auf dem Amazon Linux AMI (AL1) läuft, wird nicht mehr unterstützt. Um Ihre GlassFish Anwendung auf eine unterstützte Amazon Linux 2023-Plattform zu migrieren, stellen GlassFish Sie Ihren Anwendungscode auf einem Amazon Linux 2023 Docker-Image bereit. Weitere Informationen finden Sie im Thema the section called “Tutorial — GlassFish auf Docker: Pfad zu Amazon Linux 2023”.

Erste Schritte mit vorkonfigurierten Docker-Containern – auf Amazon Linux AMI (Vorgänger von Amazon Linux 2)

Dieser Abschnitt zeigt die lokale Entwicklung einer Beispielanwendung und die anschließende Bereitstellung in Elastic Beanstalk mit einem vorkonfigurierten Docker-Container.

Einrichten der lokalen Entwicklungsumgebung

Für diese Anleitung verwenden wir eine GlassFish Beispielanwendung.

Einrichten Ihrer Umgebung

1. Erstellen Sie einen neuen Ordner für die Beispielanwendung.

```
~$ mkdir eb-preconf-example
~$ cd eb-preconf-example
```

2. Laden Sie den Beispiel-Anwendungscode in den neuen Ordner herunter.

```
~$ wget https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-glassfish-v1.zip
~$ unzip docker-glassfish-v1.zip
~$ rm docker-glassfish-v1.zip
```

Lokales Entwickeln und Testen

Um eine Beispielanwendung zu entwickeln GlassFish

1. Fügen Sie eine `Dockerfile`-Datei zum Stammordner der Anwendung hinzu. Geben Sie in der Datei das AWS Elastic Beanstalk Docker-Basis-Image an, das zum Ausführen Ihres lokalen vorkonfigurierten Docker-Containers verwendet werden soll. Später werden Sie Ihre Anwendung auf einer vorkonfigurierten GlassFish Docker-Version von Elastic Beanstalk bereitstellen. Wählen Sie das Docker-Basis-Image, das diese Plattformversion verwendet. Informationen zum Bestimmen des aktuellen Docker-Image der Plattformversion finden Sie im Abschnitt Vorkonfigurierter Docker der Seite Von AWS Elastic Beanstalk unterstützte Plattformen im AWS Elastic Beanstalk -Plattformhandbuch.

   Example b-preconf-example~/E /Dockerfile

   ```
   # For Glassfish 5.0 Java 8
   FROM amazon/aws-eb-glassfish:5.0-al-onbuild-2.11.1
   ```

   Weitere Informationen zur Verwendung von `Dockerfile` finden Sie unter Vorbereiten Ihres Docker-Images für die Bereitstellung auf Elastic Beanstalk.

2. Erstellen Sie das Docker-Image.

   ```
   ~/eb-preconf-example$ docker build -t my-app-image .
   ```

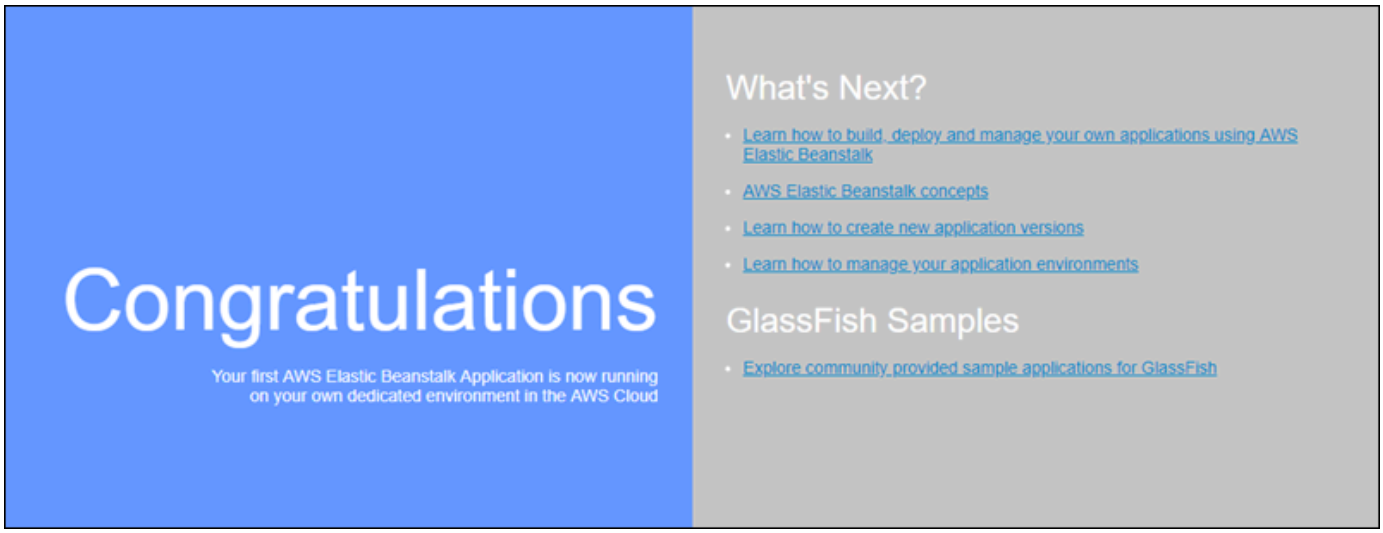
3. Führen Sie den Docker-Container aus dem Image aus.

   > **Note**
   >
   > Sie müssen das Flag `-p` einbinden, um Port 8080 auf dem Container dem Localhost-Port 3000 zuzuordnen. In Elastic Beanstalk stellen die Docker-Container die Anwendung stets über Port 8080 auf dem Container bereit. Mit den Flags `-it` wird das Image als interaktiver Prozess ausgeführt. Mithilfe des Flags `--rm` wird das Container-Dateisystem nach Beendigung des Containers bereinigt. Optional können Sie das Flag `-d` einbeziehen und das Image als Daemon ausführen.

   ```
   $ docker run -it --rm -p 3000:8080 my-app-image
   ```

4. Geben Sie die folgende URL in den Webbrowser ein, um die Beispielanwendung anzuzeigen.

```
http://localhost:3000
```



Bereitstellen in Elastic Beanstalk

Nachdem Sie die Anwendung getestet haben, können Sie diese nun in Elastic Beanstalk bereitstellen.

So stellen Sie Ihre Anwendung in Elastic Beanstalk bereit

1. Ändern Sie im Stammordner der Anwendung den Dateinamen von `Dockerfile` in `Dockerfile.local`. Dieser Schritt ist erforderlich, damit Elastic Beanstalk die `Dockerfile`-Datei mit den richtigen Anweisungen für Elastic Beanstalk verwendet, um ein angepasstes Docker-Image auf den einzelnen Amazon EC2-Instances in Ihrer Elastic Beanstalk-Umgebung zu erstellen.

> **Note**
>
> Dieser Schritt ist nicht nötig, wenn die `Dockerfile`-Datei Anweisungen enthält, mit denen das Docker-Basis-Image der Plattformversion geändert wird. Sie brauchen überhaupt keine `Dockerfile`-Datei, sofern die `Dockerfile`-Datei nur eine `FROM`-Zeile enthält und darin das Basis-Image angibt, mit dem der Container erstellt werden soll. In diesem Fall ist die `Dockerfile`-Datei überflüssig.

2. Erstellen Sie das Quell-Bundle der Anwendung.

```
~/eb-preconf-example$ zip myapp.zip -r *
```

3. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? Anwendungsname=Tutorials&EnvironmentType= LoadBalanced
4. Wählen Sie für Platform (Plattform) unter Preconfigured – Docker (Vorkonfiguriert – Docker) die Option Glassfish aus.
5. Unter Application code wählen Sie Upload your code und dann Upload aus.
6. Wählen Sie Local file (Lokale Datei), wählen Sie Browse (Durchsuchen) und öffnen Sie dann das Quell-Bundle, das Sie erstellt haben.
7. Klicken Sie auf Upload.
8. Klicken Sie auf Review and launch (Überprüfen und starten).
9. Prüfen Sie die verfügbaren Einstellungen und wählen Sie Create app (Anwendung erstellen) aus.
10. Wenn die Umgebung erstellt wurde, können Sie die bereitgestellte Anwendung anzeigen. Wählen Sie die Umgebungs-URL, die oben im Konsolendashboard angezeigt wird.

Bereitstellung einer GlassFish Anwendung auf der Docker-Plattform: ein Migrationspfad zu Amazon Linux 2023

Ziel dieses Tutorials ist es, Kunden, die die vorkonfigurierte GlassFish Docker-Plattform (basierend auf Amazon Linux AMI) verwenden, einen Migrationspfad zu Amazon Linux 2023 zu bieten.
Sie können Ihre GlassFish Anwendung auf Amazon Linux 2023 migrieren, indem Sie Ihren Anwendungscode auf einem Amazon Linux 2023 Docker-Image bereitstellen GlassFish .

Das Tutorial führt Sie durch die Verwendung der AWS Elastic Beanstalk Docker-Plattform, um eine auf dem Java GlassFish EE-Anwendungsserver basierende Anwendung in einer Elastic Beanstalk Beanstalk-Umgebung bereitzustellen.

Wir demonstrieren zwei Ansätze zum Erstellen eines Docker-Images:

- Einfach — Stellen Sie den Quellcode Ihrer GlassFish Anwendung bereit und lassen Sie Elastic Beanstalk im Rahmen der Bereitstellung Ihrer Umgebung ein Docker-Image erstellen und ausführen. Dies ist einfach einzurichten, zu Kosten einer erhöhten Bereitstellungszeit der Instance.

- Fortgeschritten – Erstellen Sie ein benutzerdefiniertes Docker-Image, das Ihren Anwendungscode und Abhängigkeiten enthält, und stellen Sie es Elastic Beanstalk für die Verwendung in Ihrer Umgebung bereit. Dieser Ansatz ist etwas komplexer und verkürzt die Bereitstellungszeit von Instances in Ihrer Umgebung.

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse zu den grundlegenden Elastic-Beanstalk-Operationen, den Elastic-Beanstalk-Befehlszeilenschnittstellen (EB CLI) und Docker verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können, um die erste Elastic-Beanstalk-Umgebung zu starten. Dieses Tutorial verwendet die EB CLI, aber Sie können auch mit der Elastic-Beanstalk-Konsole Umgebungen erstellen und Anwendungen hochladen.

Für diese Anleitung benötigen Sie außerdem die folgenden Docker-Komponenten:

- Eine funktionierende lokale Installation von Docker. Weitere Informationen finden Sie unter Docker erhalten auf der Docker-Docs-Website.
- Zugriff auf Docker-Hub. Sie müssen eine Docker-ID erstellen, um auf den Docker-Hub zuzugreifen. Weitere Informationen finden Sie unter Freigeben der Anwendung auf der Docker-Dokumentationswebsite.

Weitere Informationen über das Konfigurieren von Docker-Umgebungen auf Elastic-Beanstalk-Plattformen finden Sie unter Vorbereiten Ihres Docker-Images für die Bereitstellung auf Elastic Beanstalk in diesem Kapitel.

Einfaches Beispiel: Geben Sie Ihren Anwendungscode ein

Dies ist eine einfache Möglichkeit, Ihre Anwendung bereitzustellen. GlassFish Sie geben den Quellcode Ihrer Anwendung zusammen mit der in diesem Tutorial enthaltenen `Dockerfile` an. Elastic Beanstalk erstellt ein Docker-Image, das Ihre Anwendung und den GlassFish Software-Stack enthält. Dann führt Elastic Beanstalk das Image auf den Instances Ihrer Umgebung aus.

Ein Problem mit diesem Ansatz besteht darin, dass Elastic Beanstalk das Docker-Image lokal erstellt, wenn es eine Instance für Ihre Umgebung erstellt. Das Erstellen des Image erhöht die Zeit für die Bereitstellung von Instances. Diese Auswirkung beschränkt sich nicht auf die anfängliche Erstellung der Umgebung – auch bei Scale-Out-Aktionen kann dies stattfinden.

Um eine Umgebung mit einer Beispielanwendung zu starten GlassFish

1. Laden Sie das Beispiel `docker-glassfish-al2-v1.zip` herunter und entpacken Sie die `.zip`-Datei dann in ein Verzeichnis in Ihrer Entwicklungsumgebung.

   ```
   ~$ curl https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-glassfish-al2-v1.zip --output docker-glassfish-al2-v1.zip
   ~$ mkdir glassfish-example
   ~$ cd glassfish-example
   ~/glassfish-example$ unzip ../docker-glassfish-al2-v1.zip
   ```

   Ihre Verzeichnisstruktur sollte wie folgt aussehen.

   ```
   ~/glassfish-example
   |-- Dockerfile
   |-- Dockerrun.aws.json
   |-- glassfish-start.sh
   |-- index.jsp
   |-- META-INF
   |   |-- LICENSE.txt
   |   |-- MANIFEST.MF
   |   `-- NOTICE.txt
   |-- robots.txt
   `-- WEB-INF
       `-- web.xml
   ```

   Die folgenden Dateien sind der Schlüssel zum Erstellen und Ausführen eines Docker-Containers in Ihrer Umgebung:

   - `Dockerfile` – Enthält Anweisungen, die Docker verwendet, um ein Image mit Ihrer Anwendung und erforderlichen Abhängigkeiten zu erstellen.
   - `glassfish-start.sh` – Ein Shell-Skript, das das Docker-Image ausführt, um Ihre Anwendung zu starten.
   - `Dockerrun.aws.json`— Stellt einen Logging-Schlüssel bereit, um das GlassFish Anwendungsserver-Protokoll in Logdateianfragen aufzunehmen. Wenn Sie nicht an GlassFish Protokollen interessiert sind, können Sie diese Datei weglassen.

2. Konfigurieren Sie Ihr lokales Verzeichnis für die Bereitstellung in Elastic Beanstalk.

   ```
   ~/glassfish-example$ eb init -p docker glassfish-example
   ```

3. (Optional) Verwenden Sie den Befehl eb local run für die lokale Erstellung und Ausführung Ihres Containers.

```
~/glassfish-example$ eb local run --port 8080
```

> **Note**
>
> Weitere Informationen zum eb local-Befehl finden Sie unter the section called “eb local”. Der Befehl wird unter Windows nicht unterstützt. Alternativ können Sie Ihre Container mit den Befehlen docker build und docker run erstellen und ausführen. Weitere Informationen finden Sie in der Docker-Dokumentation.

4. (Optional) Wenn Ihr Container ausgeführt wird, verwenden Sie den eb local open-Befehl, um Ihre Anwendung in einem Webbrowser anzuzeigen. Alternativ öffnen Sie http://localhost:8080/ in einem Webbrowser.

```
~/glassfish-example$ eb local open
```

5. Verwenden Sie den Befehl eb create, um eine Umgebung zu erstellen und die Anwendung bereitzustellen.

```
~/glassfish-example$ eb create glassfish-example-env
```

6. Verwenden Sie nach dem Starten der Umgebung den Befehl eb open, um sie in einem Webbrowser anzuzeigen.

```
~/glassfish-example$ eb open
```

Wenn Sie mit dem Beispiel fertig sind, beenden Sie die Umgebung und löschen Sie zugehörige Ressourcen.

```
~/glassfish-example$ eb terminate --all
```

Fortgeschrittenes Beispiel: Bereitstellen eines vorgefertigten Docker-Images

Dies ist eine fortgeschrittenere Methode zur Bereitstellung Ihrer GlassFish Anwendung. Aufbauend auf dem ersten Beispiel erstellen Sie ein Docker-Image, das Ihren Anwendungscode und den GlassFish Software-Stack enthält, und übertragen es auf Docker Hub. Nachdem Sie diesen

einmaligen Schritt ausgeführt haben, können Sie Elastic Beanstalk-Umgebungen basierend auf Ihrem benutzerdefinierten Image starten.

Wenn Sie eine Umgebung starten und Ihr Docker-Image bereitstellen, laden Instances in Ihrer Umgebung dieses Image direkt herunter, verwenden es direkt und müssen kein Docker-Image erstellen. So wird die Bereitstellungszeit der Instance verkürzt.

> **Hinweise**
>
> - In den folgenden Schritten erstellen Sie ein öffentlich verfügbares Docker-Image.
> - Sie verwenden Docker-Befehle aus Ihrer lokalen Docker-Installation zusammen mit Ihren Docker-Hub-Anmeldeinformationen. Weitere Informationen finden Sie im vorherigen Abschnitt mit den Voraussetzungen in diesem Thema.

Um eine Umgebung mit einem vordefinierten GlassFish Anwendungs-Docker-Image zu starten

1. Laden Sie das Beispiel `docker-glassfish-al2-v1.zip` herunter und entpacken Sie es wie im vorherigen einfachen Beispiel. Wenn Sie dieses Beispiel bereits abgeschlossen haben, können Sie das bereits vorhandene Verzeichnis verwenden.
2. Erstellen Sie ein Docker-Image und schieben Sie es an Docker Hub. Geben Sie Ihre Docker-ID ein*`docker-id`*, um sich bei Docker Hub anzumelden.

   ```
   ~/glassfish-example$ docker build -t docker-id/beanstalk-glassfish-example:latest .
   ~/glassfish-example$ docker push docker-id/beanstalk-glassfish-example:latest
   ```

   > **Note**
   >
   > Bevor Sie Ihr Image übertragen, müssen Sie möglicherweise ausführe docker login. Wenn Sie den Befehl ohne Parameter ausführen, werden Sie aufgefordert, Ihre Docker-Hub-Anmeldeinformationen einzugeben.

3. Erstellen Sie ein zusätzliches Verzeichnis.

   ```
   ~$ mkdir glassfish-prebuilt
   ~$ cd glassfish-prebuilt
   ```

4. Kopieren Sie das folgende Beispiel in eine Datei mit dem Namen `Dockerrun.aws.json`.

Example~/glassfish-prebuilt/Dockerrun.aws.json

```
{
  "AWSEBDockerrunVersion": "1",
  "Image": {
    "Name": "docker-username/beanstalk-glassfish-example"
  },
  "Ports": [
    {
      "ContainerPort": 8080,
      "HostPort": 8080
    }
  ],
  "Logging": "/usr/local/glassfish5/glassfish/domains/domain1/logs"
}
```

5. Konfigurieren Sie Ihr lokales Verzeichnis für die Bereitstellung in Elastic Beanstalk.

```
~/glassfish-prebuilt$ eb init -p docker glassfish-prebuilt$
```

6. (Optional) Verwenden Sie den Befehl eb local run, um den Container lokal auszuführen.

```
~/glassfish-prebuilt$ eb local run --port 8080
```

7. (Optional) Wenn Ihr Container ausgeführt wird, verwenden Sie den eb local open-Befehl, um Ihre Anwendung in einem Webbrowser anzuzeigen. Alternativ öffnen Sie http://localhost:8080/ in einem Webbrowser.

```
~/glassfish-prebuilt$ eb local open
```

8. Verwenden Sie den Befehl eb create, um eine Umgebung zu erstellen und Ihr Docker-Image bereitzustellen.

```
~/glassfish-prebuilt$ eb create glassfish-prebuilt-env
```

9. Verwenden Sie nach dem Starten der Umgebung den Befehl eb open, um sie in einem Webbrowser anzuzeigen.

```
~/glassfish-prebuilt$ eb open
```

Wenn Sie mit dem Beispiel fertig sind, beenden Sie die Umgebung und löschen Sie zugehörige Ressourcen.

```
~/glassfish-prebuilt$ eb terminate --all
```

Überwachungsumgebungen in Elastic Beanstalk

Mit Elastic Beanstalk Health Monitoring können Sie die Verfügbarkeit von Anwendungen überprüfen und Warnmeldungen erstellen, die aktiviert werden, wenn die Messwerte Ihre Schwellenwerte überschreiten. Sie können Elastic Beanstalk Health Monitoring sowohl in der Konsole als auch in der Befehlszeile verwenden, um den Status Ihrer Umgebung zu verfolgen.

Themen

- Überwachung des Zustands der Umgebung in der AWS Managementkonsole
- Verwenden der EB-CLI zur Überwachung des Umgebungszustands
- Grundlegende Zustandsberichte
- Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk
- Alarme verwalten
- Anzeigen des Änderungsverlaufs einer Elastic Beanstalk-Umgebung
- Ereignis-Stream einer Elastic Beanstalk-Umgebung anzeigen
- Auflisten von Server-Instances/Verbinden mit Server-Instances
- Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen
- Bereitstellungsprotokolle für eine Elastic Beanstalk Beanstalk-Umgebung anzeigen

Überwachung des Zustands der Umgebung in der AWS Managementkonsole

Sie können über die Elastic Beanstalk-Konsole auf operative Informationen zu Ihrer Anwendung zugreifen. Umgebungsstatus und Anwendungszustand werden in der Konsole übersichtlich dargestellt. Auf der Seite Environments (Umgebungen) der Konsole und auf der Seite jeder Anwendung werden die Umgebungen in der Liste farbcodiert, um den Status anzuzeigen.

So überwachen Sie eine Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

3. Wählen Sie im Navigationsbereich Monitoring (Überwachung) aus.

Auf der Seite "Monitoring" werden allgemeine Statistikwerte zur Umgebung angegeben, z. B. CPU-Auslastung und durchschnittliche Latenz. Neben diesen allgemeinen Statistikdaten können Sie Überwachungsdiagramme mit der Ressourcennutzung im Verlauf der Zeit aufrufen. Klicken Sie auf ein Diagramm, um detailliertere Informationen zu sehen.

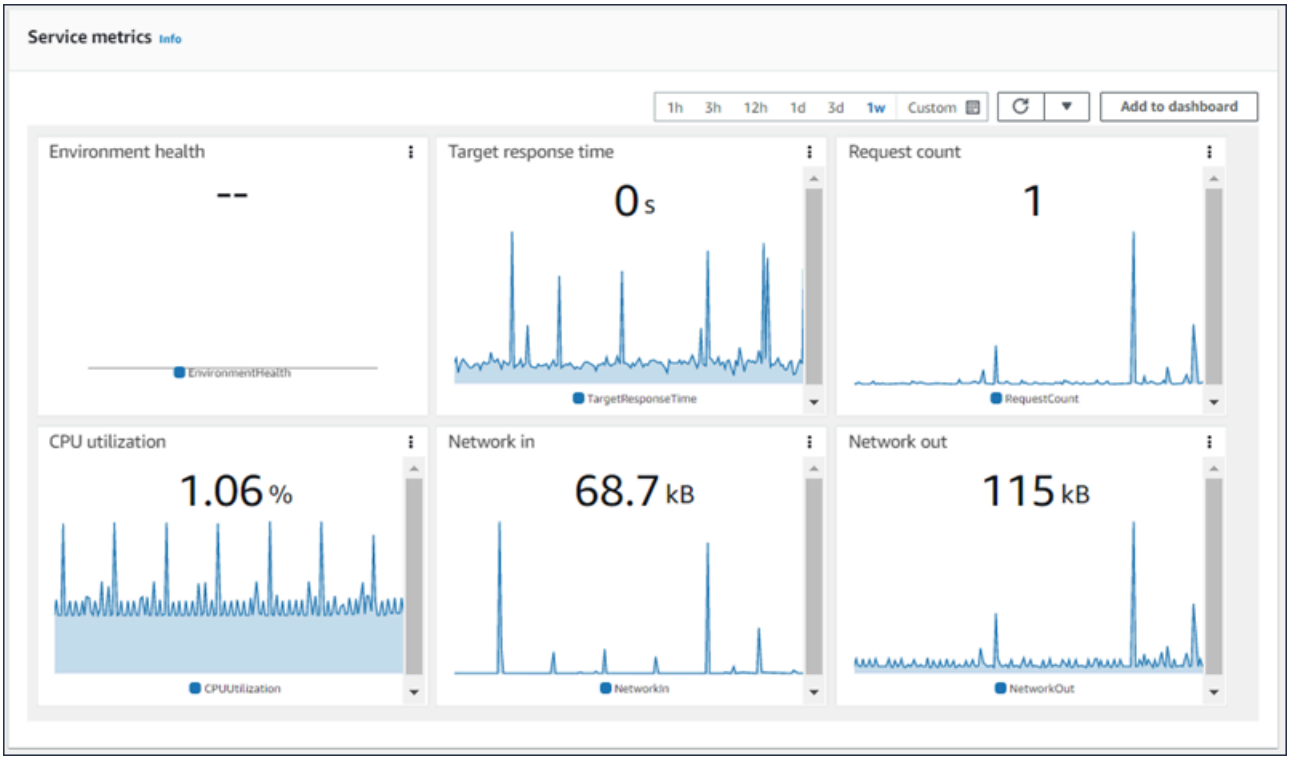
> **Note**
>
> Standardmäßig sind nur CloudWatch Basismetriken aktiviert, die Daten innerhalb von fünf Minuten zurückgeben. Sie können detailliertere CloudWatch Ein-Minuten-Metriken aktivieren, indem Sie die Konfigurationseinstellungen Ihrer Umgebung bearbeiten.

Überwachungsdiagramme

Die Seite Überwachung zeigt eine Übersicht über die Zustandsdaten für Ihre Umgebung an. Dazu gehören die von Elastic Load Balancing und Amazon bereitgestellten Standardmetriken sowie Grafiken EC2, die zeigen, wie sich der Zustand der Umgebung im Laufe der Zeit verändert hat.

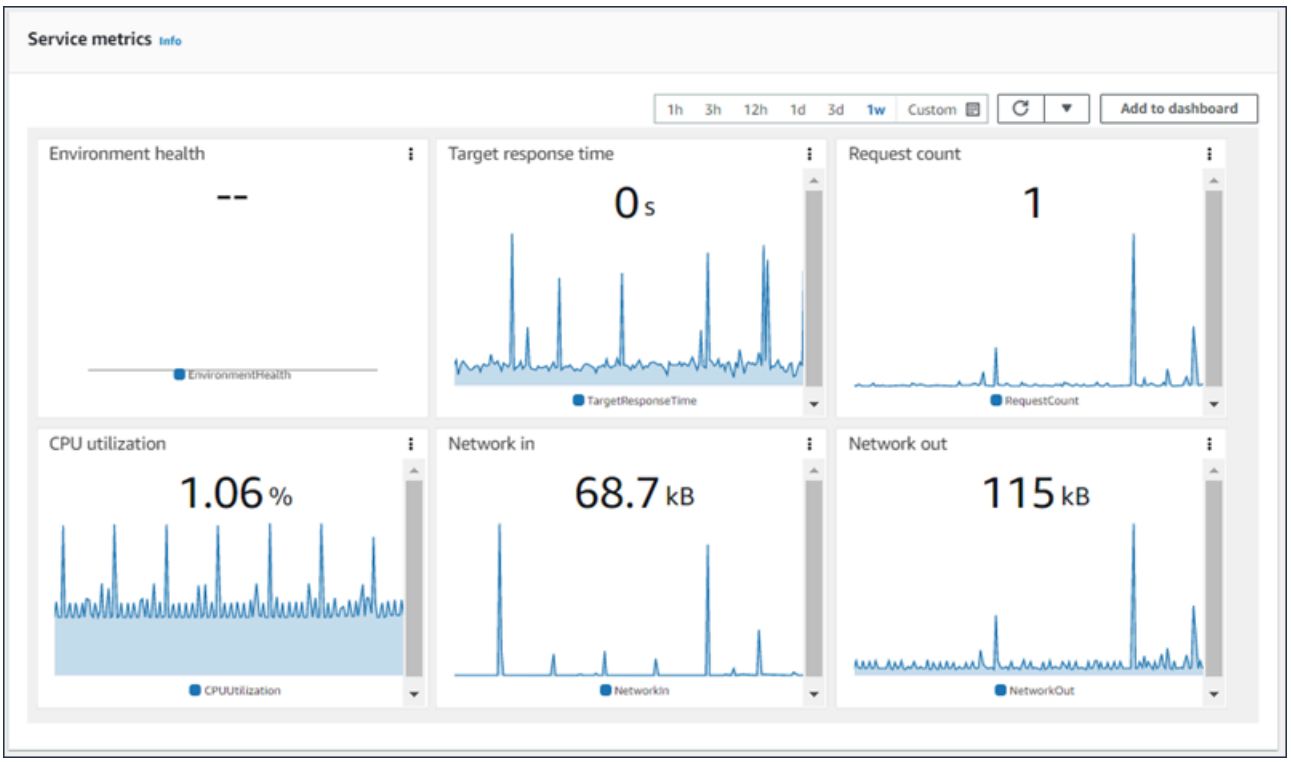
Die Leiste über den Grafiken bietet eine Vielzahl von Zeitintervallen, aus denen Sie auswählen können. Wählen Sie beispielsweise 1w aus, um Informationen anzuzeigen, die sich über die letzte Woche erstrecken. Oder wählen Sie 3h, um Informationen aus den letzten drei Stunden anzuzeigen.

Wählen Sie Benutzerdefiniert, um eine größere Auswahl an Zeitintervallen zu erhalten. Von hier aus haben Sie zwei Bereichsoptionen: Absolut oder Relativ. Mit der Option Absolut können Sie einen bestimmten Datumsbereich angeben, z. B. den 1. Januar 2023 bis 30. Juni 2023. Die Option Relativ ermöglicht die Auswahl einer Ganzzahl mit einer bestimmten Zeiteinheit: Minuten, Stunden, Tage, Wochen oder Monate. Beispiele hierfür sind 10 Stunden, 10 Tage und 10 Monate.



Anpassen der Überwachungskonsole

Um benutzerdefinierte Metriken zu erstellen und anzuzeigen, müssen Sie Amazon verwenden CloudWatch. Mit können CloudWatch Sie benutzerdefinierte Dashboards erstellen, um Ihre Ressourcen in einer einzigen Ansicht zu überwachen. Wählen Sie Zum Dashboard hinzufügen, um von der Monitoring-Seite zur CloudWatch Amazon-Konsole zu gelangen. Amazon CloudWatch bietet Ihnen die Möglichkeit, ein neues Dashboard zu erstellen oder ein vorhandenes auszuwählen. Weitere Informationen finden Sie unter Verwenden von CloudWatch Amazon-Dashboards im CloudWatch Amazon-Benutzerhandbuch.



Elastic Load Balancing und EC2Amazon-Metriken sind für alle Umgebungen aktiviert.

Bei verbesserter Integrität ist die EnvironmentHealth Metrik aktiviert und der Überwachungskonsole wird automatisch ein Diagramm hinzugefügt. Zudem wird mit erweiterten Zustandsberichten auch eine Zustandsprüfungsseite zur Management Console hinzugefügt. Eine Liste der verfügbaren erweiterten Zustandsmetriken finden Sie unter Veröffentlichen CloudWatch benutzerdefinierter Amazon-Metriken für eine Umgebung.

Verwenden der EB-CLI zur Überwachung des Umgebungszustands

Das Elastic Beanstalk Command Line Interface (EB CLI) ist ein Befehlszeilentool für die Verwaltung von AWS Elastic Beanstalk Umgebungen. Sie können die EB CLI auch zum Überwachen des Zustands der Umgebung in Echtzeit einsetzen. Dabei bietet sie mehr Granularität, als derzeit in der Elastic Beanstalk-Konsole verfügbar ist.

Nach der Installation und Konfiguration der EB-CLI können Sie eine neue Umgebung starten und Ihren Code mit dem eb create Befehl darauf bereitstellen. Wenn Sie bereits über eine Umgebung verfügen, die Sie in der Elastic Beanstalk-Konsole erstellt haben, können Sie die EB CLI anfügen, indem Sie eb init in einem Projektordner ausführen und die Anweisungen auf dem Bildschirm befolgen (der Projektordner kann leer sein).

> **Important**
>
> Stellen Sie sicher, dass Sie die neueste Version der EB CLI verwenden, indem Sie `pip install` mit der `--upgrade`-Option ausführen:
>
> ```
> $ sudo pip install --upgrade awsebcli
> ```
>
> Vollständige Anweisungen zur EB CLI-Installation finden Sie unter EB CLI mit Setup-Skript installieren (empfohlen).

Zum Verwenden der EB CLI für die Zustandsüberwachung Ihrer Umgebung müssen Sie zuerst einen lokalen Projektordner konfigurieren, indem Sie eb init ausführen und die Anweisungen befolgen. Vollständige Anweisungen finden Sie unter Konfigurieren der EB CLI.

Wenn Sie bereits über eine Umgebung in Elastic Beanstalk verfügen und die EB CLI für ihre Zustandsüberwachung verwenden möchten, fügen Sie sie mit diesem Verfahren die vorhandene Umgebung an.

So fügen Sie die EB CLI an eine vorhandene Umgebung an

1. Öffnen Sie ein Befehlszeilen-Terminal und navigieren Sie zu Ihrem Benutzerordner.
2. Erstellen und öffnen Sie einen neuen Ordner für Ihre Umgebung.
3. Führen Sie den eb init-Befehl aus und wählen Sie dann die Anwendung und die Umgebung, deren Zustand Sie überwachen möchten. Wenn Sie nur eine Umgebung haben, auf der die ausgewählte Anwendung ausgeführt wird, wählt die EB CLI diese automatisch aus, und Sie müssen die Umgebung nicht auswählen, wie im folgenden Beispiel gezeigt.

```
~/project$ eb init
Select an application to use
1) elastic-beanstalk-example
2) [ Create new Application ]
(default is 2): 1
Select the default environment.
You can change this later by typing "eb use [environment_name]".
1) elasticBeanstalkEx2-env
2) elasticBeanstalkExa-env
(default is 1): 1
```

So überwachen Sie den Zustand mit der EB CLI

1. Öffnen Sie eine Befehlszeile und navigieren Sie zu Ihrem Projektordner.
2. Führen Sie den eb health-Befehl zum Anzeigen des Zustands der Instances in Ihrer Umgebung aus. In diesem Beispiel gibt es fünf Instances, die in einer Linux-Umgebung ausgeführt werden.

```
~/project $ eb health
 elasticBeanstalkExa-env                                Ok
 2015-07-08 23:13:20
WebServer
    Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        5        0        0        0        0        0        0

  instance-id   status     cause
                                          health
    Overall     Ok
  i-d581497d    Ok
  i-d481497c    Ok
  i-136e00c0    Ok
  i-126e00c1    Ok
  i-8b2cf575    Ok

  instance-id   r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75
 p50     p10                                requests
    Overall     671.8   100.0    0.0    0.0    0.0    0.003    0.002    0.001
 0.001   0.000
  i-d581497d    143.0    1430      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-d481497c    128.8    1288      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-136e00c0    125.4    1254      0      0      0    0.004    0.002    0.001
 0.001   0.000
  i-126e00c1    133.4    1334      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-8b2cf575    141.2    1412      0      0      0    0.003    0.002    0.001
 0.001   0.000

  instance-id   type       az   running     load 1  load 5      user%  nice%
 system%  idle%   iowait%                            cpu
  i-d581497d    t2.micro   1a   12 mins        0.0    0.04        6.2    0.0
 1.0   92.5       0.1
```

```
  i-d481497c    t2.micro   1a   12 mins       0.01    0.09        5.9    0.0
1.6   92.4       0.1
  i-136e00c0    t2.micro   1b   12 mins       0.15    0.07        5.5    0.0
0.9   93.2       0.0
  i-126e00c1    t2.micro   1b   12 mins       0.17    0.14        5.7    0.0
1.4   92.7       0.1
  i-8b2cf575    t2.micro   1c   1 hour        0.19    0.08        6.5    0.0
1.2   92.1       0.1

  instance-id   status     id   version              ago
                                      deployments
  i-d581497d    Deployed   1    Sample Application   12 mins
  i-d481497c    Deployed   1    Sample Application   12 mins
  i-136e00c0    Deployed   1    Sample Application   12 mins
  i-126e00c1    Deployed   1    Sample Application   12 mins
  i-8b2cf575    Deployed   1    Sample Application   1 hour
```

In diesem Beispiel gibt es eine Instance, die in einer Windows-Umgebung ausgeführt wird.

```
~/project $ eb health
 WindowsSampleApp-env                                  Ok
      2018-05-22 17:33:19
WebServer                                              IIS 10.0 running on 64bit
 Windows Server 2016/2.2.0
  total      ok    warning  degraded  severe    info   pending  unknown
    1        1        0        0        0        0        0        0

  instance-id           status     cause
                                          health
    Overall             Ok
  i-065716fba0e08a351   Ok

  instance-id           r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90
 p75     p50     p10                         requests
    Overall              13.7   100.0    0.0    0.0    0.0    1.403    0.970
 0.710   0.413   0.079
  i-065716fba0e08a351     2.4   100.0    0.0    0.0    0.0    1.102*   0.865
 0.601   0.413   0.091

  instance-id           type       az   running     % user time    % privileged
 time  % idle time                                  cpu
  i-065716fba0e08a351   t2.large   1b   4 hours            0.2
 0.1         99.7
```

```
  instance-id           status     id   version              ago
                                      deployments
  i-065716fba0e08a351   Deployed   2    Sample Application   4 hours
```

Lesen der Ausgabe

Die Ausgabe zeigt den Namen der Umgebung, den Gesamtzustand der Umgebung und das aktuelle Datum oben im Bildschirm an.

```
elasticBeanstalkExa-env                                  Ok
 2015-07-08 23:13:20
```

In den nächsten drei Zeilen werden der Umgebungstyp (in diesem Fall“ WebServer "), die Konfiguration (Ruby 2.1 mit Puma) und eine Aufschlüsselung der Anzahl der Instanzen in jedem der sieben Status angezeigt.

```
WebServer
 Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        5        0        0        0        0        0        0
```

Der Rest der Ausgabe in vier Abschnitte unterteilt. Der erste zeigt den Status und den Grund des Status für die gesamte Umgebung und anschließend für jede Instance an. Das folgende Beispiel zeigt zwei Instances in der Umgebung mit dem Status `Info` und einem Grund, der besagt, dass eine Bereitstellung gestartet wurde.

```
  instance-id    status     cause
                                    health
    Overall      Ok
  i-d581497d     Info       Performing application deployment (running for 3 seconds)
  i-d481497c     Info       Performing application deployment (running for 3 seconds)
  i-136e00c0     Ok
  i-126e00c1     Ok
  i-8b2cf575     Ok
```

Weitere Informationen über Zustand und Farben finden Sie unter Farben und Status in Zustandsangaben.

Der Abschnitt requests enthält Informationen aus den Webserverprotokollen zu jeder Instance. In diesem Beispiel nimmt jede Instance Anfragen normal an und es gibt keine Fehler.

```
 instance-id    r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75     p50
   p10                                requests
   Overall      13.7    100.0    0.0    0.0    0.0    1.403    0.970    0.710   0.413
0.079
i-d581497d     2.4     100.0    0.0    0.0    0.0    1.102*   0.865    0.601   0.413
0.091
i-d481497c     2.7     100.0    0.0    0.0    0.0    0.842*   0.788    0.480   0.305
0.062
i-136e00c0     4.1     100.0    0.0    0.0    0.0    1.520*   1.088    0.883   0.524
0.104
i-126e00c1     2.2     100.0    0.0    0.0    0.0    1.334*   0.791    0.760   0.344
0.197
i-8b2cf575     2.3     100.0    0.0    0.0    0.0    1.162*   0.867    0.698   0.477
0.076
```

Der Abschnitt cpu enthält Betriebssystemmetriken für jede Instance. Die Ausgabe unterscheidet sich je nach Betriebssystem. Hier sehen Sie die Ausgabe für Linux-Umgebungen.

```
 instance-id   type       az   running     load 1  load 5      user%  nice%  system%
idle%   iowait%                            cpu
 i-d581497d    t2.micro   1a   12 mins       0.0    0.03        0.2    0.0      0.0
99.7       0.1
 i-d481497c    t2.micro   1a   12 mins       0.0    0.03        0.3    0.0      0.0
99.7       0.0
 i-136e00c0    t2.micro   1b   12 mins       0.0    0.04        0.1    0.0      0.0
99.9       0.0
 i-126e00c1    t2.micro   1b   12 mins      0.01    0.04        0.2    0.0      0.0
99.7       0.1
 i-8b2cf575    t2.micro   1c   1 hour        0.0    0.01        0.2    0.0      0.1
99.6       0.1
```

Und hier die Ausgabe für Windows-Umgebungen.

```
 instance-id           type       az   running     % user time    % privileged time  %
idle time
 i-065716fba0e08a351   t2.large   1b   4 hours            0.2                  0.0
    99.8
```

Weitere Informationen über die dargestellten Server- und Betriebssystemmetriken finden Sie unter Instance-Metriken.

Der letzte Abschnitt deployments zeigt den Bereitstellungsstatus jeder Instance. Wenn eine fortlaufende Bereitstellung fehlschlägt, können Sie die Bereitstellung-ID, den Status und die angezeigte Versionsbezeichnung verwenden, um Instances in Ihrer Umgebung zu identifizieren, die mit der falschen Version ausgeführt werden.

```
  instance-id   status     id   version              ago
                                  deployments
  i-d581497d    Deployed   1    Sample Application   12 mins
  i-d481497c    Deployed   1    Sample Application   12 mins
  i-136e00c0    Deployed   1    Sample Application   12 mins
  i-126e00c1    Deployed   1    Sample Application   12 mins
  i-8b2cf575    Deployed   1    Sample Application   1 hour
```

Interaktive Ansicht des Zustands

Der eb health-Befehl zeigt einen Snapshot des Zustands Ihrer Umgebung. Um die angezeigten Informationen alle zehn Sekunden zu aktualisieren, verwenden Sie die `--refresh`-Option.

```
$ eb health --refresh
 elasticBeanstalkExa-env                          Ok
 2015-07-09 22:10:04 (1 secs)
WebServer
          Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5         5        0         0         0         0         0         0

  instance-id   status     cause
                                     health
    Overall     Ok
  i-bb65c145    Ok         Application deployment completed 35 seconds ago and took 26
 seconds
  i-ba65c144    Ok         Application deployment completed 17 seconds ago and took 25
 seconds
  i-f6a2d525    Ok         Application deployment completed 53 seconds ago and took 26
 seconds
  i-e8a2d53b    Ok         Application deployment completed 32 seconds ago and took 31
 seconds
  i-e81cca40    Ok
```

```
  instance-id   r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75     p50
   p10                              requests
    Overall     671.8   100.0    0.0    0.0    0.0    0.003    0.002    0.001   0.001
0.000
 i-bb65c145    143.0    1430      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-ba65c144    128.8    1288      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-f6a2d525    125.4    1254      0      0      0    0.004    0.002    0.001   0.001
0.000
 i-e8a2d53b    133.4    1334      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-e81cca40    141.2    1412      0      0      0    0.003    0.002    0.001   0.001
0.000

 instance-id   type       az   running     load 1  load 5      user%  nice%  system%
idle%   iowait%                              cpu
 i-bb65c145    t2.micro   1a   12 mins        0.0    0.03        0.2    0.0      0.0
99.7       0.1
 i-ba65c144    t2.micro   1a   12 mins        0.0    0.03        0.3    0.0      0.0
99.7       0.0
 i-f6a2d525    t2.micro   1b   12 mins        0.0    0.04        0.1    0.0      0.0
99.9       0.0
 i-e8a2d53b    t2.micro   1b   12 mins       0.01    0.04        0.2    0.0      0.0
99.7       0.1
 i-e81cca40    t2.micro   1c   1 hour         0.0    0.01        0.2    0.0      0.1
99.6       0.1

 instance-id   status     id   version              ago
                                 deployments
 i-bb65c145    Deployed   1    Sample Application   12 mins
 i-ba65c144    Deployed   1    Sample Application   12 mins
 i-f6a2d525    Deployed   1    Sample Application   12 mins
 i-e8a2d53b    Deployed   1    Sample Application   12 mins
 i-e81cca40    Deployed   1    Sample Application   1 hour

(Commands: Help,Quit, # # # #)
```

Das folgende Beispiel zeigt eine Umgebung, die kürzlich von einer auf fünf Instances hochskaliert wurde. Der Skalierungsvorgang wurde erfolgreich abgeschlossen und alle Instances bestehen nun die Zustandsprüfungen und können Anfragen annehmen. Im interaktiven Modus wird der Zustandsstatus alle zehn Sekunden aktualisiert. In der rechten oberen Ecke zählt ein Timer bis zum nächsten Update herunter.

In der linken unteren Ecke zeigt der Bericht eine Liste der Optionen an. Um den interaktiven Modus zu verlassen, drücken Sie Q. Drücken Sie die Pfeiltasten, um zu scrollen. Für eine Liste der zusätzlichen Befehle drücken Sie H.

Optionen für die Interaktive Ansicht des Zustands

Bei der interaktiven Anzeige des Zustands der Umgebung können Sie die Ansicht über die Tastatur anpassen und Elastic Beanstalk anweisen, einzelne Instances zu ersetzen oder neu zu starten. Für eine Liste der verfügbaren Befehle während der Anzeige der Zustandsberichte im interaktiven Modus drücken Sie H.

```
  up,down,home,end   Scroll vertically
  left,right         Scroll horizontally
  F                  Freeze/unfreeze data
  X                  Replace instance
  B                  Reboot instance
  <,>                Move sort column left/right
  -,+                Sort order descending/ascending
  P                  Save health snapshot data file
  Z                  Toggle color/mono mode
  Q                  Quit this program

  Views
  1                  All tables/split view
  2                  Status Table
  3                  Request Summary Table
  4                  CPU%/Load Table
  H                  This help menu


(press Q or ESC to return)
```

Grundlegende Zustandsberichte

In diesem Thema werden die Funktionen von Elastic Beanstalk Basic Health erklärt.

AWS Elastic Beanstalk verwendet Informationen aus mehreren Quellen, um festzustellen, ob Ihre Umgebung verfügbar ist, und verarbeitet Anfragen aus dem Internet. Der Zustand einer Umgebung wird durch eine von vier Farben dargestellt und auf der Umgebungsübersichtsseite der Elastic Beanstalk-Konsole angezeigt. Es ist auch über die DescribeEnvironmentsAPI und durch Aufrufen eb status mit der EB-CLI verfügbar.

Das grundlegende Zustandsberichtssystem bietet Informationen über den Zustand der Instances in einer Elastic Beanstalk-Umgebung basierend auf Zustandsprüfungen, die von Elastic Load Balancing für lastverteilte Umgebungen oder von Amazon Elastic Compute Cloud für Umgebungen mit einer Instance durchgeführt werden.

Elastic Beanstalk überprüft nicht nur den Zustand Ihrer EC2 Instances, sondern überwacht auch die anderen Ressourcen in Ihrer Umgebung und meldet fehlende oder falsch konfigurierte Ressourcen, die dazu führen können, dass Ihre Umgebung für Benutzer nicht verfügbar ist.

Die von den Ressourcen in Ihrer Umgebung gesammelten Metriken werden CloudWatch in Intervallen von fünf Minuten auf Amazon veröffentlicht. Dazu gehören Betriebssystemmetriken von EC2 und Anforderungsmetriken von Elastic Load Balancing. Auf der Monitoring-Seite der Umgebungskonsole können Sie sich Diagramme ansehen, die auf diesen CloudWatch Metriken basieren. Für grundlegende Zustandsberichte werden diese Metriken nicht verwendet, um den Zustand einer Umgebung zu bestimmen.

Themen

- Zustandsfarben
- Zustandsprüfungen von Elastic Load Balancing
- Zustandsprüfungen für Umgebungen mit einer einzelnen Instance oder Worker-Ebene
- Zusätzliche Prüfungen
- CloudWatch Amazon-Metriken

Zustandsfarben

Elastic Beanstalk meldet den Zustand einer Webserverumgebung je nachdem, wie die Anwendung, die darin ausgeführt wird, auf die Zustandsprüfung reagiert. Elastic Beanstalk nutzt eine von vier Farben zum Beschreiben des Status, wie in der folgenden Tabelle gezeigt:

Farbe	Description
Grau	Ihre Umgebung wird aktualisiert.
Grün	Ihre Umgebung ist die letzte Zustandsprüfung bestanden. Mindestens eine Instance in Ihrer Umgebung ist verfügbar und verarbeitet Anfragen.

Farbe	Description
Gelb	Ihre Umgebung hat eine oder mehrere Zustandsprüfungen nicht bestanden. Einige Anfragen in Ihrer Umgebung schlagen fehl.
Rot	Ihre Umgebung hat drei oder mehr Zustandsprüfungen nicht bestanden oder eine Umgebungsressource ist nicht mehr verfügbar. Anfragen schlagen durchgängig fehl.

Diese Beschreibungen gelten nur für Umgebungen, die grundlegende Zustandsberichte verwenden. Unter Farben und Status in Zustandsangaben finden Sie Details im Zusammenhang mit der erweiterten Zustandsprüfung.

Zustandsprüfungen von Elastic Load Balancing

In einer Umgebung mit Lastenausgleich sendet Elastic Load Balancing alle 10 Sekunden eine Anfrage an jede Instance in einer Umgebung, um zu bestätigen, dass diese stabil sind. Standardmäßig ist der Load Balancer so konfiguriert, dass er eine TCP-Verbindung auf Port 80 öffnet. Wenn die Instance die Verbindung anerkennt, wird sie als stabil eingestuft.

Sie können diese Einstellung überschreiben, indem Sie eine vorhandene Ressource in Ihrer Anwendung angeben. Wenn Sie einen Pfad angeben, wie `/health`, ist die Zustandsprüfungs-URL auf `HTTP:80/health` festgelegt. Die Zustandsprüfungs-URL sollte auf einen Pfad gesetzt sein, der immer von Ihrer Anwendung bedient wird. Wenn eine statische Webseite festgelegt ist, die vom Webserver vor Ihrer Anwendung bereitgestellt oder zwischengespeichert wird, zeigen Zustandsprüfungen keine Probleme mit dem Anwendungsserver oder Webcontainer. Anweisungen zum Ändern der Zustandsprüfungs-URL finden Sie unter Gesundheitscheck.

Wenn eine Zustandsprüfungs-URL konfiguriert ist, erwartet Elastic Load Balancing eine GET-Anfrage, auf die die Antwort `200 OK` zurückgegeben werden soll. Die Anwendung besteht die Zustandsprüfung nicht, wenn sie nicht innerhalb von fünf Sekunden reagiert oder einen anderen HTTP-Statuscode zurückgibt. Nach 5 aufeinanderfolgenden Fehlern bei der Zustandsprüfung nimmt Elastic Load Balancing die Instanz außer Betrieb.

Weitere Informationen zu den Zustandsprüfungen von Elastic Load Balancing finden Sie unter Zustandsprüfungen im Elastic Load Balancing-Benutzerhandbuch.

> **Note**
>
> Wenn Sie eine URL für die Zustandsprüfung konfigurieren, ändert dies das Verhalten einer Auto Scaling-Gruppe in der Umgebung nicht. Eine fehlerhafte Instance wird aus dem Load Balancer entfernt, aber nicht automatisch durch Amazon EC2 Auto Scaling ersetzt, es sei denn, Sie konfigurieren Amazon EC2 Auto Scaling so, dass der Elastic Load Balancing Health Check als Grundlage für das Ersetzen von Instances verwendet wird. Informationen zur Konfiguration von Amazon EC2 Auto Scaling zum Ersetzen von Instances, die eine Elastic Load Balancing Balancing-Zustandsprüfung nicht bestehen, finden Sie unterEinstellung für die Auto Scaling Scaling-Integritätsprüfung für Ihre Elastic Beanstalk Beanstalk-Umgebung.

Zustandsprüfungen für Umgebungen mit einer einzelnen Instance oder Worker-Ebene

In einer Single-Instance- oder Worker-Tier-Umgebung bestimmt Elastic Beanstalk den Zustand der Instance, indem es ihren EC2 Amazon-Instance-Status überwacht. Die Integritätseinstellungen von Elastic Load Balancing, einschließlich der URLs HTTP-Zustandsprüfung, können in diesen Umgebungstypen nicht verwendet werden.

Weitere Informationen zu EC2 Amazon-Instance-Statuschecks finden Sie unter Monitoring Instances with Status Checks im EC2 Amazon-Benutzerhandbuch.

Zusätzliche Prüfungen

Zusätzlich zu den Elastic Load Balancing -Zustandsprüfungen überwacht Elastic Beanstalk-Ressourcen in Ihrer Umgebung und ändert den Status in Rot, wenn sie nicht bereitgestellt werden, nicht korrekt konfiguriert sind oder nicht mehr verfügbar sind. Diese Prüfungen bestätigen, dass:

- Die Auto Scaling-Gruppe der Umgebung verfügbar ist und über mindestens eine Instance verfügt.
- Die Sicherheitsgruppe der Umgebung ist verfügbar und so konfiguriert, dass eingehender Datenverkehr auf Port 80 zulässig ist.
- Der Umgebungs-CNAME ist vorhanden und weist auf den rechten Load Balancer.
- In einer Worker-Umgebung wird die Amazon Simple Queue Service (Amazon SQS)-Warteschlange mindestens einmal alle drei Minuten abgefragt.

CloudWatch Amazon-Metriken

Bei grundlegenden Gesundheitsberichten veröffentlicht der Elastic Beanstalk-Service keine Metriken auf Amazon. CloudWatch Die CloudWatch Metriken, die zur Erstellung von Diagrammen auf der Monitoring-Seite der Umgebungskonsole verwendet werden, werden von den Ressourcen in Ihrer Umgebung veröffentlicht.

EC2 Veröffentlicht beispielsweise die folgenden Metriken für die Instances in der Auto Scaling Scaling-Gruppe Ihrer Umgebung:

`CPUUtilization`

Prozentsatz der Recheneinheiten, die zurzeit verwendet werden.

`DiskReadBytes`, `DiskReadOps`, `DiskWriteBytes`, `DiskWriteOps`

Anzahl der gelesenen und geschriebenen Bytes und die Anzahl der Lese- und Schreibvorgänge.

`NetworkIn`, `NetworkOut`

Anzahl der gesendeten und empfangenen Bytes.

Elastic Load Balancing veröffentlicht die folgenden Metriken für den Load Balancer Ihrer Umgebung:

`BackendConnectionErrors`

Anzahl der fehlgeschlagenen Verbindung zwischen dem Load Balancer und den Umgebung-Instances.

`HTTPCode_Backend_2XX`, `HTTPCode_Backend_4XX`

Anzahl der erfolgreichen (2XX) und Client-Fehler-Antwortcodes (4XX), die von Instances in Ihrer Umgebung generiert wurden.

`Latency`

Anzahl der Sekunden zwischen dem Zeitpunkt, an dem der Load Balancer eine Anfrage an eine Instance weiterleitet, und dem Zeitpunkt, zu dem die Antwort empfangen wird.

`RequestCount`

Anzahl der abgeschlossenen Anfragen.

Diese Listen sind nicht vollständig. Eine vollständige Liste der Kennzahlen, die für diese Ressourcen gemeldet werden können, finden Sie in den folgenden Themen im Amazon CloudWatch Developer Guide:

Kennzahlen

Namespace	Thema
AWS::ElasticLoadBalancing::LoadBalancer	Elastic Load Balancing-Metriken und -Ressourcen
AWS::AutoScaling::AutoScalingGruppe	Amazon Elastic Compute Cloud-Metriken und -Ressourcen
AWS::SQS::Queue	Amazon SQS-Metriken und -Ressourcen
AWS: :RDS:: DBInstance	Amazon RDS-Dimensionen und -Metriken

Worker-Umgebung – Zustandsmetrik

Nur für Worker-Umgebungen veröffentlicht der SQS-Daemon eine benutzerdefinierte Metrik für den Zustand der Umgebung CloudWatch, wobei der Wert 1 Grün ist. Sie können die CloudWatch Gesundheitsmetrikdaten in Ihrem Konto mithilfe des `ElasticBeanstalk/SQSD` Namespace überprüfen. Die Metrikdimension ist `EnvironmentName` und der Metrikname lautet `Health`. Alle Instances veröffentlichen ihre Metriken auf dem gleichen Namespace.

Damit der Daemon Metriken veröffentlichen kann, muss das Instance-Profil der Umgebung die Berechtigung zum Aufrufen von `cloudwatch:PutMetricData` haben. Diese Berechtigung ist im Standard-Instance-Profil enthalten. Weitere Informationen finden Sie unter Elastic Beanstalk Instance-Profile verwalten.

Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk

In diesem Abschnitt werden die Funktionen der Elastic Beanstalk Enhanced Health-Funktion erläutert.

Enhanced Health Reporting ist eine Funktion, die Sie in Ihrer Umgebung aktivieren können, AWS Elastic Beanstalk um zusätzliche Informationen über Ressourcen in Ihrer Umgebung zu sammeln. Elastic Beanstalk analysiert die gesammelten Informationen, um ein besseres Bild der Gesamtumgebungsintegrität zu erhalten und um bei der Identifizierung von Problemen zu helfen, die dazu führen können, dass Ihre Anwendung nicht mehr verfügbar ist.

Zusätzlich zu Änderungen bei der Funktionsweise von Zustandsfarben, fügt der erweiterte Zustandsbericht einen Status-Deskriptor hinzu, der auf den Schweregrad von Problemen hinweist, wenn eine Umgebung gelb oder rot ist. Wenn weitere Informationen über den aktuellen Status verfügbar sind, können Sie die Schaltfläche Causes wählen, um detaillierte Zustandsinformationen auf der Seite Zustand anzuzeigen.

Um detaillierte Integritätsinformationen zu den Amazon-EC2-Instances bereitzustellen, die in Ihrer Umgebung ausgeführt werden, enthält Elastic Beanstalk für jede Plattformversion, die erweiterte Integritätsberichte unterstützt, einen Integritäts-Agenten im Amazon-Machine-Image (AMI). Der Integritäts-Agent überwacht Webserverprotokolle und Systemmetriken und leitet sie an den Elastic-Beanstalk-Service weiter. Elastic Beanstalk analysiert diese Metriken und Daten von Elastic-Load-Balancing und Amazon-EC2-Auto-Scaling, um ein Gesamtbild der Integrität einer Umgebung zu erhalten.

Zusätzlich zum Sammeln und Präsentieren von Informationen über die Ressourcen Ihrer Umgebung überwacht Elastic Beanstalk die Ressourcen in Ihrer Umgebung auf verschiedene Fehlerbedingungen und bietet Benachrichtigungen, um Fehler zu vermeiden und Konfigurationsprobleme zu lösen. Faktoren, die den Zustand Ihrer Umgebung beeinflussen, umfassen die Ergebnisse jeder einzelnen Anforderung, die von Ihrer Anwendung bereitgestellt wird, Metriken des Betriebssystems Ihrer Instance und den Status der letzten Bereitstellung.

Sie können den Integritätsstatus in Echtzeit auf der Umgebungsübersichtsseite in der Elastic-Beanstalk-Konsole oder mit dem Befehl eb health in der Elastic-Beanstalk-Befehlszeilenschnittstelle (EB-CLI) anzeigen. Um den Zustand der Umgebung und Instance im Laufe der Zeit aufzuzeichnen und zu verfolgen, können Sie Ihre Umgebung so konfigurieren, dass die von Elastic Beanstalk gesammelten Informationen für erweiterte Statusberichte CloudWatch als benutzerdefinierte Metriken an Amazon veröffentlicht werden. CloudWatch Gebühren für benutzerdefinierte Metriken fallen für alle Metriken an, mit Ausnahme von`EnvironmentHealth`, was kostenlos ist.

> **Hinweise zur Windows-Plattform**
>
> Wenn Sie erweiterte Zustandsberichte für eine Windows Server-Umgebung aktivieren, ändern Sie nicht die IIS-Protokollierungskonfiguration. Damit die erweiterte

Statusüberwachung einwandfrei funktioniert, muss die IIS-Protokollierung mit dem W3C-Format und den Protokollereigniszielen ETW event only (Nur ETW-Ereignis) oder Both log file and ETW event (Sowohl Protokolldatei als auch ETW-Ereignis) konfiguriert werden. Außerdem dürfen Sie den Windows-Service des Elastic-Beanstalk-Integritäts-Agenten auf keiner der Instances Ihrer Umgebung deaktivieren oder anhalten. Damit erweiterte Zustandsinformationen für eine Instance gesammelt und als Bericht zusammengefasst werden, muss dieser Service aktiviert sein und ausgeführt werden.

Wenn Sie zum ersten Mal eine Umgebung erstellen, fordert Elastic Beanstalk Sie auf, die erforderlichen Rollen zu erstellen, und aktiviert standardmäßig erweiterte Zustandsberichte. Lesen Sie Details darüber, wie die erweiterten Integritätsberichte funktionieren, oder besuchen Sie Aktivieren der erweiterten Elastic-Beanstalk-Integritätsberichte, um die Funktion sofort zu verwenden.

Themen

- Der Elastic-Beanstalk-Integritäts-Agent
- Faktoren bei der Bestimmung des Instance- und Umgebungszustands
- Anpassung der Regel für die Zustandsprüfung
- Rollen in erweiterten Zustandsberichten
- Erweiterte Zustandsautorisierung
- Ereignisse in erweiterten Zustandsberichten
- Verhalten der erweiterten Zustandsberichte bei Aktualisierungen, Bereitstellungen und Skalierung
- Aktivieren der erweiterten Elastic-Beanstalk-Integritätsberichte
- Erweiterte Integritätsüberwachung mithilfe der Environment Management Console
- Farben und Status in Zustandsangaben
- Instance-Metriken
- Konfigurieren von Regeln für den erweiterten Zustand einer Umgebung
- Veröffentlichen CloudWatch benutzerdefinierter Amazon-Metriken für eine Umgebung
- Verwenden der erweiterten Integritätsberichte mit der Elastic-Beanstalk-API
- Format der Protokolle der erweiterten Zustandsberichte
- Benachrichtigungen und Fehlerbehebung
- KI-gestützte Umgebungsanalyse

Der Elastic-Beanstalk-Integritäts-Agent

Der Elastic-Beanstalk-Integritäts-Agent ist ein Daemon-Prozess (oder in Windows-Umgebungen ein Service), der auf den einzelnen Amazon-EC2-Instances in Ihrer Umgebung ausgeführt wird, das Betriebssystem und Integritätsmetriken auf Anwendungsebene überwacht und Probleme an Elastic Beanstalk meldet. Der Zustandsagent ist in allen Plattformversionen ab Version 2.0 einer jeden Plattform enthalten.

Der Health Agent meldet ähnliche Metriken wie die, die Amazon EC2 Auto Scaling und Elastic Load Balancing im Rahmen der grundlegenden Statusberichterstattung veröffentlicht haben, einschließlich CPU-Last, HTTP-Codes und Latenz. CloudWatch Der Integritäts-Agent meldet jedoch direkt an Elastic Beanstalk – mit höherer Granularität und Häufigkeit als die grundlegenden Integritätsberichte.

Für grundlegende Zustandsberichte werden diese Metriken alle fünf Minuten veröffentlicht und können mit Diagrammen in der Environment Management Console überwacht werden. Bei den erweiterten Integritätsberichten meldet der Elastic-Beanstalk-Integritäts-Agent die Metriken alle 10 Sekunden an Elastic Beanstalk. Elastic Beanstalk verwendet die Metriken des Integritäts-Agenten, um den Integritätsstatus jeder Instance in der Umgebung zu bestimmen, und, in Kombination mit anderen Faktoren, um die Gesamtintegrität der Umgebung zu bestimmen.

Der allgemeine Zustand der Umgebung kann in Echtzeit auf der Umgebungsübersichtsseite der Elastic Beanstalk-Konsole angezeigt werden und wird alle 60 Sekunden CloudWatch von Elastic Beanstalk veröffentlicht. Sie können detaillierte Metriken, die vom Zustandsagenten in Echtzeit gemeldet werden, mit dem eb health-Befehl in der EB-CLI anzeigen.

Gegen eine zusätzliche Gebühr können Sie wählen, ob Sie einzelne Metriken auf Instanz- und Umgebungsebene alle 60 Sekunden veröffentlichen möchten. CloudWatch In CloudWatch veröffentlichte Metriken können dann verwendet werden, um Überwachungsdiagramme in der Environment Management Console zu erstellen.

Für erweiterte Zustandsberichte fällt nur eine Gebühr an, wenn Sie erweiterte Zustandsmetriken in CloudWatch veröffentlichen. Wenn Sie erweiterte Zustandsberichte verwenden, werden die grundlegenden Zustandsmetriken noch kostenlos veröffentlicht, auch wenn Sie keine erweiterten Zustandsmetriken veröffentlichen.

Unter Instance-Metriken finden Sie Details zu Metriken, die vom Zustandsagenten veröffentlicht werden. Einzelheiten zur Veröffentlichung erweiterter Integritätskennzahlen auf finden Sie CloudWatch unter. Veröffentlichen CloudWatch benutzerdefinierter Amazon-Metriken für eine Umgebung

Faktoren bei der Bestimmung des Instance- und Umgebungszustands

Zusätzlich zu den grundlegenden Systemprüfungen der Integritätsberichte, einschließlich Zustandsprüfungen von Elastic Load Balancing und Ressourcenüberwachung, sammeln die erweiterten Integritätsberichte von Elastic Beanstalk zusätzliche Daten zur Integrität der Instances in Ihrer Umgebung. Dazu zählen Betriebssystemmetriken, Serverprotokolle und der Zustand laufender Umgebungsoperationen, wie Bereitstellungen und Updates. Das Integritätsberichtesystem von Elastic Beanstalk kombiniert Informationen aus allen verfügbaren Quellen und analysiert sie, um die Gesamtintegrität der Umgebung zu bestimmen.

Operationen und Befehle

Wenn Sie eine Operation in Ihrer Umgebung durchführen, z. B. die Bereitstellung einer neuen Version einer Anwendung, nimmt Elastic Beanstalk mehrere Änderungen vor, die sich auf den Integritätsstatus der Umgebung auswirken.

Wenn Sie beispielsweise eine neue Version einer Anwendung in einer Umgebung bereitstellen, in der mehreren Instances ausgeführt werden, werden beim Überwachen des Zustands der Umgebung mithilfe der EB-CLI möglicherweise Meldung ähnlich der folgenden angezeigt:

```
  id              status     cause
    Overall       Info       Command is executing on 3 out of 5 instances
  i-bb65c145      Pending    91 % of CPU is in use. 24 % in I/O wait
                             Performing application deployment (running for 31 seconds)
  i-ba65c144      Pending    Performing initialization (running for 12 seconds)
  i-f6a2d525      Ok         Application deployment completed 23 seconds ago and took 26
seconds
  i-e8a2d53b      Pending    94 % of CPU is in use. 52 % in I/O wait
                             Performing application deployment (running for 33 seconds)
  i-e81cca40      Ok
```

In diesem Beispiel lautet der Gesamtstatus der Umgebung Ok und die Ursache dieses Status ist, dass der Befehl drei von fünf Instances ausführt. Drei der Instances in der Umgebung haben den Status Pending, was bedeutet, dass ein Vorgang ausgeführt wird.

Wenn ein Vorgang abgeschlossen ist, meldet Elastic Beanstalk zusätzliche Informationen über die Operation. Zum Beispiel zeigt Elastic Beanstalk die folgenden Informationen zu einer Instance, die bereits mit der neuen Version der Anwendung aktualisiert wurde:

```
i-f6a2d525     Ok        Application deployment completed 23 seconds ago and took 26
 seconds
```

Instance-Zustandsinformationen umfassen auch Details zur aktuellen Bereitstellung für jede Instance in Ihrer Umgebung. Jede Instance meldet eine Bereitstellungs-ID und einen Status. Die Bereitstellungs-ID ist eine ganze Zahl, die sich jedes Mal erhöht, wenn Sie eine neue Version Ihrer Anwendung bereitstellen oder Einstellungen für die Konfigurationsoptionen für die Instance, wie Umgebungsvariablen, ändern. Sie können die Bereitstellungsinformationen zur Identifizierung von Instances verwenden, auf denen die falsche Version Ihrer Anwendung ausgeführt wird, nachdem eine fortlaufende Bereitstellung fehlgeschlagen ist.

In der Spalte mit der Ursache schließt Elastic Beanstalk Informationsmeldungen zu erfolgreichen Operationen und anderen fehlerfreien Zuständen in mehreren Integritätsprüfungen ein, diese werden aber nicht unbegrenzt beibehalten. Ursachen für fehlerhafte Umgebungsstatus verbleiben, bis die Umgebung in einen fehlerfreien Status zurückkehrt.

Befehls-Timeout

Elastic Beanstalk wendet einen Befehls-Timeout ab dem Zeitpunkt an, an dem eine Operation beginnt, damit eine Instance in einen fehlerfreien Zustand übergehen kann. Dieser Befehls-Timeout wird in der Update- und Bereitstellungskonfiguration Ihrer Umgebung festgelegt (im aws:elasticbeanstalk: Befehl-Namespace) und standardmäßig auf zehn Minuten gesetzt.

Während fortlaufenden Updates wendet Elastic Beanstalk einen separaten Timeout auf jeden Stapel in der Operation an. Dieser Timeout wird als Teil der fortlaufenden Update-Konfiguration der Umgebung (im aws:autoscaling: updatepolicy: rollingupdate-Namespace) festgelegt. Wenn alle Instances im Stapel innerhalb des Timeouts für fortlaufende Updates fehlerfrei sind, fährt die Operation mit dem nächsten Stapel fort. Wenn dies nicht der Fall ist, schlägt die Operation fehl.

> **Note**
>
> Wenn Ihre Anwendung Integritätsprüfungen nicht mit dem Status OK besteht, sie auf einer anderen Stufe jedoch stabil ist, können Sie die `HealthCheckSuccessThreshold`-Option im `aws:elasticbeanstalk:command namespace`-Namespace so festlegen, dass die Stufe geändert wird, auf der Elastic Beanstalk eine Instance als stabil betrachtet.

Damit eine Webserverumgebung als fehlerfrei angesehen wird, muss jede Instance in der Umgebung oder im Stapel zwölf aufeinanderfolgende Zustandsprüfungen im Verlauf von zwei Minuten bestehen.

Bei einer Umgebung mit Worker-Ebene muss jede Instance 18 Zustandsprüfungen bestehen. Bevor ein Befehls-Timeout auftritt, senkt Elastic Beanstalk den Integritätsstatus einer Umgebung nicht, wenn die Integritätsprüfungen fehlschlagen. Wenn die Instances in der Umgebung innerhalb des Befehls-Timeouts fehlerfrei werden, ist die Operation erfolgreich.

HTTP-Anforderungen

Wenn keine Operation in einer Umgebung ausgeführt wird, ist die primäre Quelle für Informationen über den Instance- und Umgebungszustand die Webserverprotokolle für jede Instance. Um die Integrität einer Instance und die Gesamtintegrität der Umgebung zu prüfen, berücksichtigt Elastic Beanstalk die Anzahl der Anforderungen, das Ergebnis jeder Anforderung und die Geschwindigkeit, mit der jede Anforderung gelöst wurde.

Auf Linux-basierten Plattformen liest und analysiert Elastic Beanstalk Web-Server-Protokolle zum Abrufen von Informationen über HTTP-Anfragen. Auf der Windows Server-Plattform erhält Elastic Beanstalk diese Informationen direkt von dem IIS-Webserver.

Ihre Umgebung verfügt möglicherweise nicht über einen aktiven Webserver. So beinhaltet die Multicontainer Docker-Plattform beispielsweise keinen Webserver. Andere Plattformen enthalten einen Webserver, der von Ihrer Anwendung möglicherweise deaktiviert wird. In diesen Fällen ist für Ihre Umgebung eine zusätzliche Konfiguration erforderlich, um dem Elastic Beanstalk-Integritäts-Agenten Protokolle in dem Format zur Verfügung zu stellen, das zur Weiterleitung von Integritätsinformationen an den Elastic-Beanstalk-Service benötigt wird. Details dazu finden Sie unter Format der Protokolle der erweiterten Zustandsberichte.

Betriebssystemmetriken

Elastic Beanstalk überwacht Betriebssystemmetriken, die vom Integritäts-Agenten gemeldet wurden, um Instances zu identifizieren, die konsequent geringe Systemressourcen haben.

Unter Instance-Metriken finden Sie Details zu Metriken, die vom Zustandsagenten veröffentlicht werden.

Anpassung der Regel für die Zustandsprüfung

Erweiterte Integritätsberichte von Elastic Beanstalk basieren auf mehreren Regeln, um die Integrität Ihrer Umgebung zu bestimmen. Einige dieser Regeln sind möglicherweise nicht für Ihre jeweilige Anwendung geeignet. Ein häufiger Fall ist eine Anwendung, die standardmäßig viele HTTP-4xx-Fehler zurückgibt. Elastic Beanstalk kommt unter Verwendung einer seiner Standardregeln zu dem Schluss, dass etwas schief läuft, und ändert den Integritätsstatus Ihrer Umgebung je nach Fehlerrate

von „OK“ auf „Warning (Warnung)“, „Degraded (Schwach)“ oder „Severe (Stark)“. Um diesen Fall ordnungsgemäß zu verarbeiten, gestattet Ihnen Elastic Beanstalk, diese Regel zu konfigurieren und HTTP-4xx-Fehler von Anwendungen zu ignorieren. Details hierzu finden Sie unter Konfigurieren von Regeln für den erweiterten Zustand einer Umgebung.

Rollen in erweiterten Zustandsberichten

Erweiterte Integritätsberichte erfordern zwei Rollen – eine Servicerolle für Elastic Beanstalk und ein Instance-Profil für die Umgebung. Die Service-Rolle ermöglicht es Elastic Beanstalk, in Ihrem Namen mit anderen AWS Services zu interagieren, um Informationen über die Ressourcen in Ihrer Umgebung zu sammeln. Das Instance-Profil ermöglicht es den Instances in Ihrer Umgebung, Protokolle in Amazon-S3 zu schreiben und erweiterte Integritätsinformationen an den Elastic-Beanstalk-Service zu übermitteln.

Wenn Sie eine Elastic-Beanstalk-Umgebung mit der Elastic-Beanstalk-Konsole oder der EB-CLI erstellen, erstellt Elastic Beanstalk eine Standard-Servicerolle und fügt die erforderlichen verwalteten Richtlinien an ein Standard-Instance-Profil für Ihre Umgebung an.

Wenn Sie die API, ein SDK oder die verwenden, um Umgebungen AWS CLI zu erstellen, müssen Sie diese Rollen im Voraus erstellen und sie bei der Umgebungserstellung angeben, um Enhanced Health verwenden zu können. Anweisungen zum Erstellen geeigneter Rollen für Ihre Umgebungen finden Sie unter Elastic Beanstalk Service-Rollen, Instanzprofile und Benutzerrichtlinien.

Es wird empfohlen, verwaltete Richtlinien für Ihr Instance-Profil und Ihre Servicerolle zu verwenden. Verwaltete Richtlinien sind AWS Identity and Access Management (IAM-) Richtlinien, die Elastic Beanstalk verwaltet. Durch die Verwendung verwalteter Richtlinien wird sichergestellt, dass Ihre Umgebung über alle erforderlichen Berechtigungen verfügt, um ordnungsgemäß zu funktionieren.

Für das Instance-Profil können Sie die verwalteten Richtlinien `AWSElasticBeanstalkWebTier` oder `AWSElasticBeanstalkWorkerTier` verwenden, je nachdem, ob es sich um eine Umgebung auf Webserverebene oder auf Worker-Ebene handelt. Ausführliche Informationen zu diesen beiden Richtlinien für verwaltete Instance-Profile finden Sie unter the section called “Instance-Profile”.

Erweiterte Zustandsautorisierung

Die Richtlinien, die das Elastic-Beanstalk-Instance-Profil verwaltet, enthalten die Berechtigungen der Aktion `elasticbeanstalk:PutInstanceStatistics`. Diese Aktion ist nicht Teil der Elastic-Beanstalk-API. Sie ist Teil einer anderen API, die Umgebungs-Instances intern verwenden, um erweiterte Zustandsinformationen an den Elastic-Beanstalk-Service zu übermitteln. Sie rufen diese API nicht direkt auf.

Wenn Sie eine neue Umgebung erstellen, wird die Autorisierung für die`elasticbeanstalk:PutInstanceStatistics` Aktion standardmäßig aktiviert. Um die Sicherheit Ihrer Umgebung zu erhöhen und das Spoofing von Gesundheitsdaten in Ihrem Namen zu verhindern, empfehlen wir die aktivierte Einstellung. Wenn Sie verwaltete Richtlinien für Ihr Instance-Profil verwenden, steht diese Funktion für Ihre neue Umgebung ohne weitere Konfiguration zur Verfügung. Wenn Sie anstelle einer verwalteten Richtlinie ein benutzerdefiniertes Instance-Profil verwenden, zeigt Ihre Umgebung möglicherweise den Zustandstatus Keine Daten an. Dies geschieht, weil die Instances nicht für die Aktion autorisiert sind, die erweiterte Zustandsdaten an den Service übermittelt.

Um die Aktion zu autorisieren, fügen Sie die folgende Anweisung in Ihr Instance-Profil ein.

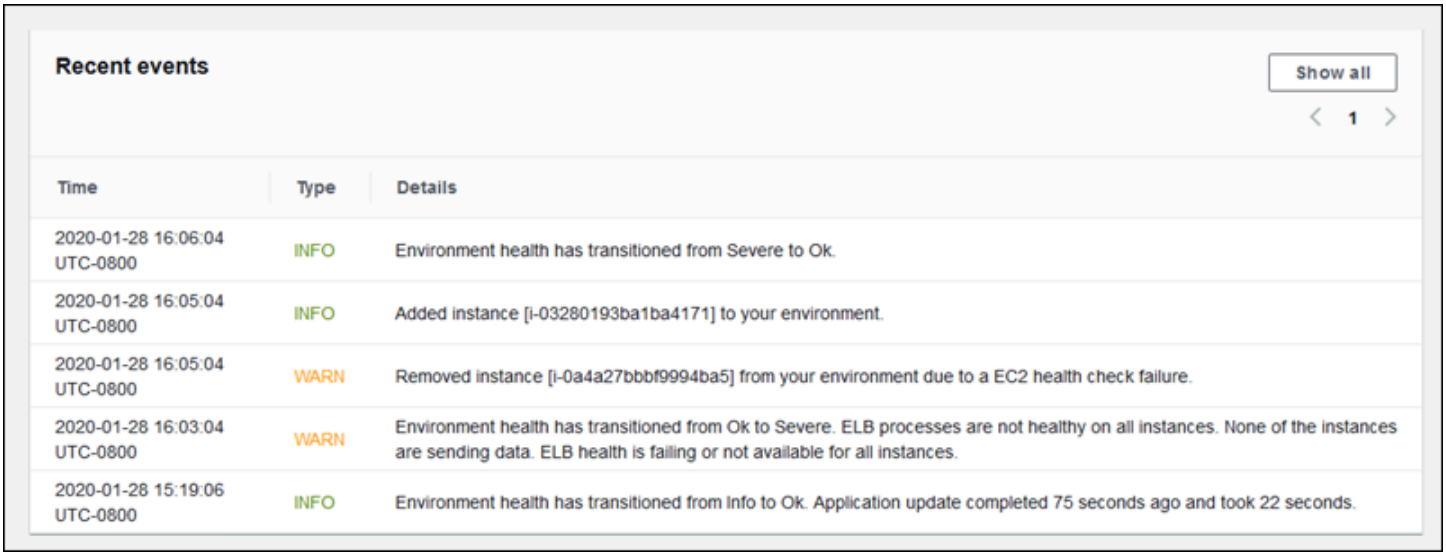
```
{
  "Sid": "ElasticBeanstalkHealthAccess",
  "Action": [
    "elasticbeanstalk:PutInstanceStatistics"
  ],
  "Effect": "Allow",
  "Resource": [
    "arn:aws:elasticbeanstalk:*:*:application/*",
    "arn:aws:elasticbeanstalk:*:*:environment/*"
  ]
}
```

Wenn Sie die erweiterte Zustandsberechtigung zu diesem Zeitpunkt nicht verwenden möchten, deaktivieren Sie sie, indem Sie die Einstellung `EnhancedHealthAuthEnabled` Option in der the section called “aws:elasticbeanstalk:healthreporting:system” Namespace zu `false`. Wenn diese Option deaktiviert ist, sind die zuvor beschriebenen Berechtigungen nicht erforderlich. Sie können sie aus dem Instanceprofil entfernen, um Zugriff auf Ihre Anwendungen und Umgebungen mit den geringsten Rechten zu erhalten.

> **Note**
>
> Bisher war die Standardeinstellung für `EnhancedHealthAuthEnabled` `false`, welches im Ergebnis die Aktion `elasticbeanstalk:PutInstanceStatistics` standardmäßig ebenfalls deaktiviert. Um diese Aktion für eine vorhandene Umgebung zu aktivieren, stellen Sie die `EnhancedHealthAuthEnabled`-Option im the section called “aws:elasticbeanstalk:healthreporting:system”-Namespace auf `true` ein. Sie können diese Option mithilfe einer Optionseinstellung in einer Konfigurationsdatei konfigurieren.

Ereignisse in erweiterten Zustandsberichten

Das System für erweiterte Zustandsberichte generiert Ereignisse, wenn eine Umgebung zwischen Zuständen wechselt. Das folgende Beispiel zeigt ausgegebene Ereignisse durch eine Umgebungsübertragung zwischen den Zuständen Info, OK und Severe (Schwerwiegend).



Wenn in einen schlechteren Zustand gewechselt wird, enthält das Ereignis der erweiterten Zustandsberichte eine Meldung mit der Ursache des Wechsels.

Nicht alle Änderungen des Status auf Instance-Ebene bewirken, dass Elastic Beanstalk ein Ereignis ausgibt. Um Fehlalarme zu verhindern, generiert Elastic Beanstalk nur dann ein integritätsbezogenes Ereignis, wenn ein Problem über mehrere Prüfungen hinweg bestehen bleibt.

Echtzeit-Integritätsinformationen auf Umgebungsebene, einschließlich Zustand, Farbe und Ursache, sind auf der Umgebungsübersichtsseite der Elastic-Beanstalk-Konsole und in der EB-CLI verfügbar. Indem Sie die EB-CLI an Ihre Umgebung anfügen und den eb health-Befehl ausführen, können Sie auch Echtzeitstatus aus jeder der Instances in Ihrer Umgebung anzeigen.

Verhalten der erweiterten Zustandsberichte bei Aktualisierungen, Bereitstellungen und Skalierung

Das Aktivieren von erweiterten Zustandsberichten kann beeinflussen, wie sich Ihre Umgebung während Konfigurations-Updates und -bereitstellungen verhält. Elastic Beanstalk schließt eine Reihe von Aktualisierungen erst dann ab, wenn alle Instances die Integritätsprüfungen durchgängig bestehen. Da die erweiterten Zustandsberichte einen höheren Standard für den Zustand anwenden und mehr Faktoren überwachen, bestehen Instances, die die grundlegende ELB-

Zustandsprüfung der Zustandsberichte bestehen, nicht notwendigerweise die Prüfung mit erweiterten Zustandsberichten. Weitere Informationen dazu, wie Zustandsprüfungen den Update-Prozess beeinflussen, finden Sie in den Themen fortlaufende Konfigurations-Updates und fortlaufende Bereitstellungen.

Erweiterte Integritätsberichte können auch die Notwendigkeit, eine ordnungsgemäße Integritätsprüfungs-URL für Elastic-Load-Balancing festzulegen, hervorheben. Wenn Ihre Umgebung nach oben skaliert wird, um den Bedarf zu erfüllen, beginnen neue Instances mit der Annahme von Anforderungen, sobald sie ausreichend ELB-Zustandsprüfungen bestehen. Wenn eine Zustandsprüfungs-URL nicht konfiguriert ist, dauert es nur 20 Sekunden, bis eine neue Instance eine TCP-Anwendung akzeptieren kann.

Wenn Ihre Anwendung den Start nicht abgeschlossen hat, bis der Load Balancer sie für ausreichend stabil für den Empfang von Datenverkehr erklärt hat, sehen Sie zahlreiche fehlerhafte Anforderungen und Ihre Umgebung besteht die Zustandsprüfungen nicht mehr. Eine Zustandsprüfungs-URL, die auf einen Pfad abzielt, der von Ihrer Anwendung verarbeitet wird, kann dieses Problem verhindern. ELB-Zustandsprüfungen schlagen so lange fehl, bis eine GET-Anforderung an die Zustandsprüfungs-URL als Statuscode 200 zurückgibt.

Aktivieren der erweiterten Elastic-Beanstalk-Integritätsberichte

In diesem Thema wird erklärt, wie erweiterte Gesundheitsberichte aktiviert werden. Es enthält Verfahren, mit denen Sie die erweiterte Gesundheitsfunktion für Ihre Umgebung mit der Elastic Beanstalk Beanstalk-Konsole, der EB-CLI und mit einer .ebextensions-Konfiguration aktivieren können.

Zu den neuen Umgebungen, die mit den neuesten Plattformversionen erstellt wurden, gehört der AWS Elastic Beanstalk Health Agent, der erweiterte Gesundheitsberichte unterstützt. Wenn Sie Ihre Umgebung in der Elastic-Beanstalk-Konsole oder mit der EB-CLI erstellen, sind erweiterte Integritätsberichte standardmäßig aktiviert. Sie können die Einstellungen der erweiterten Zustandsberichte auch im Quellcode Ihrer Anwendung mithilfe von Konfigurationsdateien festlegen.

Erweiterte Zustandsberichte erfordern ein Instance-Profil und eine Servicerolle mit den Standardberechtigungen. Wenn Sie eine Umgebung in der Elastic-Beanstalk-Konsole erstellen, legt Elastic Beanstalk die erforderlichen Rollen automatisch fest. Anweisungen zum Erstellen Ihrer ersten Umgebung finden Sie unter Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können.

Themen

- Aktivieren der erweiterten Integritätsberichte mit der Elastic-Beanstalk-Konsole

- Aktivieren der erweiterten Zustandsberichte mit der EB-CLI
- Aktivieren der erweiterten Zustandsberichte mit einer Konfigurationsdatei

Aktivieren der erweiterten Integritätsberichte mit der Elastic-Beanstalk-Konsole

So aktivieren Sie erweiterte Integritätsberichte in einer laufenden Umgebung mit der Elastic-Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).
5. Klicken Sie unter Health reporting (Zustandsberichte) für die Option System auf Enhanced (Erweitert).

   > **Note**
   >
   > Die Optionen für erweiterte Zustandsberichte werden nicht angezeigt, wenn Sie eine nicht unterstützte Plattform oder Version verwenden.

6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Die Elastic-Beanstalk-Konsole erstellt standardmäßig erweiterte Integritätsberichte, wenn Sie eine neue Umgebung mit einer Version 2 (v2)-Plattform erstellen. Sie können erweiterte Zustandsberichte deaktivieren, indem Sie die Option für Zustandsberichte während der Umgebungserstellung ändern.

So deaktivieren Sie erweiterte Integritätsberichte, wenn Sie eine Umgebung mit der Elastic-Beanstalk-Konsole erstellen

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Erstellen Sie eine Anwendung oder wählen Sie eine bestehende.

3. Erstellen Sie eine Umgebung. Wählen Sie auf der Seite Create a new environment (Neue Umgebung erstellen) zuerst Configure more options (Weitere Optionen konfigurieren) aus und klicken Sie dann auf Create environment (Umgebung erstellen).
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).
5. Klicken Sie unter Health reporting (Zustandsberichte) für die Option System auf Basic (Grundlegend).
6. Wählen Sie Speichern.

Aktivieren der erweiterten Zustandsberichte mit der EB-CLI

Wenn Sie eine neue Umgebung mit dem eb create-Befehl erstellen, aktiviert die EB-CLI erweiterte Zustandsberichte standardmäßig und wendet das Standard-Instance-Profil und die Standard-Servicerolle an.

Sie können mit der `--service-role`-Option eine andere Servicerolle nach Namen festlegen.

Wenn Sie eine Umgebung haben, auf der grundlegende Zustandsberichte auf einer Version 2 (v2)-Plattformversion ausgeführt werden, und Sie zu erweiterten Zustandsberichten wechseln möchten, führen Sie die folgenden Schritte aus.

So aktivieren Sie die erweiterten Zustandsberichte auf einer laufenden Umgebung mit der EB-CLI

1. Verwenden Sie den eb config-Befehl, um die Konfigurationsdatei im Standard-Texteditor zu öffnen.

   ```
   ~/project$ eb config
   ```

2. Suchen Sie den `aws:elasticbeanstalk:environment`-Namespace im Bereich mit den Einstellungen. Stellen Sie sicher, dass der Wert von `ServiceRole` nicht null ist und mit dem Namen Ihrer Servicerolle übereinstimmt.

   ```
   aws:elasticbeanstalk:environment:
     EnvironmentType: LoadBalanced
     ServiceRole: aws-elasticbeanstalk-service-role
   ```

3. Ändern Sie unter dem `aws:elasticbeanstalk:healthreporting:system:`-Namespace den Wert von `SystemType` zu **enhanced**.

```
  aws:elasticbeanstalk:healthreporting:system:
    SystemType: enhanced
```

4. Speichern Sie die Konfigurationsdatei und schließen Sie den Text-Editor.
5. Die EB-CLI startet ein Umgebungs-Update, um Ihre Konfigurationsänderungen anzuwenden. Warten Sie, bis der Vorgang abgeschlossen ist, oder drücken Sie Ctrl+C, um den Vorgang sicher zu beenden.

```
~/project$ eb config
Printing Status:
INFO: Environment update is starting.
INFO: Health reporting type changed to ENHANCED.
INFO: Updating environment no-role-test's configuration settings.
```

Aktivieren der erweiterten Zustandsberichte mit einer Konfigurationsdatei

Sie können erweiterte Zustandsberichte aktivieren, indem Sie eine Konfigurationsdatei zu Ihrem Quell-Bundle hinzufügen. Das folgende Beispiel zeigt eine Konfigurationsdatei, mit der erweiterte Zustandsberichte aktiviert und der Standardservice und das Standard-Instance-Profil zur Umgebung hinzugefügt werden:

Example.ebextensions/enhanced-health.config

```
option_settings:
  aws:elasticbeanstalk:healthreporting:system:
    SystemType: enhanced
  aws:autoscaling:launchconfiguration:
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
  aws:elasticbeanstalk:environment:
    ServiceRole: aws-elasticbeanstalk-service-role
```

Wenn Sie Ihr eigenes Instance-Profil oder Ihre eigene Servicerolle erstellt haben, ersetzen Sie den hervorgehobenen Text mit den Namen dieser Rollen.

Erweiterte Integritätsüberwachung mithilfe der Environment Management Console

Wenn Sie die erweiterte Statusberichterstattung in aktivieren AWS Elastic Beanstalk, können Sie den Zustand der Umgebung in der Umgebungsmanagementkonsole überwachen.

Themen

- Überblick über die Umgebung
- Seite „Umgebungsintegrität“
- Überwachungsseite

Überblick über die Umgebung

In der Umgebungsübersicht werden der Integritätsstatus der Umgebung angezeigt und Ereignisse aufgelistet, die Informationen zu aktuellen Änderungen in Bezug auf den Integritätsstatus bereitstellen.

So zeigen Sie die Umgebungsübersicht an

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.

Ausführliche Informationen zur aktuellen Integrität der Umgebung finden Sie auf der Seite Health (Integrität) durch Auswahl von Causes (Ursachen). Alternativ können Sie im Navigationsbereich die Option Health (Integrität) auswählen.

Seite „Umgebungsintegrität“

Auf der Seite Health werden der Gesundheitsstatus, Kennzahlen und Ursachen für die Umgebung und für jede EC2 Amazon-Instance in der Umgebung angezeigt.

> **Note**
>
> Elastic Beanstalk zeigt die Seite Health (Integrität) nur dann an, wenn Sie die erweiterte Integritätsüberwachung für die Umgebung aktiviert haben.

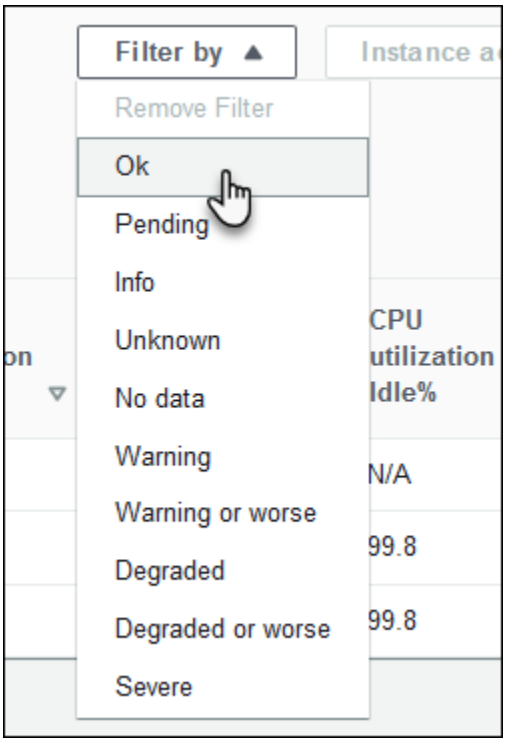
Das folgende Image zeigt die Seite Health (Integrität) in einer Linux-Umgebung.



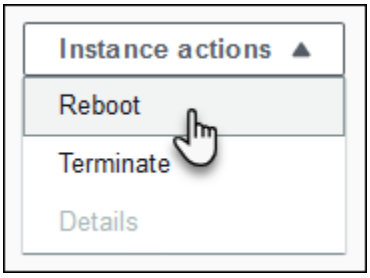
Das folgende Image zeigt die Seite Health (Integrität) für eine Windows-Umgebung. Beachten Sie, dass sich CPU-Metriken von denen in einer Linux-Umgebung unterscheiden.



Oben auf der Seite können Sie die Gesamtzahl der Umgebungs-Instances sowie die Anzahl der Instances pro Status anzeigen. Wenn Sie nur Instances mit einem bestimmten Status anzeigen möchten, wählen Sie Filter By (Filtern nach) und dann einen Status aus.



Um eine fehlerhafte Instance neu zu starten oder zu beenden, klicken Sie auf Instance Actions (Instance-Aktionen) und anschließend auf Reboot (Neu starten) oder Terminate (Beenden).



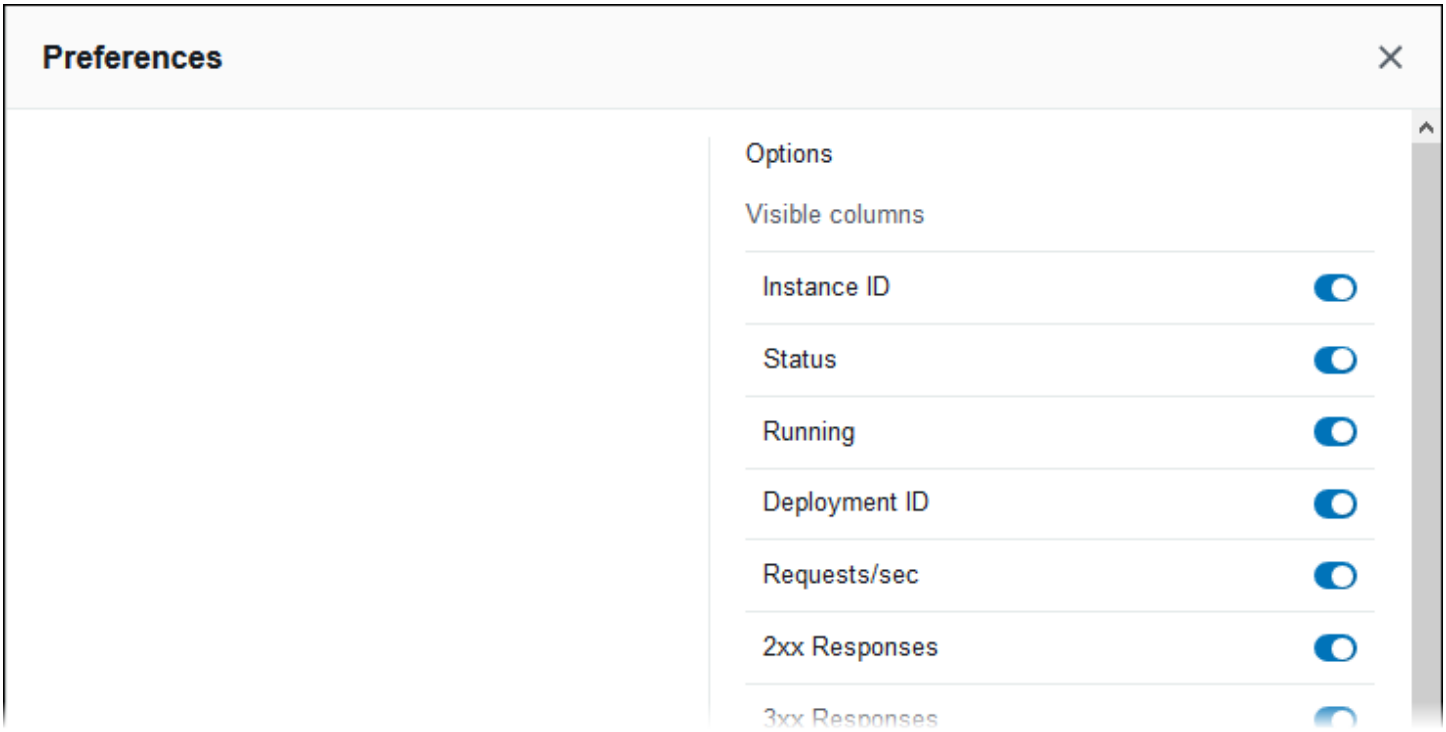
Elastic Beanstalk aktualisiert die Seite Health (Integrität) alle 10 Sekunden. Es werden Informationen zur Integrität von Umgebung und Instance angezeigt.

Für jede EC2 Amazon-Instance in der Umgebung werden auf der Seite die ID und der Status der Instance, die Zeit seit dem Start der Instance, die ID der letzten Bereitstellung, die auf der Instance ausgeführt wurde, die Antworten und die Latenz der Anfragen, die die Instance bedient hat, sowie Informationen zur Last und CPU-Auslastung angezeigt. In der Zeile Overall (Gesamt) werden Informationen zur durchschnittlichen Antwortzeit und Latenz für die gesamte Umgebung angezeigt.

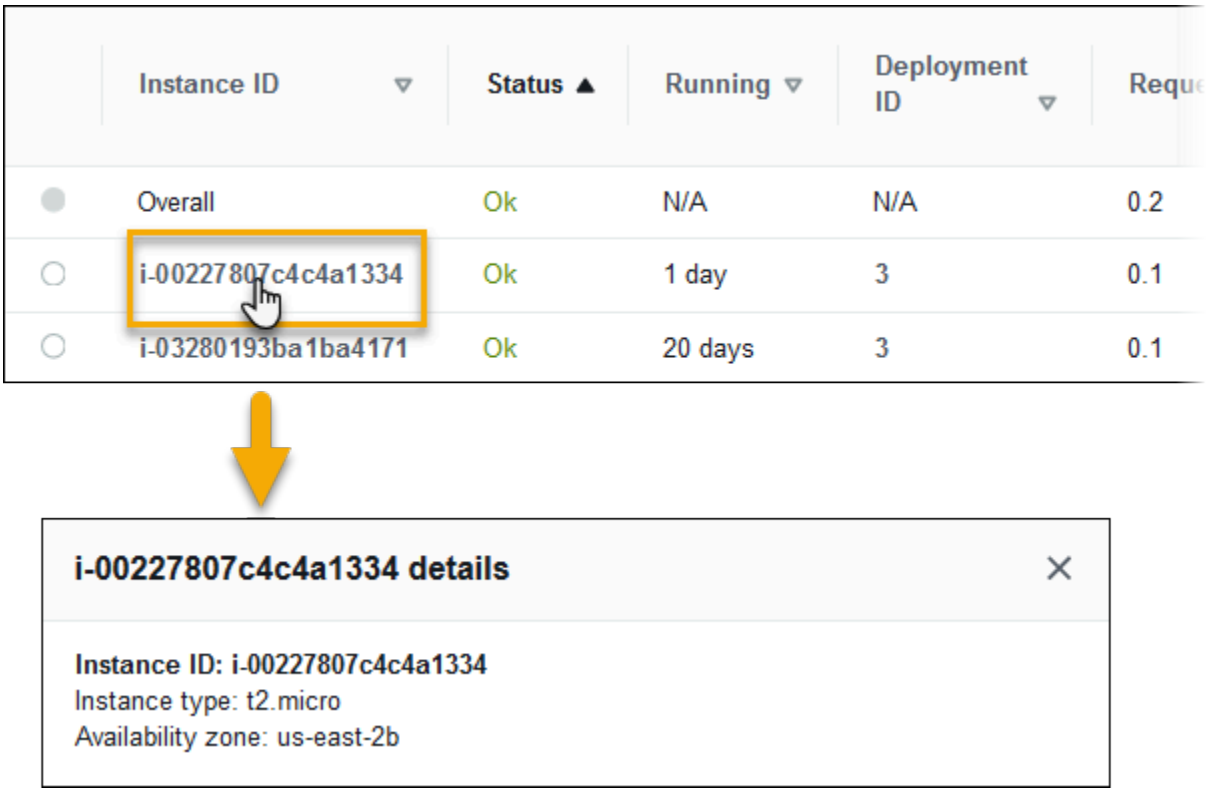
Die Seite zeigt zahlreiche Details in einer sehr breiten Tabelle an. Um Spalten auszublenden, wählen Sie



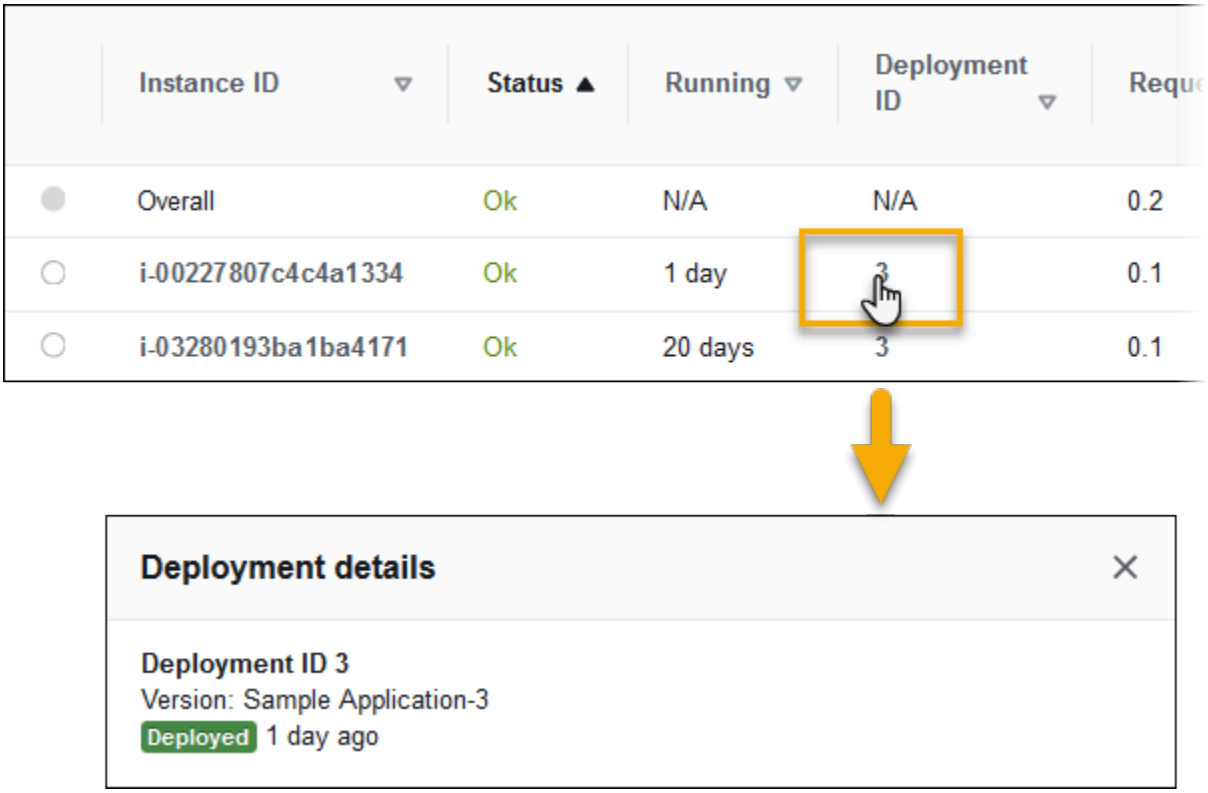
(Preferences (Einstellungen)) aus. Markieren oder löschen Sie Spaltennamen und klicken Sie anschließend auf Confirm (Bestätigen).



Wählen Sie die Instance ID (Instance-ID) einer Instance aus, um weitere Informationen zur Instance anzuzeigen, einschließlich Availability Zone und Instance-Typ.



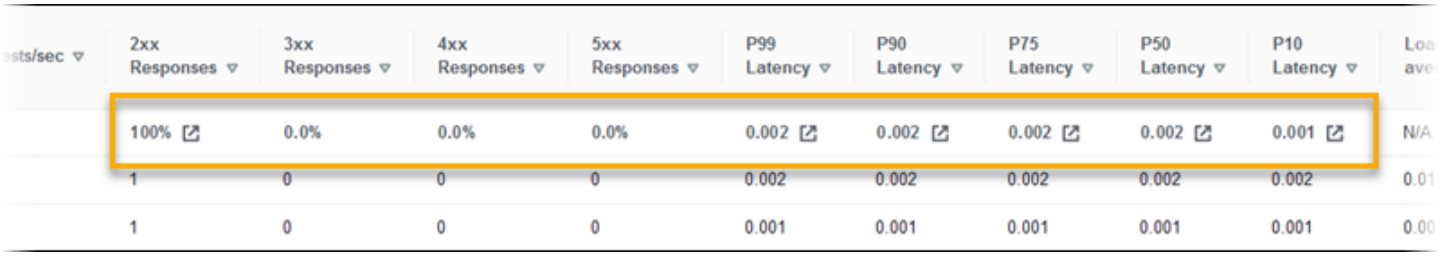
Wählen Sie die Deployment ID (Bereitstellungs-ID) einer Instance aus, um Informationen zur letzten Bereitstellung auf der Instance anzuzeigen.



Zu den Bereitstellungsinformationen gehören:

- Deployment ID (Bereitstellungs-ID) – Die eindeutige ID für die Bereitstellung. IDs Die Bereitstellung beginnt bei 1 und wird jedes Mal um eins erhöht, wenn Sie eine neue Anwendungsversion bereitstellen oder Konfigurationseinstellungen ändern, die sich auf die Software oder das Betriebssystem auswirken, die auf den Instances in Ihrer Umgebung ausgeführt werden.
- Version – Die Versionsbezeichnung des Anwendungsquellcodes, der in der Bereitstellung verwendet wird.
- Status – Der Status der Bereitstellung, der `In Progress`, `Deployed` oder `Failed` lauten kann.
- Time (Zeit) – Für angefangene Bereitstellungen ist dies die Zeit, zu der die Bereitstellung gestartet wurde. Für abgeschlossene Bereitstellungen ist dies die Zeit, zu der die Bereitstellung beendet wurde.

Wenn Sie die X-Ray-Integration in Ihrer Umgebung aktivieren und Ihre Anwendung mit dem AWS X-Ray SDK ausstatten, fügt die Health-Seite Links zur AWS X-Ray Konsole in der Übersichtszeile hinzu.



sts/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency	P10 Latency	Lo... ave...
	100%	0.0%	0.0%	0.0%	0.002	0.002	0.002	0.002	0.001	N/A
	1	0	0	0	0.002	0.002	0.002	0.002	0.002	0.01
	1	0	0	0	0.001	0.001	0.001	0.001	0.001	0.00

Wählen Sie einen Link, um Traces im Zusammenhang mit der markierten Statistik in der AWS X-Ray Konsole anzuzeigen.

Überwachungsseite

Auf der Seite Überwachung werden zusammenfassende Statistiken und Grafiken für die benutzerdefinierten CloudWatch Amazon-Metriken angezeigt, die vom erweiterten Gesundheitsberichtssystem generiert wurden. Unter Überwachung des Zustands der Umgebung in der AWS Managementkonsole finden Sie Anweisungen zum Hinzufügen von Diagrammen und Statistiken zu dieser Seite.

Farben und Status in Zustandsangaben

Erweiterte Zustandsberichte zeigen den Zustand von Instances und der Umgebung insgesamt mit vier Farben an, ähnlich wie bei grundlegenden Zustandsberichten. Erweitere Zustandsberichte bieten außerdem sieben Zustandsstatus, wobei es sich um einzelne Wortdeskriptoren handelt, die eine bessere Angabe des Zustands Ihrer Umgebung bieten.

Instance-Status und Umgebungsstatus

Jedes Mal, wenn Elastic Beanstalk eine Integritätsprüfung für Ihre Umgebung ausführt, wird mit den erweiterten Integritätsberichten die Integrität jeder Instance in Ihrer Umgebung durch die Analyse der verfügbaren Daten überprüft. Wenn eine der untergeordneten Prüfungen fehlschlägt, setzt Elastic Beanstalk die Integrität der Instance herunter.

Elastic Beanstalk zeigt die Integritätsinformationen für die gesamte Umgebung (Farbe, Status und Ursache) in der Environment Management Console an. Diese Informationen sind auch in der EB-CLI verfügbar. Status- und Ursachenmeldungen für einzelne Instances werden alle 10 Sekunden aktualisiert und sind in der EB-CLI verfügbar, wenn Sie den Status mit eb health anzeigen.

Elastic Beanstalk nutzt Änderungen bei der Instance-Integrität, um die Umgebungsintegrität zu evaluieren, ändert jedoch nicht sofort den Umgebungsintegritätsstatus. Wenn eine Instance die

Integritätsprüfungen mindestens drei Mal in einem 1-Minuten-Zeitraum nicht besteht, kann Elastic Beanstalk die Integrität der Umgebung herabsetzen. Abhängig von der Anzahl der Instances in der Umgebung und dem identifizierten Problem kann eine fehlerhafte Instance dazu führen, dass Elastic Beanstalk eine Informationsmeldung anzeigt oder den Integritätsstatus der Umgebung von Grün (OK) in Gelb (Warning (Warnung)) oder Rot (Degraded (Schwach)) oder (Severe (Stark)) ändert.

OK (Grün)

Dieser Status wird angezeigt, wenn:

- Eine Instance besteht die Zustandsprüfungen und der Zustandsagent meldet keine Probleme.
- Die meisten Instances in der Umgebung bestehen die Zustandsprüfungen und der Zustandsagent meldet keine wesentlichen Probleme.
- Eine Instance besteht Zustandsprüfungen und führt Anfragen normal aus.

Beispiel: Ihre Umgebung wurde kürzlich bereitgestellt und führt Anfragen normal aus. 5 % der Anfragen geben Serienfehler 400 zurück. Die Bereitstellung wurde auf jeder Instance normal abgeschlossen.

Nachricht (Instance): Anwendungsbereitstellung vor 23 Sekunden abgeschlossen (Dauer: 26 Sekunden).

Warning (Gelb)

Dieser Status wird angezeigt, wenn:

- Der Zustandsagent meldet eine geringe Anzahl an Anfragefehlern oder andere Probleme für eine Instance oder Umgebung.
- Ein Vorgang ist auf einer Instance in Bearbeitung und dauert sehr lange.

Beispiel: Eine Instance in der Umgebung hat den Status Severe (Stark).

Nachricht (Umgebung): Eingeschränkte Services bei 1 von 5 Instances.

Degraded (Rot)

Dieser Status wird angezeigt, wenn der Zustandsagent eine große Anzahl an Anfragefehlern oder andere Probleme für eine Instance oder Umgebung meldet.

Beispiel: Umgebung wird auf bis zu 5 Instance aufwärts skaliert.

Nachricht (Umgebung): 4 aktive Instances liegen unter der Mindestgröße für Auto Scaling-Gruppen von 5.

Severe (Rot)

Dieser Status wird angezeigt, wenn der Zustandsagent eine sehr große Anzahl an Anfragefehlern oder andere Probleme für eine Instance oder Umgebung meldet.

Beispiel: Elastic Beanstalk kann den Load Balancer nicht zum Abrufen der Instance-Integrität kontaktieren.

Nachricht (Umgebung): ELB-Zustandsprüfung schlägt fehl oder ist nicht für alle Instances verfügbar. Keine der Instances sendet Daten. Die Rolle „arn:aws:iam: :123456789012:role/“ konnte nicht übernommen werden. aws-elasticbeanstalk-service-role Überprüfen Sie, ob die Rolle vorhanden ist und richtig konfiguriert wurde.

Nachricht (Instances): Instance-ELB-Zustandsprüfung war 37 Minuten nicht verfügbar. Keine Daten. Zuletzt vor 37 Minuten gesehen.

Info (Grün)

Dieser Status wird angezeigt, wenn:

- Eine Operation wird auf einer Instance ausgeführt.
- Eine Operation wird auf mehreren Instances in einer Umgebung ausgeführt.

Beispiel: Eine neue Anwendungsversion wird für laufende Instances bereitgestellt.

Nachricht (Umgebung): Befehl wird auf 3 von 5 Instances ausgeführt.

Nachricht (Instance): Durchführen der Anwendungsbereitstellung (drei Sekunden lang ausgeführt).

Pending (Grau)

Dieser Status wird angezeigt, wenn auf einer Instance innerhalb des Befehls-Timeout eine Operation ausgeführt wird.

Beispiel: Sie haben kürzlich die Umgebung erstellt und Instances werden per Bootstrapping bereitgestellt.

Nachricht: Durchführen der Initialisierung (12 Sekunden lang ausgeführt).

Unknown (Grau)

Dieser Status wird angezeigt, wenn Elastic Beanstalk und der Integritäts-Agent eine unzureichende Anzahl an Daten für eine Instance melden.

Beispiel: Keine Daten werden empfangen.

Suspended (Grau)

Dieser Status wird angezeigt, wenn Elastic Beanstalk die Überwachung der Integrität der Umgebung gestoppt hat. Die Umgebung funktioniert möglicherweise nicht korrekt. Einige gravierende Integritätsbedingungen bewirken bei längerer Dauer, dass Elastic Beanstalk die Umgebung in den Status Suspended (Gesperrt) versetzt.

Beispiel: Elastic Beanstalk kann nicht auf die Servicerolle der Umgebung zugreifen.

Beispiel: Die von Elastic Beanstalk für die Umgebung erstellte Auto-Scaling-Gruppe wurde gelöscht.

Nachricht: Der Umgebungszustand hat von OK zu Severe (Stark) gewechselt. Es sind keine Instances vorhanden. Die gewünschte Kapazität für die Auto-Scaling-Gruppe ist auf 1 gesetzt.

Instance-Metriken

Instance-Metriken bieten Informationen über den Zustand der Instances in Ihrer Umgebung. Der Elastic-Beanstalk-Integritäts-Agent wird auf jeder Instance ausgeführt. Er erfasst Metriken zu Instances und leitet diese an Elastic Beanstalk weiter. Dort erfolgt die Analyse der Metriken, um die Integrität der Instances in Ihrer Umgebung zu bestimmen.

Der Elastic-Beanstalk-Integritäts-Agent für die Instance erfasst Metriken zu Instances von Webservern und dem Betriebssystem. Um Informationen vom Webserver auf Linux-basierten Plattformen zu erhalten, liest und analysiert Elastic Beanstalk Web-Server-Protokolle. Auf der Windows Server-Plattform erhält Elastic Beanstalk diese Informationen direkt von dem IIS-Webserver. Webserver bieten Informationen zu eingehenden HTTP-Anforderungen, z. B. wie viele Anforderungen eingegangen sind, wie viele davon zu Fehlern führten und wie lange die Verarbeitung gedauert hat. Das Betriebssystem bietet Snapshot-Informationen zum Status der Instance-Ressourcen, wie z. B. CPU-Auslastung und Verteilung der aufgewendeten Zeit je Prozesstyp.

Der Integritäts-Agent erfasst Webserver- und Betriebssystemmetriken und leitet diese in Intervallen von 10 Sekunden an Elastic Beanstalk weiter. Elastic Beanstalk analysiert die Daten und verwendet

die Ergebnisse, um den Integritätsstatus für die einzelnen Instances und die Umgebung zu aktualisieren.

Themen

- Webserver-Metriken
- Betriebssystemmetriken
- Erfassung von Webserver-Metriken in IIS auf Windows Server

Webserver-Metriken

Auf Linux-basierten Plattformen liest der Elastic-Beanstalk-Integritäts-Agent die Webservermetriken aus den Protokollen aus, die vom Webcontainer oder Server, der die Anforderungen für alle Instances der Umgebung verarbeitet, generiert werden. Elastic-Beanstalk-Plattformen sind so konfiguriert, dass zwei Protokolle generiert werden, nämlich ein visuell lesbares und ein maschinenlesbares Format. Der Integritäts-Agent leitet die maschinenlesbaren Protokolle alle 10 Sekunden an Elastic Beanstalk weiter.

Weitere Informationen zum Protokollformat von Elastic Beanstalk finden Sie unter Format der Protokolle der erweiterten Zustandsberichte.

Auf der Windows-Server-Plattform fügt Elastic Beanstalk ein Modul zur Anfrage-Pipeline des IIS-Webservers hinzu und erfasst Metriken über HTTP-Anfragezeiten und Antwortcodes. Das Modul sendet diese Metriken über einen leistungsfähigen IPC-Kanal (Inter-Process Communication) an den On-Instance-Zustandsagenten. Weitere Informationen zur Implementierung finden Sie in Erfassung von Webserver-Metriken in IIS auf Windows Server.

Gemeldete Webservermetriken

`RequestCount`

Anzahl der vom Webserver pro Sekunde verarbeiteten Anforderungen in den letzten zehn Sekunden. Dargestellt als Durchschnittswert `r/sec` (Anfragen pro Sekunde) in der EB-CLI und der Seite „Umgebungsintegrität“.

`Status2xx`, `Status3xx`, `Status4xx`, `Status5xx`

Anzahl der Anforderungen je Statuscode in den letzten zehn Sekunden. Beispielsweise geben erfolgreiche Anforderungen den Code "200 OK" und Weiterleitungen den Code "301" zurück.

Falls die eingegebene URL mit keiner Ressource in der Anwendung übereinstimmt, wird der Code "404" zurückgegeben.

In der EB-CLI und der Seite „Umgebungsintegrität“ werden diese Metriken sowohl als reine Anzahl der Anforderungen für Instances als auch als Prozentsatz der generellen Anforderungen für die Umgebungen dargestellt.

`p99.9`, `p99`, `p95`, `p90`, `p85`, `p75`, `p50`, `p10`

Durchschnittliche Latenzzeit der am langsamsten verarbeiteten x Prozent an Anforderungen in den letzten zehn Sekunden (wobei x die Differenz zwischen der Ziffer und 100 angibt). Beispielsweise gibt `p99 1.403` gibt an, dass für die am langsamsten verarbeiteten 1 Prozent der Anforderungen in den letzten zehn Sekunden die durchschnittliche Latenz 1,403 Sekunden betrug.

Betriebssystemmetriken

Der Elastic-Beanstalk-Integritäts-Agent meldet die folgenden Betriebssystemmetriken. Elastic Beanstalk verwendet diese Metriken, um Instances zu ermitteln, bei denen die Auslastung durchgängig hoch ist. Die Metriken unterscheiden sich je nach Betriebssystem.

Gemeldete Betriebssystemmetriken – Linux

`Running`

Der seit dem Instance-Start verstrichene Zeitraum.

`Load 1`, `Load 5`

Durchschnittliche Auslastung in den letzten Ein-Minuten- und Fünf-Minuten-Zeiträumen. Dieser Dezimalwert gibt die durchschnittliche Anzahl der in diesem Zeitraum ausgeführten Prozesse an. Wenn die angezeigte Zahl höher ist als die Anzahl der verfügbaren v CPUs (Threads), entspricht der Rest der durchschnittlichen Anzahl der Prozesse, die gewartet haben.

Wenn Ihr Instance-Typ beispielsweise vier V CPUs hat und die Last 4,5 beträgt, warteten in diesem Zeitraum durchschnittlich 0,5 Prozesse, was einem Prozess entspricht, der 50 Prozent der Zeit gewartet hat.

`User %`, `Nice %`, `System %`, `Idle %`, `I/O Wait %`

Prozentangabe des Zeitraums, den die CPU in den letzten zehn Sekunden im jeweiligen Status gewesen ist.

Gemeldete Betriebssystemmetriken – Windows

`Running`

Der seit dem Instance-Start verstrichene Zeitraum.

`% User Time`, `% Privileged Time`, `% Idle Time`

Prozentangabe des Zeitraums, den die CPU in den letzten zehn Sekunden im jeweiligen Status gewesen ist.

Erfassung von Webserver-Metriken in IIS auf Windows Server

Auf der Windows-Server-Plattform fügt Elastic Beanstalk ein Modul zur Anfrage-Pipeline des IIS-Webservers hinzu und erfasst Metriken über HTTP-Anfragezeiten und Antwortcodes. Das Modul sendet diese Metriken über einen leistungsfähigen IPC-Kanal (Inter-Process Communication) an den On-Instance-Zustandsagenten. Der Integritäts-Agent aggregiert diese Metriken, kombiniert sie mit Metriken des Betriebssystems und sendet sie an den Elastic-Beanstalk-Service.

Implementierungsinformationen

Um Metriken aus IIS zu erfassen, implementiert Elastic Beanstalk eine verwaltete `IHttpModule` und abonniert sie zu den `BeginRequest`- und `EndRequest`-Ereignissen. Dies ermöglicht es dem Modul, HTTP-Anfragelatenz und Antwortcodes für alle vom IIS bearbeiteten Webanfragen zu melden. Um das Modul der IIS-Anfrage-Pipeline hinzuzufügen, registriert Elastic Beanstalk das Modul im `<modules>`-Abschnitt der IIS-Konfigurationsdatei `%windir%\System32\inetsrv\config\applicationHost.config`.

Das Elastic-Beanstalk-Modul in IIS sendet die erfassten Metriken der Web-Anfrage an den Integritäts-Agenten für die Instance. Dabei handelt es sich um einen Windows-Service mit dem Namen `HealthD`. Um diese Daten zu senden, verwendet das Modul `NetNamedPipeBinding`, die eine sichere und zuverlässige Bindung bietet, die für die Kommunikation auf der Maschine optimiert ist.

Konfigurieren von Regeln für den erweiterten Zustand einer Umgebung

AWS Elastic Beanstalk Die erweiterte Zustandsberichterstattung stützt sich auf eine Reihe von Regeln zur Bestimmung des Zustands Ihrer Umgebung. Einige dieser Regeln sind möglicherweise nicht für Ihre jeweilige Anwendung geeignet. Im Folgenden finden Sie einige häufig verwendete Beispiele:

- Sie verwenden clientseitige Testwerkzeuge. In diesem Fall werden häufige HTTP-Clientfehler (4xx) erwartet.
- Sie verwenden AWS WAF in Verbindung mit dem Application Load Balancer Ihrer Umgebung, um unerwünschten eingehenden Datenverkehr zu blockieren. In diesem Fall gibt der Application Load Balancer für jede abgelehnte eingehende Nachricht „HTTP 403“ zurück.

Standardmäßig berücksichtigt Elastic Beanstalk alle HTTP 4xx-Fehler von Anwendungen, wenn die Integrität der Umgebung ermittelt wird. Abhängig von der Fehlerquote ändert sich der Zustandsstatus der Umgebung von OK in Warnung, Beeinträchtigt oder Schwerwiegend. Damit Fälle, wie die erwähnten Beispiele, korrekt gehandhabt werden, können Sie mit Elastic Beanstalk einige Regeln für die erweiterte Integrität konfigurieren. Sie können wählen, ob HTTP 4xx-Fehler von Anwendungen auf Instances der Umgebung oder ob HTTP 4xx-Fehler, die vom Load Balancer der Umgebung zurückgegeben werden, ignoriert werden sollen. In diesem Thema wird beschrieben, wie diese Konfigurationsänderungen vorgenommen werden.

> **Note**
>
> Derzeit sind dies die einzigen verfügbaren Anpassungen von Regeln zum erweiterten Zustand. Die erweiterte Integrität kann nicht so konfigurieren werden, dass andere HTTP-Fehler zusätzlich zu 4xx ignoriert werden.

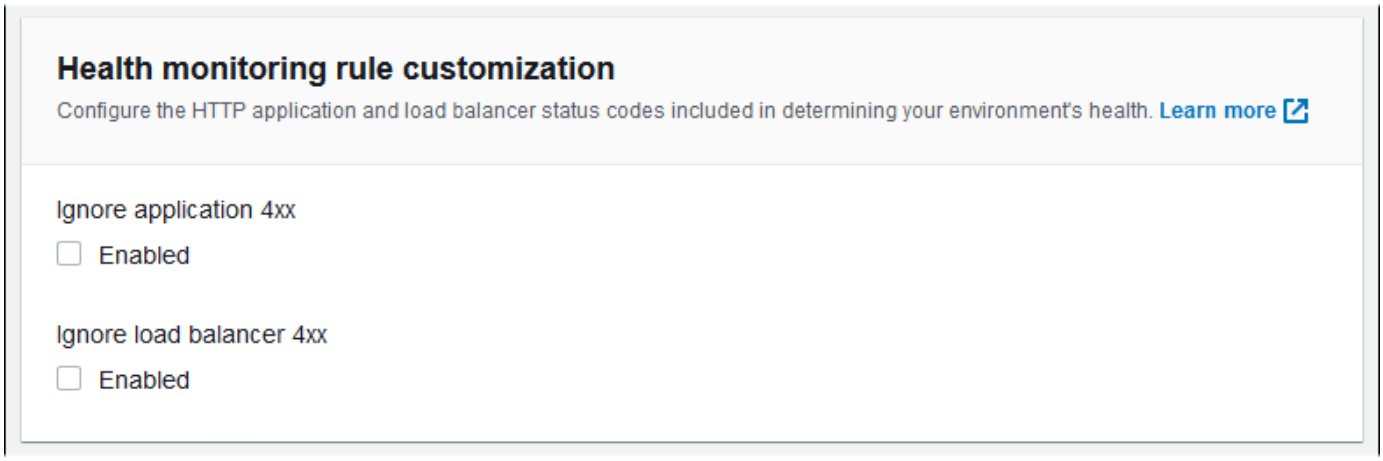
Konfigurieren von Regeln für die erweiterte Integrität mit der Elastic-Beanstalk-Konsole

Sie können die Elastic-Beanstalk-Konsole verwenden, um Regeln für die erweiterte Integrität in einer Umgebung zu konfigurieren.

So konfigurieren Sie die HTTP 4xx-Statuscodeprüfung mithilfe der Elastic-Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).

5. Aktivieren oder deaktivieren Sie unter Anpassung der Integritätsüberwachungsregel die gewünschten Ignore (Ignorieren)-Optionen.



6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Konfigurieren von erweiterten Zustandsregeln unter Verwendung der EB-CLI

Sie können mit der EB-CLI Regeln für die erweiterte Integrität konfigurieren, indem Sie die Konfiguration Ihrer Umgebung lokal speichern, einen Eintrag hinzufügen, der Regeln für die erweiterte Integrität konfiguriert, und die Konfiguration dann in Elastic Beanstalk hochladen. Sie können die gespeicherte Konfiguration zu einer Umgebung während oder nach der Erstellung hinzufügen.

So konfigurieren Sie die HTTP 4xx-Statuscodeprüfung mit der EB-CLI und gespeicherten Konfigurationen

1. Initialisieren Sie Ihren Projektordner mit eb init.
2. Erstellen Sie eine Umgebung, indem Sie den eb create-Befehl ausführen.
3. Speichern Sie eine Konfigurationsvorlage lokal, indem Sie den eb config save-Befehl ausführen. Im folgenden Beispiel wird die `--cfg`-Option verwendet, um den Namen der Konfiguration anzugeben.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-
state.cfg.yml
```

4. Öffnen Sie die gespeicherte Konfigurationsdatei in einem Texteditor.

5. Fügen Sie unter `OptionSettings` > `aws:elasticbeanstalk:healthreporting:system:` einen `ConfigDocument`-Schlüssel hinzu, um alle Regeln für den erweiterten Zustand aufzulisten, die konfiguriert werden können. Im Folgenden wird mit `ConfigDocument` die Überprüfung der HTTP 4xx-Statuscodes von Anwendungen deaktiviert, während die Überprüfung des HTTP 4xx-Codes des Load Balancer aktiviert bleibt.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      Rules:
        Environment:
          Application:
            ApplicationRequests4xx:
              Enabled: false
          ELB:
            ELBRequests4xx:
              Enabled: true
      Version: 1
    SystemType: enhanced
...
```

> **Note**
>
> Sie können `Rules` und `CloudWatchMetrics` in derselben `ConfigDocument`-Optionseinstellung kombinieren. `CloudWatchMetrics` sind in Veröffentlichen CloudWatch benutzerdefinierter Amazon-Metriken für eine Umgebung beschrieben. Wenn Sie zuvor `CloudWatchMetrics` aktiviert haben, hat die Konfigurationsdatei, die Sie mit dem Befehl eb config save abrufen, bereits einen `ConfigDocument`-Schlüssel mit einem `CloudWatchMetrics`-Abschnitt. Löschen Sie ihn nicht – fügen Sie einen `Rules`-Abschnitt in denselben `ConfigDocument`-Optionswert ein.

6. Speichern Sie die Konfigurationsdatei und schließen Sie den Text-Editor. Bei diesem Beispiel wird die aktualisierte Konfigurationsdatei mit einem Namen (`02-cloudwatch-enabled.cfg.yml`) gespeichert, der sich von dem der heruntergeladenen Konfigurationsdatei unterscheidet. Dadurch wird eine separat gespeicherte Konfiguration erstellt, wenn die Datei hochgeladen wird. Sie können demselben Namen wie die heruntergeladene Datei verwenden, um die vorhandene Konfiguration zu überschreiben, ohne dass Sie eine neue erstellen müssen.

7. Verwenden Sie zum Hochladen der aktualisierten Konfigurationsdatei in Elastic Beanstalk den Befehl eb config put.

   ```
   $ eb config put 02-cloudwatch-enabled
   ```

   Schließen Sie die Dateinamenerweiterung bei Verwendung der Befehle eb config `get` und `put` mit gespeicherten Konfigurationen nicht mit ein.
8. Wenden Sie die gespeicherte Konfiguration auf Ihre laufende Umgebung an.

   ```
   $ eb config --cfg 02-cloudwatch-enabled
   ```

   Die `--cfg`-Option gibt eine benannte Konfigurationsdatei an, die auf die Umgebung angewendet wird. Sie können die Konfigurationsdatei lokal oder in Elastic Beanstalk speichern. Wenn eine Konfigurationsdatei mit dem angegebenen Namen in beiden Speicherorten vorhanden ist, verwendet die EB-CLI die lokale Datei.

Konfigurieren von erweiterten Zustandsregeln unter Verwendung eines Konfigurationsdokuments

Das Konfigurationsdokument (config) für Regeln für erweiterte Zustände ist ein JSON-Dokument, das die zu konfigurierenden Regeln auflistet.

Das folgende Beispiel zeigt ein Konfigurationsdokument, das die Überprüfung von HTTP 4xx-Statuscodes von Anwendungen deaktiviert und die Überprüfung von HTTP 4xx-Statuscodes für Load Balancer aktiviert.

```
{
  "Rules": {
    "Environment": {
      "Application": {
        "ApplicationRequests4xx": {
          "Enabled": false
        }
      },
      "ELB": {
        "ELBRequests4xx": {
          "Enabled": true
        }
```

```
      }
    }
  },
  "Version": 1
}
```

Für den AWS CLIübergeben Sie das Dokument als Wert für den `Value` Schlüssel in einem Optionseinstellungsargument, das selbst ein JSON-Objekt ist. In diesem Fall müssen Anführungszeichen im eingebetteten Dokument durch Escape-Zeichen geschützt werden. Mit dem folgenden Befehl wird überprüft, ob die Konfigurationseinstellungen gültig sind.

```
$ aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings '[
    {
        "Namespace": "aws:elasticbeanstalk:healthreporting:system",
        "OptionName": "ConfigDocument",
        "Value": "{\"Rules\": { \"Environment\": { \"Application\":
 { \"ApplicationRequests4xx\": { \"Enabled\": false } }, \"ELB\": { \"ELBRequests4xx\":
 {\"Enabled\": true } } } }, \"Version\": 1 }"
    }
]'
```

Für eine `.ebextensions`-Konfigurationsdatei in YAML können Sie das JSON-Dokument unverändert bereitstellen.

```
  option_settings:
    - namespace: aws:elasticbeanstalk:healthreporting:system
      option_name: ConfigDocument
      value: {
  "Rules": {
    "Environment": {
      "Application": {
        "ApplicationRequests4xx": {
          "Enabled": false
        }
      },
      "ELB": {
        "ELBRequests4xx": {
          "Enabled": true
        }
      }
    }
```

```
  },
  "Version": 1
}
```

Veröffentlichen CloudWatch benutzerdefinierter Amazon-Metriken für eine Umgebung

Sie können die im Rahmen der AWS Elastic Beanstalk erweiterten Gesundheitsberichterstattung gesammelten Daten CloudWatch als benutzerdefinierte Metriken an Amazon veröffentlichen. Durch die Veröffentlichung von Metriken CloudWatch können Sie Veränderungen der Anwendungsleistung im Laufe der Zeit überwachen und potenzielle Probleme identifizieren, indem Sie verfolgen, wie die Ressourcennutzung und die Latenz von Anfragen mit der Auslastung skalieren.

Indem Sie Metriken veröffentlichen CloudWatch, stellen Sie sie auch für die Verwendung in Überwachungsdiagrammen und Alarmen zur Verfügung. Eine kostenlose Metrik, EnvironmentHealth, wird automatisch aktiviert, wenn Sie die erweiterte Statusberichterstattung verwenden. Für andere benutzerdefinierte Messwerte EnvironmentHealthfallen keine CloudWatch Standardgebühren an.

Um CloudWatch benutzerdefinierte Metriken für eine Umgebung zu veröffentlichen, müssen Sie zunächst die erweiterte Zustandsberichterstattung für die Umgebung aktivieren. Detaillierte Anweisungen finden Sie unter Aktivieren der erweiterten Elastic-Beanstalk-Integritätsberichte.

Themen

- Metriken der erweiterten Zustandsberichte
- Konfiguration von CloudWatch Metriken mit der Elastic Beanstalk Beanstalk-Konsole
- Konfiguration CloudWatch benutzerdefinierter Metriken mit der EB-CLI
- Bereitstellen von benutzerdefinierten Metrikkonfigurations-Dokumenten

Metriken der erweiterten Zustandsberichte

Wenn Sie die erweiterte Zustandsberichterstattung in Ihrer Umgebung aktivieren, veröffentlicht das erweiterte Zustandsberichtssystem automatisch eine CloudWatch benutzerdefinierte Metrik, EnvironmentHealth. Um zusätzliche Metriken zu veröffentlichen CloudWatch, konfigurieren Sie Ihre Umgebung mit diesen Metriken, indem Sie die Elastic Beanstalk Beanstalk-Konsole, EB CLI oder .ebextensions verwenden.

Sie können die folgenden erweiterten Integritätsmetriken aus Ihrer Umgebung in veröffentlichen. CloudWatch

Verfügbare Metriken – alle Plattformen

`EnvironmentHealth`

Nur Umgebung. Dies ist die einzige CloudWatch Metrik, die das erweiterte Zustandsberichtssystem veröffentlicht, sofern Sie keine zusätzlichen Messwerte konfigurieren. Der Umgebungszustand wird durch einen von sieben Status dargestellt. In der CloudWatch Konsole werden diese Status den folgenden Werten zugeordnet:

- 0 – OK
- 1 – Info
- 5 – Unknown
- 10 – No data
- 15 – Warning
- 20 – Degraded
- 25 – Severe

`InstancesSevere`, `InstancesDegraded`, `InstancesWarning`, `InstancesInfo`, `InstancesOk`, `InstancesPending`, `InstancesUnknown`, `InstancesNoData`

Nur Umgebung. Diese Metriken geben die Anzahl der Instances in der Umgebung mit dem jeweiligen Zustand an. `InstancesNoData` gibt die Anzahl der Instances an, für die keine Daten empfangen wurden.

`ApplicationRequestsTotal`, `ApplicationRequests5xx`, `ApplicationRequests4xx`, `ApplicationRequests3xx`, `ApplicationRequests2xx`

Instance und Umgebung. Gibt die Gesamtanzahl der Anforderungen an, die von der Instance oder Umgebung abgeschlossen wurden, und die Anzahl der Anfragen, die mit jeder Statuscodekategorie abgeschlossen wurde.

`ApplicationLatencyP10`, `ApplicationLatencyP50`, `ApplicationLatencyP75`, `ApplicationLatencyP85`, `ApplicationLatencyP90`, `ApplicationLatencyP95`, `ApplicationLatencyP99`, `ApplicationLatencyP99.9`

Instance und Umgebung. Gibt die durchschnittliche Zeit in Sekunden an, die es dauert, bis die schnellsten x Prozent der Anfragen abgeschlossen wurden.

`InstanceHealth`

Nur Instance. Gibt den aktuellen Zustand der Instance an. Der Instance-Zustand wird durch einen von sieben Status dargestellt. In der CloudWatch Konsole werden diese Status den folgenden Werten zugeordnet:

- 0 – OK
- 1 – Info
- 5 – Unknown
- 10 – No data
- 15 – Warning
- 20 – Degraded
- 25 – Severe

Verfügbaren Metriken – Linux

`CPUIrq`, `CPUIdle`, `CPUUser`, `CPUSystem`, `CPUSoftirq`, `CPUIowait`, `CPUNice`

Nur Instance. Gibt eine Prozentangabe des Zeitraums an, den die CPU in der letzten Minute im jeweiligen Status gewesen ist.

`LoadAverage1min`

Nur Instance. Die durchschnittliche CPU-Auslastung der Instance innerhalb der letzten Minute.

`RootFilesystemUtil`

Nur Instance. Gibt den prozentualen Anteil des verwendeten Speicherplatzes an.

Verfügbaren Metriken – Windows

`CPUIdle`, `CPUUser`, `CPUPrivileged`

Nur Instance. Gibt eine Prozentangabe des Zeitraums an, den die CPU in der letzten Minute im jeweiligen Status gewesen ist.

Konfiguration von CloudWatch Metriken mit der Elastic Beanstalk Beanstalk-Konsole

Sie können die Elastic Beanstalk Beanstalk-Konsole verwenden, um Ihre Umgebung so zu konfigurieren, dass erweiterte Gesundheitsberichtsmetriken veröffentlicht CloudWatch und für die Verwendung mit Überwachungsdiagrammen und Alarmen verfügbar gemacht werden.

So konfigurieren Sie CloudWatch benutzerdefinierte Metriken in der Elastic Beanstalk Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).
5. Wählen Sie unter Health reporting (Zustandsberichte) die Instance-und Umgebungsmetriken aus, die Sie in CloudWatch veröffentlichen möchten. Zur Auswahl mehrerer Metriken drücken Sie die Strg-Taste während der Auswahl.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Wenn Sie CloudWatch benutzerdefinierte Metriken aktivieren, werden sie der Liste der auf der Monitoring-Seite verfügbaren Metriken hinzugefügt.

Konfiguration CloudWatch benutzerdefinierter Metriken mit der EB-CLI

Sie können mit der EB-CLI benutzerdefinierte Metriken konfigurieren, indem Sie die Konfiguration Ihrer Umgebung lokal speichern, einen Eintrag hinzufügen, der die Metriken für die Veröffentlichung definiert, und die Konfiguration dann in Elastic Beanstalk hochladen. Sie können die gespeicherte Konfiguration zu einer Umgebung während oder nach der Erstellung hinzufügen.

So konfigurieren Sie CloudWatch benutzerdefinierte Metriken mit der EB-CLI und gespeicherten Konfigurationen

1. Initialisieren Sie Ihren Projektordner mit eb init.
2. Erstellen Sie eine Umgebung, indem Sie den eb create-Befehl ausführen.

3. Speichern Sie eine Konfigurationsvorlage lokal, indem Sie den eb config save-Befehl ausführen. Im folgenden Beispiel wird die `--cfg`-Option verwendet, um den Namen der Konfiguration anzugeben.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-
state.cfg.yml
```

4. Öffnen Sie die gespeicherte Konfigurationsdatei in einem Texteditor.

5. Fügen Sie unter `OptionSettings` > einen `ConfigDocument` Schlüssel hinzu`aws:elasticbeanstalk:healthreporting:system:`, um jede der gewünschten CloudWatch Metriken zu aktivieren. Wenn beispielsweise der folgende `ConfigDocument` `ApplicationRequests5xx`- und `ApplicationRequests4xx` -Metriken auf Umgebungsebene und `ApplicationRequestsTotal`-Metriken auf Instance-Ebene veröffentlicht.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      CloudWatchMetrics:
        Environment:
          ApplicationRequests5xx: 60
          ApplicationRequests4xx: 60
        Instance:
          ApplicationRequestsTotal: 60
      Version: 1
    SystemType: enhanced
...
```

Im Beispiel gibt 60 die Anzahl der Sekunden zwischen Messungen an. Derzeit ist dies der einzige unterstützte Wert.

> **Note**
>
> Sie können `CloudWatchMetrics` und `Rules` in derselben `ConfigDocument`-Optionseinstellung kombinieren. `Rules` sind in Konfigurieren von Regeln für den erweiterten Zustand einer Umgebung beschrieben.

> Wenn Sie zuvor `Rules` verwendet haben, um Regeln für den erweiterten Zustand zu konfigurieren, hat die Konfigurationsdatei, die Sie mit dem Befehl eb config save abrufen, bereits einen `ConfigDocument`-Schlüssel mit einem `Rules`-Abschnitt. Löschen Sie ihn nicht – fügen Sie einen `CloudWatchMetrics`-Abschnitt in denselben `ConfigDocument`-Optionswert ein.

6. Speichern Sie die Konfigurationsdatei und schließen Sie den Text-Editor. Bei diesem Beispiel wird die aktualisierte Konfigurationsdatei mit einem Namen (`02-cloudwatch-enabled.cfg.yml`) gespeichert, der sich von dem der heruntergeladenen Konfigurationsdatei unterscheidet. Dadurch wird eine separat gespeicherte Konfiguration erstellt, wenn die Datei hochgeladen wird. Sie können demselben Namen wie die heruntergeladene Datei verwenden, um die vorhandene Konfiguration zu überschreiben, ohne dass Sie eine neue erstellen müssen.
7. Verwenden Sie zum Hochladen der aktualisierten Konfigurationsdatei in Elastic Beanstalk den Befehl eb config put.

   ```
   $ eb config put 02-cloudwatch-enabled
   ```

   Schließen Sie die Dateierweiterung bei Verwendung der Befehle eb config `get` und `put` mit gespeicherten Konfigurationen nicht mit ein.
8. Wenden Sie die gespeicherte Konfiguration auf Ihre laufende Umgebung an.

   ```
   $ eb config --cfg 02-cloudwatch-enabled
   ```

   Die `--cfg`-Option gibt eine benannte Konfigurationsdatei an, die auf die Umgebung angewendet wird. Sie können die Konfigurationsdatei lokal oder in Elastic Beanstalk speichern. Wenn eine Konfigurationsdatei mit dem angegebenen Namen in beiden Speicherorten vorhanden ist, verwendet die EB-CLI die lokale Datei.

Bereitstellen von benutzerdefinierten Metrikkonfigurations-Dokumenten

Das Konfigurationsdokument (Config) für CloudWatch benutzerdefinierte Amazon-Metriken ist ein JSON-Dokument, das die zu veröffentlichenden Metriken auf Umgebungs- und Instanzebene auflistet. Das folgende Beispiel zeigt ein Konfigurationsdokument, das alle verfügbaren benutzerdefinierten Metriken aktiviert.

```
{
```

```
  "CloudWatchMetrics": {
    "Environment": {
      "ApplicationLatencyP99.9": 60,
      "InstancesSevere": 60,
      "ApplicationLatencyP90": 60,
      "ApplicationLatencyP99": 60,
      "ApplicationLatencyP95": 60,
      "InstancesUnknown": 60,
      "ApplicationLatencyP85": 60,
      "InstancesInfo": 60,
      "ApplicationRequests2xx": 60,
      "InstancesDegraded": 60,
      "InstancesWarning": 60,
      "ApplicationLatencyP50": 60,
      "ApplicationRequestsTotal": 60,
      "InstancesNoData": 60,
      "InstancesPending": 60,
      "ApplicationLatencyP10": 60,
      "ApplicationRequests5xx": 60,
      "ApplicationLatencyP75": 60,
      "InstancesOk": 60,
      "ApplicationRequests3xx": 60,
      "ApplicationRequests4xx": 60
    },
    "Instance": {
      "ApplicationLatencyP99.9": 60,
      "ApplicationLatencyP90": 60,
      "ApplicationLatencyP99": 60,
      "ApplicationLatencyP95": 60,
      "ApplicationLatencyP85": 60,
      "CPUUser": 60,
      "ApplicationRequests2xx": 60,
      "CPUIdle": 60,
      "ApplicationLatencyP50": 60,
      "ApplicationRequestsTotal": 60,
      "RootFilesystemUtil": 60,
      "LoadAverage1min": 60,
      "CPUIrq": 60,
      "CPUNice": 60,
      "CPUIowait": 60,
      "ApplicationLatencyP10": 60,
      "LoadAverage5min": 60,
      "ApplicationRequests5xx": 60,
      "ApplicationLatencyP75": 60,
```

```
      "CPUSystem": 60,
      "ApplicationRequests3xx": 60,
      "ApplicationRequests4xx": 60,
      "InstanceHealth": 60,
      "CPUSoftirq": 60
    }
  },
  "Version": 1
}
```

Für das AWS CLIübergeben Sie das Dokument als Wert für den `Value` Schlüssel in einem Optionseinstellungsargument, das selbst ein JSON-Objekt ist. In diesem Fall müssen Anführungszeichen im eingebetteten Dokument durch Escape-Zeichen geschützt werden.

```
$ aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings '[
    {
        "Namespace": "aws:elasticbeanstalk:healthreporting:system",
        "OptionName": "ConfigDocument",
        "Value": "{\"CloudWatchMetrics\": {\"Environment\":
 {\"ApplicationLatencyP99.9\": 60,\"InstancesSevere\": 60,\"ApplicationLatencyP90\":
 60,\"ApplicationLatencyP99\": 60,\"ApplicationLatencyP95\": 60,\"InstancesUnknown
\": 60,\"ApplicationLatencyP85\": 60,\"InstancesInfo\": 60,\"ApplicationRequests2xx
\": 60,\"InstancesDegraded\": 60,\"InstancesWarning\": 60,\"ApplicationLatencyP50\":
 60,\"ApplicationRequestsTotal\": 60,\"InstancesNoData\": 60,\"InstancesPending
\": 60,\"ApplicationLatencyP10\": 60,\"ApplicationRequests5xx\": 60,
\"ApplicationLatencyP75\": 60,\"InstancesOk\": 60,\"ApplicationRequests3xx\": 60,
\"ApplicationRequests4xx\": 60},\"Instance\": {\"ApplicationLatencyP99.9\": 60,
\"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": 60,\"ApplicationLatencyP95\":
 60,\"ApplicationLatencyP85\": 60,\"CPUUser\": 60,\"ApplicationRequests2xx\":
 60,\"CPUIdle\": 60,\"ApplicationLatencyP50\": 60,\"ApplicationRequestsTotal\":
 60,\"RootFilesystemUtil\": 60,\"LoadAverage1min\": 60,\"CPUIrq\": 60,\"CPUNice
\": 60,\"CPUIowait\": 60,\"ApplicationLatencyP10\": 60,\"LoadAverage5min\": 60,
\"ApplicationRequests5xx\": 60,\"ApplicationLatencyP75\": 60,\"CPUSystem\": 60,
\"ApplicationRequests3xx\": 60,\"ApplicationRequests4xx\": 60,\"InstanceHealth\": 60,
\"CPUSoftirq\": 60}},\"Version\": 1}"
    }
]'
```

Für eine `.ebextensions`-Konfigurationsdatei in YAML können Sie das JSON-Dokument unverändert bereitstellen.

```
option_settings:
  - namespace: aws:elasticbeanstalk:healthreporting:system
    option_name: ConfigDocument
    value: {
"CloudWatchMetrics": {
  "Environment": {
    "ApplicationLatencyP99.9": 60,
    "InstancesSevere": 60,
    "ApplicationLatencyP90": 60,
    "ApplicationLatencyP99": 60,
    "ApplicationLatencyP95": 60,
    "InstancesUnknown": 60,
    "ApplicationLatencyP85": 60,
    "InstancesInfo": 60,
    "ApplicationRequests2xx": 60,
    "InstancesDegraded": 60,
    "InstancesWarning": 60,
    "ApplicationLatencyP50": 60,
    "ApplicationRequestsTotal": 60,
    "InstancesNoData": 60,
    "InstancesPending": 60,
    "ApplicationLatencyP10": 60,
    "ApplicationRequests5xx": 60,
    "ApplicationLatencyP75": 60,
    "InstancesOk": 60,
    "ApplicationRequests3xx": 60,
    "ApplicationRequests4xx": 60
  },
  "Instance": {
    "ApplicationLatencyP99.9": 60,
    "ApplicationLatencyP90": 60,
    "ApplicationLatencyP99": 60,
    "ApplicationLatencyP95": 60,
    "ApplicationLatencyP85": 60,
    "CPUUser": 60,
    "ApplicationRequests2xx": 60,
    "CPUIdle": 60,
    "ApplicationLatencyP50": 60,
    "ApplicationRequestsTotal": 60,
    "RootFilesystemUtil": 60,
    "LoadAverage1min": 60,
    "CPUIrq": 60,
    "CPUNice": 60,
```

```
      "CPUIowait": 60,
      "ApplicationLatencyP10": 60,
      "LoadAverage5min": 60,
      "ApplicationRequests5xx": 60,
      "ApplicationLatencyP75": 60,
      "CPUSystem": 60,
      "ApplicationRequests3xx": 60,
      "ApplicationRequests4xx": 60,
      "InstanceHealth": 60,
      "CPUSoftirq": 60
    }
  },
  "Version": 1
}
```

Verwenden der erweiterten Integritätsberichte mit der Elastic-Beanstalk-API

Da AWS Elastic Beanstalk erweiterte Integritätsberichte Rollen- und Lösungs-Stack-Anforderungen haben, müssen Sie Skripts und Code, die Sie vor der Veröffentlichung von Enhanced Health Reporting verwendet haben, aktualisieren, bevor Sie sie verwenden können. Um die Abwärtskompatibilität zu gewährleisten, ist bei einer Umgebungserstellung mithilfe der Elastic-Beanstalk-API die Funktion der erweiterten Integritätsberichte standardmäßig deaktiviert.

Sie konfigurieren die erweiterte Statusberichterstattung, indem Sie die Servicerolle, das Instance-Profil und die CloudWatch Amazon-Konfigurationsoptionen für Ihre Umgebung festlegen. Dafür gibt es drei Möglichkeiten: Sie legen die Konfigurationsoptionen im Ordner `.ebextensions` fest, Sie nutzen gespeicherte Konfigurationen oder Sie nehmen die Konfiguration direkt im Parameter `create-environment` des Aufrufs `option-settings` vor.

Um mit der API oder der AWS Befehlszeilenschnittstelle (CLI) eine Umgebung zu erstellen, die Enhanced Health unterstützt, müssen Sie: SDKs

- Erstellen Sie eine Servicerolle und ein Instance-Profil mit den entsprechenden Berechtigungen.
- Erstellen Sie eine neue Umgebung mit einer neuen Plattformversion.
- Legen Sie die Konfigurationsoptionen für Zustandssystemtyp, Instance-Profil und Servicerolle fest.

Verwenden Sie die folgenden Konfigurationsoptionen in den Namespaces `aws:elasticbeanstalk:healthreporting:system`,

`aws:autoscaling:launchconfiguration` und `aws:elasticbeanstalk:environment`, um die erweiterten Zustandsberichte in der Umgebung zu konfigurieren.

Konfigurationsoptionen für erweiterte Zustandsberichte

SystemType

Namespace: `aws:elasticbeanstalk:healthreporting:system`

Legen Sie den Wert auf fest, um die Funktion der erweiterten Zustandsberichte zu nutzen **enhanced**.

IamInstanceProfile

Namespace: `aws:autoscaling:launchconfiguration`

Geben Sie den Namen eines Instance-Profils an, das für die Verwendung mit Elastic Beanstalk konfiguriert ist.

ServiceRole

Namespace: `aws:elasticbeanstalk:environment`

Geben Sie den Namen einer Servicerolle an, die für die Verwendung mit Elastic Beanstalk konfiguriert ist.

ConfigDocument (optional)

Namespace: `aws:elasticbeanstalk:healthreporting:system`

Ein JSON-Dokument, das die Instanz- und Umgebungsmetriken definiert, in denen veröffentlicht CloudWatch werden soll. Beispiel:

```
{
  "CloudWatchMetrics":
    {
    "Environment":
      {
      "ApplicationLatencyP99.9":60,
      "InstancesSevere":60
      }
    "Instance":
      {
      "ApplicationLatencyP85":60,
```

```
      "CPUUser": 60
      }
    }
  "Version":1
}
```

> **Note**
>
> Je nach der Bereitstellung für Elastic Beanstalk erfordern Config-Dokumente möglicherweise eine spezielle Formatierung (z. B. Escape-Anführungszeichen). Beispiele finden Sie unter Bereitstellen von benutzerdefinierten Metrikkonfigurations-Dokumenten.

Format der Protokolle der erweiterten Zustandsberichte

AWS Elastic Beanstalk Plattformen verwenden ein benutzerdefiniertes Webserver-Protokollformat, um Informationen über HTTP-Anfragen effizient an das erweiterte Gesundheitsberichtssystem weiterzuleiten. Das System analysiert die Protokolle, identifiziert Probleme und legt den Instance- und Umgebungszustand entsprechend fest. Wenn Sie den Webserver-Proxy in Ihrer Umgebung deaktivieren und Anfragen direkt über den Webcontainer verarbeiten, können Sie erweiterte Integritätsberichte trotzdem vollumfänglich nutzen, indem Sie den Server für die Ausgabe von Protokollen in den Speicherort und im Format konfigurieren, den bzw. das der Elastic-Beanstalk-Integritäts-Agent verwendet.

> **Note**
>
> Die Informationen auf dieser Seite gilt nur für Linux-basierte Plattformen. Auf der Windows-Server-Plattform erhält Elastic Beanstalk diese Informationen über HTTP-Anfragen direkt von dem IIS-Webserver. Details hierzu finden Sie unter Erfassung von Webserver-Metriken in IIS auf Windows Server.

Konfiguration von Webserver-Protokollen

Elastic-Beanstalk-Plattformen sind so konfiguriert, dass sie zwei Protokolle mit Informationen über HTTP-Anfragen ausgeben. Die erste ist im Verbose-Format und bietet detaillierte Informationen über die Anfrage, einschließlich der Benutzer-Agent-Informationen des Anforderers und eines lesbaren Zeitstempels.

/var/log/nginx/access.log

Im folgenden Beispiel wird ein nginx-Proxy auf einer Ruby-Webserverumgebung ausgeführt, das Format ist jedoch vergleichbar für Apache.

```
172.31.24.3 - - [23/Jul/2015:00:21:20 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:21 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
```

Das zweite Protokoll ist im Terse-Format. Es enthält Informationen, die nur für erweiterte Zustandsberichte relevant sind. Dieses Protokoll wird in einen Unterordner mit dem Namen `healthd` ausgegeben und stündlich rotiert. Alte Protokolle werden sofort nach der Rotation gelöscht.

/.log.2015-07-23-00 var/log/nginx/healthd/application

Das folgende Beispiel zeigt ein Protokoll in einem für Computer lesbaren Format.

```
1437609879.311"/"200"0.083"0.083"177.72.242.17
1437609879.874"/"200"0.347"0.347"177.72.242.17
1437609880.006"/bad/path"404"0.001"0.001"177.72.242.17
1437609880.058"/"200"0.530"0.530"177.72.242.17
1437609880.928"/bad/path"404"0.001"0.001"177.72.242.17
```

Das Protokollformat für erweiterte Zustandsberichte enthält die folgenden Informationen:

- Die Uhrzeit der Anfrage in Unix-Zeit
- Den Pfad der Anfrage
- Den HTTP-Statuscode für das Ergebnis

- Die Anfragezeit
- Die Upstream-Zeit
- Den `X-Forwarded-For`-HTTP-Header

Für nginx-Proxys werden Zeiten in Floating-Point-Sekunden mit drei Dezimalstellen gedruckt. Für Apache werden ganze Mikrosekunden verwendet.

> **Note**
>
> Wenn Sie eine Warnung ähnlich der Folgenden in einer Protokolldatei sehen, in der Datum und Uhrzeit `DATE-TIME` ist, und Sie einen benutzerdefinierten Proxy verwenden, z. B. wie in einer Multicontainer-Docker-Umgebung, müssen Sie eine .ebextension zur Konfiguration Ihrer Umgebung verwenden, sodass die `healthd` Ihre Protokolldateien lesen kann:
>
> ```
> W, [DATE-TIME #1922] WARN -- : log file "/var/log/nginx/healthd/
> application.log.DATE-TIME" does not exist
> ```
>
> Sie können mit .ebextension im Beispiel für Multicontainer-Docker beginnen.

/etc/nginx/conf.d/webapp_healthd.conf

Das folgende Beispiel zeigt die Protokollkonfiguration für nginx mit dem `healthd`-Protokollformat hervorgehoben.

```
upstream my_app {
  server unix:///var/run/puma/my_app.sock;
}

log_format healthd '$msec"$uri"'
                '$status"$request_time"$upstream_response_time"'
                '$http_x_forwarded_for';

server {
  listen 80;
  server_name _ localhost; # need to listen to localhost for worker tier

  if ($time_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})T(\d{2})") {
    set $year $1;
    set $month $2;
```

```
    set $day $3;
    set $hour $4;
  }

  access_log  /var/log/nginx/access.log  main;
  access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour healthd;

  location / {
    proxy_pass http://my_app; # match the name of upstream directive which is defined
 above
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
  }

  location /assets {
    alias /var/app/current/public/assets;
    gzip_static on;
    gzip on;
    expires max;
    add_header Cache-Control public;
  }

  location /public {
    alias /var/app/current/public;
    gzip_static on;
    gzip on;
    expires max;
    add_header Cache-Control public;
  }
}
```

etc/httpd/conf.d/healthd/.conf

Im folgenden Beispiel wird die Protokollkonfiguration für Apache gezeigt.

```
LogFormat "%{%s}t\"%U\"%s\"%D\"%D\"%{X-Forwarded-For}i" healthd
CustomLog "|/usr/sbin/rotatelogs /var/log/httpd/healthd/application.log.%Y-%m-%d-%H
 3600" healthd
```

Generieren von Protokollen für erweiterte Zustandsberichte

Um Protokolle für den Zustandsagenten bereitzustellen, müssen Sie die folgenden Schritte ausführen:

- Protokolle im korrekten Format ausgeben, wie im vorherigen Abschnitt gezeigt
- Protokolle in Ausgabe `/var/log/nginx/healthd/`
- Protokolle mit folgendem Format benennen: `application.log.$year-$month-$day-$hour`
- Protokolle einmal pro Stunde rotieren
- Protokolle nicht kürzen

Benachrichtigungen und Fehlerbehebung

> **Testen Sie Amazon Q Developer CLI zur KI-gestützten Fehlerbehebung**
>
> Amazon Q Developer CLI kann Ihnen helfen, Umgebungsprobleme schnell zu beheben. Die Q CLI bietet Lösungen, indem sie den Umgebungsstatus überprüft, Ereignisse überprüft, Protokolle analysiert und klärende Fragen stellt. Weitere Informationen und detaillierte Anleitungen finden Sie in den Blogs unter Troubleshooting Elastic Beanstalk Environments with Amazon Q Developer CLI. AWS

Auf dieser Seite sind Meldungen zu häufig auftretenden Problemen sowie Links zu weiteren Informationen aufgeführt. Meldungen werden in Bereich „Übersicht über die Umgebung“ der Elastic Beanstalk Beanstalk-Konsole angezeigt und in Ereignissen aufgezeichnet, bei denen Gesundheitsprobleme über mehrere Prüfungen hinweg andauern.

Bereitstellungen

Elastic Beanstalk überwacht Ihre Umgebung nach Bereitstellungen auf Konsistenz. Wenn eine fortlaufende Bereitstellung fehlschlägt, kann die Version Ihrer Anwendung auf den Instances in Ihrer Umgebung variieren. Dies kann der Fall sein, wenn eine Bereitstellung auf einem oder mehreren Stapeln erfolgreich ist, aber vor allen abgeschlossenen Stapeln fehlschlägt.

In 2 von 5 Instanzen wurde eine falsche Anwendungsversion gefunden. Erwartete Version „v1" (Bereitstellung 1).

Falsche Anwendungsversion auf Umgebungsinstanzen. Erwartete Version „v1" (Bereitstellung 1).

Die erwartete Anwendungsversion wird auf einigen oder allen Instances in einer Umgebung nicht ausgeführt.

Falsche Anwendungsversion „v2" (Bereitstellung 2). Erwartete Version „v1" (Bereitstellung 1).

Die Anwendung, die auf einer Instance bereitgestellt wurde, unterscheidet sich von der erwarteten Version. Wenn eine Bereitstellung fehlschlägt, wird die erwartete Version auf die Version aus der letzten erfolgreichen Bereitstellung zurückgesetzt. Im vorangegangenen Beispiel war die erste Bereitstellung (Version "v1") erfolgreich, aber die zweite Bereitstellung (Version "v2") ist fehlgeschlagen. Alle Instances mit "v2" werden als fehlerhaft angesehen.

Um dieses Problem zu lösen, starten Sie eine andere Bereitstellung. Sie können eine frühere Version erneut bereitstellen, von der Sie wissen, dass sie funktioniert, oder Ihre Umgebung so konfigurieren, dass Zustandsprüfungen während der Bereitstellung ignoriert werden und die neue Version erneut bereitgestellt wird, um einen Abschluss der Bereitstellung zu erzwingen.

Sie können auch die Instances identifizieren und beenden, die mit der falschen Anwendungsversion ausgeführt werden. Elastic Beanstalk startet Instances mit der richtigen Version, um alle Instances zu ersetzen, die Sie beenden. Verwenden Sie den EB-CLI-Zustandsbefehl, um Instances zu identifizieren, die mit der falschen Anwendungsversion ausgeführt werden.

Anwendungsserver

15 % der Abfragen geben den Fehler HTTP 4xx zurück

20 % der Abfragen an den ELB geben den Fehler HTTP 4xx zurück.

Ein hoher Prozentsatz der HTTP-Anfragen an eine Instance oder eine Umgebung schlagen mit 4xx-Fehlern fehl.

Ein Serienstatuscode 400 gibt an, dass der Benutzer eine fehlerhafte Anfrage getätigt hat, beispielsweise die Anfrage einer Seite, die nicht vorhanden ist (404 File Not Found) oder auf die der Benutzer keinen Zugriff hat (403 Forbidden). Eine geringe Anzahl von 404s ist nicht ungewöhnlich, aber eine große Anzahl kann bedeuten, dass es interne oder externe Links zu nicht verfügbaren Seiten gibt. Diese Probleme können behoben werden, indem fehlerhafte interne Links repariert und Umleitungen für fehlerhafte externe Links hinzugefügt werden.

5 % der Anfragen schlagen mit HTTP-5xx fehl

3 % der Abfragen an den ELB schlagen mit HTTP 5xx fehl.

Ein hoher Prozentsatz der HTTP-Anfragen an eine Instance oder eine Umgebung schlagen mit Serienstatuscodes 500 fehl.

Ein Serienstatuscode 500 gibt an, dass beim Anwendungsserver ein interner Fehler aufgetreten ist. Diese Fehler zeigen an, dass es einen Fehler in Ihrem Anwendungscode gibt, der identifiziert und schnell behoben werden sollte.

95 % der CPU wird verwendet

Auf einer Instance meldet der Zustandsagent einen extrem hohen Prozentsatz an CPU-Auslastung und setzt den Instance-Zustand auf Warning (Warnung) oder Degraded (Schwach).

Skalieren Sie Ihre Umgebung, um Last von Instances zu nehmen.

Worker-Instances

20 Nachrichten in der Warteschlange (vor 25 Sekunden)

Anfragen werden schneller zur Warteschlange Ihrer Worker-Umgebung hinzugefügt, als sie verarbeitet werden können. Skalieren Sie Ihre Umgebung zur Erhöhung der Kapazität.

5 Nachrichten in Warteschlange für unzustellbare Nachrichten (vor 15 Sekunden)

Worker-Anfragen schlagen wiederholt fehl und werden zur hinzugefügt the section called “Warteschlangen mit unzustellbaren Briefen”. Prüfen Sie die Anfragen in der Warteschlange für unzustellbare Nachrichten, um zu sehen, warum sie nicht erfüllt werden.

Sonstige Ressourcen

4 aktive Instances liegen unter der Mindestgröße für Auto Scaling-Gruppen von 5

Die Anzahl der Instances in Ihrer Umgebung ist geringer als die minimale Konfiguration für die Auto-Scaling-Gruppe.

Benachrichtigungen der Auto Scaling-Gruppe (Gruppenname) wurden gelöscht oder geändert

Die Benachrichtigungen, die für Ihre Auto-Scaling-Gruppe konfiguriert sind, wurden außerhalb von Elastic Beanstalk geändert.

KI-gestützte Umgebungsanalyse

AWS Elastic Beanstalk Die KI-gestützte Analyse identifiziert die Grundursachen und empfiehlt Lösungen für umweltbedingte Gesundheitsprobleme. Wenn in Ihrer Umgebung Probleme

auftreten, können Sie mithilfe der Operationen `RequestEnvironmentInfo` und der `RetrieveEnvironmentInfo` API-Operationen mit dem `analyze` Informationstyp eine KI-Analyse anfordern, um KI-generierte Erkenntnisse und empfohlene Lösungen zu erhalten.

> **Note**
>
> Die KI-Analyse ist nur für unterstützte Amazon Linux 2- und AL2023 Plattformversionen verfügbar, die am oder nach dem 16. Februar 2026 veröffentlicht wurden.

Funktionsweise

Wenn Sie eine KI-Analyse anfordern, führt Elastic Beanstalk ein Skript auf einer Instance in Ihrer Umgebung aus, das aktuelle Ereignisse, den Instanzstatus und Logs (bis zu 170.000 Tokens an Daten) sammelt. Anschließend werden diese Daten an Amazon Bedrock in Ihrem Konto gesendet und Sie erhalten Einblicke und empfohlene nächste Schritte.

Voraussetzungen

Bevor Sie die KI-Analyse verwenden, stellen Sie sicher, dass Ihre Umgebung die folgenden Anforderungen erfüllt:

- Umgebung, auf der eine unterstützte Plattformversion ausgeführt wird
- Instanzprofil mit den erforderlichen Berechtigungen (siehe Erforderliche Berechtigungen unten)
- Details zu anthropischen Anwendungsfällen — Bei der KI-Analyse werden Modelle von Anthropic Claude über Amazon Bedrock verwendet. Anthropic verlangt von Ihnen, dass Sie ein einmaliges Formular mit Einzelheiten zu einem Anwendungsfall einreichen, bevor Sie ihre Modelle aufrufen können. Um dieses Formular einzureichen, wählen Sie ein beliebiges Anthropic-Modell aus dem Modellkatalog in der Amazon Bedrock-Konsole aus oder rufen Sie die API auf `PutUseCaseForModelAccess`. Sie müssen dies nur einmal pro AWS Konto tun. Wenn Sie das Formular über das Verwaltungskonto der AWS Organizations einreichen, deckt es automatisch alle Mitgliedskonten der Organisation ab. Weitere Informationen finden Sie unter Zugriff auf Amazon Bedrock Foundation-Modelle.
- GovCloud Regionen — Wenn Sie Regionen AWS GovCloud (USA) verwenden, müssen Sie den Zugriff auf das neueste Modell von Anthropic Claude Sonnet and/or Opus in Amazon Bedrock aktivieren, bevor Sie die KI-Analyse verwenden können. Anweisungen zur Aktivierung des Modellzugriffs in GovCloud Regionen finden Sie unter Zugriff auf Amazon Bedrock Foundation-

Modelle verwalten. Informationen zum neuesten verfügbaren Modell von Anthropic Claude Sonnet and/or Opus finden Sie unter Unterstützte Regionen und Modelle für Inferenzprofile.

Erforderliche Berechtigungen

Um die KI-Analyse verwenden zu können, muss das Amazon EC2 EC2-Instance-Profil für Ihre Umgebung über Berechtigungen zum Aufrufen von Amazon Bedrock verfügen. Fügen Sie Ihrem Instance-Profil die folgenden Berechtigungen hinzu:

- `bedrock:InvokeModel`
- `bedrock:ListFoundationModels`
- `elasticbeanstalk:DescribeEvents`
- `elasticbeanstalk:DescribeEnvironmentHealth`

Weitere Informationen zur Konfiguration von Instanzprofilen finden Sie unterElastic Beanstalk Instance-Profile verwalten.

KI-Analyse in der Konsole verwenden

Aus der Umgebungsübersicht

Wenn der Integritätsstatus Ihrer Umgebung „Warnung“, „Eingeschwächt“ oder „Schwerwiegend“ lautet, wird im Bereich „Umgebungsübersicht“ eine KI-Analyseschaltfläche angezeigt. Klicken Sie auf diese Schaltfläche, um eine KI-Analyse Ihrer Umgebung zu starten.

Von der Seite „Logs“

Sie können die KI-Analyse auch über die Seite Logs im Navigationsbereich aufrufen. Klicken Sie auf die Schaltfläche KI-Analyse, um eine KI-gestützte Analyse des aktuellen Zustands Ihrer Umgebung anzufordern.

Verwenden Sie die KI-Analyse mit dem AWS CLI

Sie können die Elastic Beanstalk Beanstalk-API über verwenden AWS CLI , um KI-Analysen programmgesteuert anzufordern und abzurufen.

KI-Analyse anfordern

Verwenden Sie die `RequestEnvironmentInfo`Operation, bei der der `InfoType` Parameter auf gesetzt ist`analyze`.

Example AWS CLI - KI-Analyse anfordern

```
aws elasticbeanstalk request-environment-info \
    --environment-name my-env \
    --info-type analyze \
    --region us-east-1
```

KI-Analyse abrufen

Verwenden Sie die `RetrieveEnvironmentInfo`Operation mit dem auf eingestellten `InfoType` Parameter`analyze`, um die Analyseergebnisse abzurufen.

Example AWS CLI - Rufen Sie die KI-Analyse ab

```
aws elasticbeanstalk retrieve-environment-info \
    --environment-name my-env \
    --info-type analyze \
    --region us-east-1
```

Die Antwort umfasst eine KI-generierte Analyse des aktuellen Zustands der Umgebung sowie empfohlene Lösungen für alle identifizierten Probleme.

KI-Analyse mit der EB CLI verwenden

Wenn Sie die EB-CLI verwenden, können Sie mit der Option `--analyze` (`-ai`) des eb logs Befehls eine KI-Analyse anfordern. Der Befehl fordert die Analyse an, wartet, bis sie abgeschlossen ist, und zeigt die Ergebnisse an.

Example EB CLI - KI-Analyse anfordern

```
$ eb logs --analyze
```

Die `--analyze` Option ist nicht kompatibel mit `--instance--all`,`--zip`, oder`--log-group`. Die vollständige Befehlsreferenz finden Sie unter[eb logs](#).

> **Note**
>
> Die `--analyze` Option erfordert EB CLI Version 3.27 oder höher.

Wichtige Überlegungen

- Preisgestaltung — Die KI-Analyse verwendet Amazon Bedrock, um Ihre Umgebungsdaten zu verarbeiten, und für Modellaufrufe gelten die Standardpreise von Amazon Bedrock. Details zu den Preisen finden Sie unter Amazon Bedrock – Preise.
- Plattformanforderung — Die KI-Analyse ist nur für Amazon Linux 2 und AL2023 basierte Plattformversionen verfügbar, die am oder nach dem 16. Februar 2026 veröffentlicht wurden. Um diese Funktion nutzen zu können, aktualisieren Sie Ihre Umgebung auf eine unterstützte Plattformversion. Weitere Informationen finden Sie unter Aktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung.
- Berechtigungen — Stellen Sie vor der Verwendung der KI-Analyse sicher, dass Ihr Instance-Profil über die erforderlichen Amazon Bedrock-Berechtigungen (`bedrock:InvokeModel`und`bedrock:ListFoundationModels`) und Elastic Beanstalk Beanstalk-Berechtigungen (`elasticbeanstalk:DescribeEvents`und) verfügt. `elasticbeanstalk:DescribeEnvironmentHealth`
- Datenschutz — Die Analyse sendet Umgebungsereignisse und Protokolle zur Verarbeitung an Amazon Bedrock in Ihrem Konto. Informationen darüber, wie Amazon Bedrock mit Ihren Daten umgeht, finden Sie unter Amazon Bedrock Security and Compliance.
- Servicekontingente — Die KI-Analyse verwendet ein Anthropic-Claude-Modell in Amazon Bedrock, das Standardkontingente für Anfragen pro Minute und Token pro Minute hat. Wenn Sie auf Drosselungsfehler stoßen, können Sie eine Erhöhung des Kontingents beantragen. Weitere Informationen finden Sie unter Anfordern einer Kontingenterhöhung.

Unterstützte Plattformversionen

Die KI-Analyse wird auf Amazon Linux 2 und AL2023 basierten Plattformversionen unterstützt, die am oder nach dem 16. Februar 2026 veröffentlicht wurden. Informationen zur Überprüfung Ihrer Plattformversion finden Sie in den Versionshinweisen zu Elastic Beanstalk.

Alarme verwalten

Dieses Thema führt Sie durch die Schritte zum Erstellen von Alarmen für Metriken, die Sie überwachen. Es enthält auch Anweisungen zum Anzeigen Ihrer vorhandenen Alarme und zum Überprüfen ihres Status.

Sie können Alarme für Metriken erstellen, die Sie mit der Elastic Beanstalk-Konsole überwachen. Alarme helfen Ihnen dabei, Änderungen an Ihrer AWS Elastic Beanstalk Umgebung zu überwachen, sodass Sie Probleme leicht erkennen und beheben können, bevor sie auftreten. Sie können beispielsweise einen Alarm festlegen, der Sie benachrichtigt, wenn die CPU-Auslastung in einer Umgebung einem bestimmten Schwellenwert überschreitet, wodurch sichergestellt wird, dass Sie benachrichtigt werden, bevor ein potentielles Problem auftritt. Weitere Informationen finden Sie unter Elastic Beanstalk mit Amazon verwenden CloudWatch.

> **Note**
>
> Elastic Beanstalk verwendet CloudWatch für Überwachung und Alarme, was bedeutet, dass die CloudWatch Kosten für alle Alarme, die Sie verwenden, Ihrem AWS Konto belastet werden.

Weitere Informationen zur Überwachung spezifischer Metriken finden Sie unter Grundlegende Zustandsberichte.

So prüfen Sie den Status Ihrer Alarme

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie im Navigationsbereich auf Alarms (Alarme).

   Die Seite zeigt eine Liste vorhandener Alarme an. Wenn sich Alarme im Alarmstatus befinden, werden sie mit dem Warnsymbol () gekennzeichnet.

   
4. Um Alarme zu filtern, wählen Sie das Drop-down-Menü und dann einen Filter aus.
5. Um einen Alarm zu bearbeiten oder zu löschen, wählen Sie das Bearbeitungssymbol () bzw. das Löschsymbol ().

So erstellen Sie einen Alarm

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Monitoring (Überwachung) aus.
4. Suchen Sie die Metrik, für die Sie einen Alarm erstellen möchten, und wählen Sie dann das Alarmsymbol
   () 
   .
   Die Seite Add alarm (Alarm hinzufügen) wird angezeigt.
5. Geben Sie Informationen zum Alarm ein:
   - Name: Ein Name für diesen Alarm.
   - Description (optional): Eine kurze Beschreibung dieses Alarms.
   - Period: Das Zeitintervall zwischen den Auslesungen.
   - Threshold: Beschreibt das Verhalten und den Wert, den die Metrik überschreiten muss, um einen Alarm auszulösen.
   - Change state after: Der Zeitraum nach dem Überschreiten eines Schwellenwerts, der eine Änderung des Status des Alarms auslöst.
   - Notify: Das Amazon SNS-Thema, das benachrichtigt wird, wenn ein Alarm einen anderen Zustand annimmt.
   - Benachrichtigung, wenn sich der Zustand in folgende Status ändert:
     - OK: Die Metrik liegt innerhalb des festgelegten Schwellenwerts.
     - Alarm: Die Metrik hat den festgelegten Schwellenwert überschritten.
     - Insufficient data: Der Alarm wurde soeben gestartet; die Metrik ist nicht verfügbar oder es sind nicht genügend Daten verfügbar, damit die Metrik den Alarmstatus bestimmen kann.
6. Wählen Sie Hinzufügen aus. Der Umgebungsstatus wechselt zu Grau, während die Umgebung aktualisiert wird. Sie können den erstellten Alarm anzeigen, indem Sie im Navigationsbereich Alarms (Alarme) wählen.

Anzeigen des Änderungsverlaufs einer Elastic Beanstalk-Umgebung

In diesem Thema wird erklärt, wie Sie die Elastic Beanstalk Console verwenden können, um einen Verlauf der Konfigurationsänderungen anzuzeigen, die an Ihren Elastic Beanstalk Beanstalk-Umgebungen vorgenommen wurden.

Elastic Beanstalk ruft Ihren Änderungsverlauf aus Ereignissen ab, die in AWS CloudTrail aufgezeichnet wurden, und zeigt sie in einer Liste an, durch die Sie einfach navigieren und filtern können.

Das Fenster „Änderungsverlauf“ zeigt die folgenden Informationen für Änderungen an Ihren Umgebungen an:

- Das Datum und die Uhrzeit, zu der eine Änderung vorgenommen wurde
- Den IAM-Benutzer, der für eine vorgenommene Änderung verantwortlich war
- Das Quelltool (entweder die Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) oder die Konsole), mit dem die Änderung vorgenommen wurde
- Den Konfigurationsparameter und neue Werte, die festgelegt wurden

Alle sensiblen Daten, die Teil der Änderung sind, wie die Namen der von der Änderung betroffenen Datenbankbenutzer, werden nicht angezeigt.

So zeigen Sie den Änderungsverlauf an

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Change history (Änderungsverlauf) aus.

   Auf der Seite „Change History“ (Änderungsverlauf) wird eine detaillierte Liste der Konfigurationsänderungen angezeigt, die an Ihren Elastic-Beanstalk-Umgebungen vorgenommen wurden.

Beachten Sie beim Navigieren in den Informationen auf dieser Seite die folgenden Punkte:

- Sie können durch die Liste blättern, indem Sie **<** (Zurück) oder **>** (Weiter) wählen oder eine bestimmte Seitenzahl wählen.

- Wählen Sie in der Spalte Konfigurationsänderungen das Pfeilsymbol aus, um zwischen dem Erweitern und Reduzieren der Liste der Änderungen unter der Überschrift Vorgenommene Änderungen zu wechseln.
- Verwenden Sie die Suchleiste, um Ihre Ergebnisse aus der Liste des Änderungsverlaufs zu filtern. Sie können eine beliebige Zeichenfolge eingeben, um die Liste der angezeigten Änderungen einzugrenzen.

Beachten Sie Folgendes zum Filtern der angezeigten Ergebnisse:

- Beim Suchefilter wird die Groß-/Kleinschreibung nicht beachtet.
- Sie können angezeigte Änderungen basierend auf Informationen in der Spalte Konfigurationsänderungen filtern, auch wenn sie nicht sichtbar sind, da sie innerhalb der vorgenommenen Änderungen reduziert wurden.
- Sie können nur die angezeigten Ergebnisse filtern. Der Filter bleibt jedoch bestehen, auch wenn Sie auf eine andere Seite gehen möchten, um weitere Ergebnisse anzuzeigen. Ihre gefilterten Ergebnisse werden an den Ergebnissatz auf der nächsten Seite angehängt.

Die folgenden Beispiele zeigen, wie die auf dem vorherigen Bildschirm angezeigten Daten gefiltert werden können:

- Geben Sie **`GettingStartedApp-env`** in das Suchfeld ein, um die Ergebnisse so einzugrenzen, dass sie nur die Änderungen enthalten, die an der Umgebung mit dem Namen GettingStartedApp-env vorgenommen wurden.
- Geben Sie **`example3`** in das Suchfeld ein, um die Ergebnisse so einzugrenzen, dass sie nur Änderungen enthalten, die von IAM-Benutzern vorgenommen wurden, deren Benutzername die Zeichenfolge example3 enthält.
- Geben Sie **`2020-10`** in das Suchfeld ein, um die Ergebnisse so einzugrenzen, dass nur Änderungen enthalten sind, die im Oktober 2020 vorgenommen wurden. Ändern Sie den Suchwert in **`2020-10-16`**, um die angezeigten Ergebnisse weiter zu filtern, sodass nur Änderungen enthalten sind, die am 16. Oktober 2020 vorgenommen wurden.
- Geben Sie **`proxy:staticfiles`** in das Suchfeld ein, um die Ergebnisse so einzugrenzen, dass sie nur die Änderungen einschließen, die am Namespace namens aws:elasticbeanstalk:environment:proxy:staticfiles vorgenommen wurden. Die Zeilen, die angezeigt werden, sind das Ergebnis des Filters. Dies gilt auch für Ergebnisse, die unter Änderungen reduziert sind.

Ereignis-Stream einer Elastic Beanstalk-Umgebung anzeigen

In diesem Thema wird erklärt, wie Sie auf Ereignisse und Benachrichtigungen zugreifen, die mit Ihrer Anwendung verknüpft sind.

Ereignisse mit der Elastic Beanstalk Beanstalk-Konsole anzeigen

So zeigen Sie Ereignisse mit der Elastic Beanstalk Beanstalk-Konsole an

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich die Option Events.

   Die Seite „Event“ (Ereignisse) zeigt eine Liste aller Ereignisse an, die für die Umgebung aufgezeichnet wurden. Sie können durch die Liste blättern, indem Sie **<** (vorherige), **>** (nächste) oder Seitenzahlen wählen. Sie können nach der Art der angezeigten Ereignisse filtern, indem Sie die Drop-down-Liste Severity (Schweregrad) verwenden.

Ereignisse mit Befehlszeilentools anzeigen

Die EB CLI und die AWS CLI bieten Befehle zum Abrufen von Ereignissen. Wenn Sie Ihre Umgebung mit der EB CLI verwalten, verwenden Sie eb events, um eine Liste der Ereignisse zu drucken. Dieser Befehl verfügt zudem über eine `--follow`-Option, die weiterhin neue Ereignisse angezeigt, bis Sie Strg+C drücken, um die Ausgabe zu beenden.

Um Ereignisse mit dem abzurufen AWS CLI, verwenden Sie den `describe-events` Befehl und geben Sie die Umgebung mit Namen oder ID an:

```
$ aws elasticbeanstalk describe-events --environment-id e-gbjzqccra3
{
    "Events": [
        {
            "ApplicationName": "elastic-beanstalk-example",
            "EnvironmentName": "elasticBeanstalkExa-env",
            "Severity": "INFO",
```

```
            "RequestId": "a4c7bfd6-2043-11e5-91e2-9114455c358a",
            "Message": "Environment update completed successfully.",
            "EventDate": "2015-07-01T22:52:12.639Z"
        },
...
```

Weitere Informationen zu den Befehlszeilentools finden Sie unter???.

Auflisten von Server-Instances/Verbinden mit Server-Instances

In diesem Thema wird erklärt, wie Sie eine Liste der EC2 Amazon-Instances anzeigen, auf denen Ihre Elastic Beanstalk Beanstalk-Anwendungsumgebung ausgeführt wird, und wie Sie eine Verbindung zu ihnen herstellen können.

In der Elastic Beanstalk Beanstalk-Konsole können Sie sich eine Liste der EC2 Amazon-Instances ansehen, auf denen Ihre AWS Elastic Beanstalk Anwendungsumgebung ausgeführt wird. Eine Verbindung zu den Instances stellen Sie über einen beliebigen SSH-Client her. Die Verbindung zu Instances, auf denen Windows ausgeführt wird, können Sie mit Remote Desktop herstellen.

Important

Bevor Sie auf Ihre von Elastic Beanstalk bereitgestellten EC2 Amazon-Instances zugreifen können, müssen Sie ein EC2 Amazon-Schlüsselpaar erstellen und Ihr von Elastic Beanstalk bereitgestelltes Amazon so konfigurieren, dass es das Amazon-Schlüsselpaar verwendet. EC2instances EC2 Sie können Ihre EC2 Amazon-Schlüsselpaare mithilfe der AWS Management Console einrichten. Anweisungen zum Erstellen eines key pair für Amazon EC2 finden Sie im Amazon-Leitfaden „ EC2 Erste Schritte“. Weitere Informationen zur Konfiguration Ihrer EC2 Amazon-Instances für die Verwendung eines EC2 Amazon-Schlüsselpaars finden Sie unterEC2 key pair.
Standardmäßig ermöglicht Elastic Beanstalk keine Remoteverbindungen zu EC2 Instances in einem Windows-Container, außer denen in älteren Windows-Containern. (Elastic Beanstalk konfiguriert EC2 Instances in älteren Windows-Containern so, dass sie Port 3389 für RDP-Verbindungen verwenden.) Sie können Remoteverbindungen zu Ihren EC2 Windows-Instances aktivieren, indem Sie einer Sicherheitsgruppe eine Regel hinzufügen, die eingehenden Datenverkehr zu den Instances autorisiert. Es wird ausdrücklich empfohlen, diese Regel bei Beendigung der Remote-Verbindung zu entfernen. Sie können die Regel bei der nächsten Remote-Anmeldung wieder hinzufügen. Weitere Informationen finden Sie unter Hinzufügen einer Regel für eingehenden RDP-Verkehr zu einer Windows-Instance und

Verbinden mit der Windows Instanz im Amazon Elastic Compute Cloud-Benutzerhandbuch für Microsoft Windows.///

So zeigen Sie EC2 Amazon-Instances für eine Umgebung an und stellen eine Verbindung zu ihnen her

1. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich der Konsole Load Balancers aus.
3. Load Balancer, die von Elastic Beanstalk erstellt wurden, haben awseb im Namen. Suchen Sie nach dem Load Balancer für Ihre Umgebung und klicken Sie darauf.
4. Wählen Sie im unteren Abschnitt der Konsole die Registerkarte Instances aus.

   Die Liste der Instances, die vom Load Balancer für die Elastic Beanstalk-Umgebung verwendet werden, wird angezeigt. Notieren Sie sich die ID der Instance, zu der eine Verbindung hergestellt werden soll.
5. Wählen Sie im Navigationsbereich der EC2 Amazon-Konsole Instances aus und suchen Sie Ihre Instance-ID in der Liste.
6. Klicken Sie mit der rechten Maustaste auf die Instance-ID für die EC2 Amazon-Instance, die im Load Balancer Ihrer Umgebung ausgeführt wird, und wählen Sie dann im Kontextmenü Connect aus.
7. Notieren Sie sich die öffentliche DNS-Adresse der Instance von der Registerkarte Description.
8. Stellen Sie mithilfe des SSH-Clients Ihrer Wahl eine Connect zu einer Instance her, auf der Linux ausgeführt wird, und geben Sie dann ssh -i .ec2/mykeypair.pem ec2-user@<public-DNS - > ein. of-the-instance

Weitere Informationen zum Herstellen einer Verbindung mit einer Amazon EC2 Linux-Instance finden Sie unter Erste Schritte mit Amazon EC2 Linux-Instances im EC2 Amazon-Benutzerhandbuch.

Wenn Ihre Elastic Beanstalk Beanstalk-Umgebung die Plattform .NET auf Windows Server verwendet, finden Sie weitere Informationen unter Erste Schritte mit Amazon EC2 Windows-Instances im EC2 Amazon-Benutzerhandbuch.

Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen

In diesem Thema werden die Arten von Instance-Logs erklärt, die Elastic Beanstalk bereitstellt. Es enthält auch detaillierte Anweisungen zum Abrufen und Verwalten dieser Dateien.

Die Amazon EC2-Instances der Elastic Beanstalk-Umgebung generieren Protokolle, die Sie zur Behebung von Fehlern mit der Anwendung oder mit Konfigurationsdateien aufrufen können. Protokolle, die vom Webserver, dem Anwendungsserver und den Skripten der Elastic Beanstalk-Plattform erstellt und lokal auf einzelnen Instances gespeichert CloudFormation werden. Sie können diese über die Environment Management Console oder die EB CLI ganz einfach abrufen. Sie können Ihre Umgebung auch so konfigurieren, dass Protokolle in Echtzeit an Amazon CloudWatch Logs gestreamt werden.

Als Protokollfragmente werden die letzten 100 Zeilen der am häufigsten verwendeten Protokolldateien bezeichnet, z. B. Elastic Beanstalk-Betriebsprotokolle sowie Protokolle vom Webserver oder Anwendungsserver. Wenn Sie Protokollfragmente in der Environment Management Console oder mit eb logs anfordern, verkettet eine Instance der Umgebung die letzten Protokolleinträge zu einer einzigen Textdatei und lädt diese in Amazon S3 hoch.

Bei Bundle-Protokollen handelt es sich um vollständige Protokolle für ein breiteres Spektrum an Protokolldateien, darunter Protokolle von "yum" und "cron" und verschiedene Protokolle von CloudFormation. Bei der Anforderung von Bundle-Protokollen komprimiert eine Instance der Umgebung die vollständigen Protokolldateien in ein ZIP-Archiv und lädt dieses in Amazon S3 hoch.

Damit rotierte Protokolle in Amazon S3 hochgeladen werden können, müssen die Instances der Umgebung über ein Instance-Profil mit Schreibberechtigung für den Elastic Beanstalk Amazon S3-Bucket verfügen. Diese Berechtigungen sind im Instance-Standardprofil enthalten, zu dessen Erstellung Sie von Elastic Beanstalk aufgefordert werden, wenn Sie zum ersten Mal eine Umgebung in der Elastic Beanstalk-Konsole starten.

So rufen Sie Instance-Protokolle ab:

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Protokolle aus.

4. Wählen Sie Request Logs (Protokolle anfordern) und wählen Sie dann den Typ der Protokolle, die abgerufen werden sollen. Um Protokollfragmente abzurufen, wählen Sie Last 100 Lines aus. Für den Abruf von Bundle-Protokollen wählen Sie Full Logs aus.
5. Wenn Elastic Beanstalk mit dem Abrufen Ihrer Protokolle fertig ist, wählen Sie Download (Herunterladen) aus.

Elastic Beanstalk speichert Tail- und Bundle-Protokolle in einem Amazon S3-Bucket und generiert eine vordefinierte Amazon S3-URL, über die Sie auf Ihre Protokolle zugreifen können. Elastic Beanstalk löscht die Dateien von Amazon S3 nach 15 Minuten.

> **Warning**
>
> Jeder im Besitz der vorsignierten Amazon S3-URL kann vor dem Löschen auf die Dateien zugreifen. Geben Sie nur vertrauenswürdigen Parteien Zugriff auf die URL.

> **Note**
>
> Ihre Benutzerrichtlinie muss die Berechtigung `s3:DeleteObject` haben. Elastic Beanstalk verwendet Ihre Benutzerberechtigungen, um die Protokolle von Amazon S3 zu löschen.

Damit Protokolle erhalten bleiben, können Sie Ihre Umgebung so konfigurieren, dass Protokolle nach dem Rotieren automatisch in Amazon S3 veröffentlicht werden. Befolgen Sie zum Rotieren von Protokollen in Amazon S3 die Anleitung unter Konfigurieren der Anzeige von Instance-Protokollen. Instances in der Umgebung versuchen dann, Protokolle hochzuladen, die einmal pro Stunde rotiert wurden.

Wenn die Anwendung Protokolle an einem Speicherort generiert, der nicht zur Standardkonfiguration der Umgebungsplattform gehört, können Sie die Standardkonfiguration mithilfe von Konfigurationsdateien erweitern (`.ebextensions`). Die Protokolldateien der Anwendung können zu Protokollfragmenten, zu Bundle-Protokollen oder zur Protokollrotation hinzugefügt werden.

Für Echtzeit-Log-Streaming und Langzeitspeicherung konfigurieren Sie Ihre Umgebung so, dass Logs an Amazon CloudWatch Logs gestreamt werden.

Informationen zur KI-gestützten Analyse Ihrer Umgebungsprotokolle, Ereignisse und des Zustands Ihrer Instances zur Identifizierung von Ursachen und Lösungen für Integritätsprobleme finden Sie unterKI-gestützte Umgebungsanalyse.

Sections

- Speicherort der Protokolle auf Amazon EC2-Instances
- Speicherort der Protokolle in Amazon S3
- Protokollrotations-Einstellungen auf Linux
- Erweitern der Standardkonfiguration für Protokollaufgaben
- Protokolldateien nach Amazon CloudWatch Logs streamen

Speicherort der Protokolle auf Amazon EC2-Instances

Protokolle werden an Standardspeicherorten auf den Amazon EC2-Instances Ihrer Umgebung gespeichert. Elastic Beanstalk erzeugt die folgenden Protokolle.

Amazon Linux 2

- `/var/log/eb-engine.log`

Amazon Linux-AMI (AL1)

> **Note**
>
> Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

- `/var/log/eb-activity.log`
- `/var/log/eb-commandprocessor.log`

Windows Server

- `C:\Program Files\Amazon\ElasticBeanstalk\logs\`
- `C:\cfn\log\cfn-init.log`

Diese Protokolle enthalten Meldungen über Bereitstellungsaktivitäten, einschließlich solcher über Konfigurationsdateien (`.ebextensions`).

Jede Anwendung und jeder Webserver speichert Protokolle in einem eigenen Ordner:

- Apache – `/var/log/httpd/`
- IIS – `C:\inetpub\wwwroot\`
- Node.js – `/var/log/nodejs/`
- nginx – `/var/log/nginx/`
- Passenger – `/var/app/support/logs/`
- Puma – `/var/log/puma/`
- Python – `/opt/python/log/`
- Tomcat – `/var/log/tomcat/`

Speicherort der Protokolle in Amazon S3

Wenn Sie Protokollfragmente oder Bundle-Protokolle aus Ihrer Umgebung anfordern oder wenn Instances rotierte Protokolle hochgeladen haben, werden diese in Ihrem Elastic Beanstalk-Bucket in Amazon S3 gespeichert. Elastic Beanstalk erstellt einen Bucket, der `elasticbeanstalk-`*`region`*`-`*`account-id`* nach jeder AWS Region benannt ist, in der Sie Umgebungen erstellen. In diesem Bucket werden Protokolle unter dem Pfad `resources/environments/logs/`*`logtype`*`/`*`environment-id`*`/`*`instance-id`* gespeichert.

Beispielsweise werden Logs von der Instanz `i-0a1fd158` in der Elastic Beanstalk Beanstalk-Umgebung unter AWS Region `e-mpcwnwheky` `us-west-2` im Account `123456789012` an den folgenden Orten gespeichert:

- Tail Logs –

  `s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/tail/e-mpcwnwheky/i-0a1fd158`
- Bundle Logs –

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/
bundle/e-mpcwnwheky/i-0a1fd158
```

- Rotated Logs –

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/
publish/e-mpcwnwheky/i-0a1fd158
```

> **Note**
>
> Die Umgebungs-ID finden Sie in der Environment Management Console.

Elastic Beanstalk löscht Tail- und Bundle-Protokolle von Amazon S3 automatisch 15 Minuten nach ihrer Erstellung. Rotierte Protokolle bleiben bestehen, bis Sie sie löschen oder nach Amazon Glacier verschieben.

Protokollrotations-Einstellungen auf Linux

Auf Linux-Plattformen verwendet Elastic Beanstalk `logrotate`, um Protokolle periodisch zu rotieren. Nach der lokalen Rotation des Protokolls erfasst es die Protokollrotationsaufgabe und lädt es in Amazon S3 hoch (sofern dies konfiguriert ist). Lokal rotierte Protokolle werden standardmäßig nicht in Protokollfragmente oder Bundle-Protokolle aufgenommen.

Elastic Beanstalk-Konfigurationsdateien für `logrotate` finden Sie unter `/etc/logrotate.elasticbeanstalk.hourly/`. Diese Rotationseinstellungen sind plattformspezifisch und können sich in späteren Versionen der Plattform ändern. Führen Sie `man logrotate` aus, um weitere Informationen zu den verfügbaren Einstellungen sowie Beispielkonfigurationen zu erhalten.

Die Konfigurationsdateien werden von Cron-Aufträgen in `/etc/cron.hourly/` aufgerufen. Führen Sie zum Erhalten weiterer Informationen zu `cron` `man cron` aus.

Erweitern der Standardkonfiguration für Protokollaufgaben

In Elastic Beanstalk werden Dateien in den Unterordnern von `/opt/elasticbeanstalk/tasks` (Linux) oder `C:\Program Files\Amazon\ElasticBeanstalk\config` (Windows Server) auf der Amazon EC2-Instance verwendet, um Aufgaben für Protokollfragmente, Bundle-Protokolle und Protokollrotation zu konfigurieren.

Auf Amazon Linux:

- Tail Logs –

  `/opt/elasticbeanstalk/tasks/taillogs.d/`
- Bundle Logs –

  `/opt/elasticbeanstalk/tasks/bundlelogs.d/`
- Rotated Logs –

  `/opt/elasticbeanstalk/tasks/publishlogs.d/`

Auf Windows Server:

- Tail Logs –

  `c:\Program Files\Amazon\ElasticBeanstalk\config\taillogs.d\`
- Bundle Logs –

  `c:\Program Files\Amazon\ElasticBeanstalk\config\bundlelogs.d\`
- Rotated Logs –

  `c:\Program Files\Amazon\ElasticBeanstalk\config\publogs.d\`

Beispielsweise werden mit der Datei `eb-activity.conf` auf Linux zwei Protokolldateien zur Protokollfragmentaufgabe hinzugefügt:

`/opt/elasticbeanstalk/tasks/taillogs.d/eb-activity.conf`

```
/var/log/eb-commandprocessor.log
/var/log/eb-activity.log
```

Sie können Umgebungskonfigurationsdateien (`.ebextensions`) verwenden, um Ihre eigenen `.conf`-Dateien diesen Ordnern hinzuzufügen. Eine `.conf`-Datei listet für Ihre Anwendung spezifische Protokolldateien auf, die Elastic Beanstalk zu den Protokolldateiaufgaben hinzufügt.

Verwenden Sie den Abschnitt `files`, um Konfigurationsdateien zu den Aufgaben hinzuzufügen, die geändert werden sollen. Der folgende Konfigurationstext fügt beispielsweise eine Protokollkonfigurationsdatei zu den einzelnen Instances Ihrer Umgebung hinzu. Die

Protokollkonfigurationsdatei `cloud-init.conf` fügt `/var/log/cloud-init.log` zu Protokollfragmenten hinzu.

```
files:
  "/opt/elasticbeanstalk/tasks/taillogs.d/cloud-init.conf" :
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/log/cloud-init.log
```

Fügen Sie diesen Text zu einer Datei mit der Dateinamenerweiterung `.config` zu Ihrem Quell-Bundle in einen Ordner namens `.ebextensions` hinzu.

```
~/workspace/my-app
|-- .ebextensions
|   `-- tail-logs.config
|-- index.php
`-- styles.css
```

Auf Linux-Plattformen können Sie bei der Konfiguration von Protokollaufgaben auch Platzhalterzeichen einsetzen. Mithilfe dieser Konfigurationsdatei werden alle Dateien mit der Erweiterung `.log`, die sich im Ordner `log` des Anwendungsstamms befinden, zu Bundle-Protokollen hinzugefügt.

```
files:
  "/opt/elasticbeanstalk/tasks/bundlelogs.d/applogs.conf" :
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/app/current/log/*.log
```

Konfiguration von Protokollaufgaben unterstützen keine Platzhalterzeichen auf Windows-Plattformen.

> **Note**
>
> Wenn Sie sich mit den Verfahren zur Anpassung von Protokollen vertraut machen möchten, können Sie mithilfe der EB-CLI eine Beispielanwendung bereitstellen. Dazu erstellt die EB-CLI ein lokales Anwendungsverzeichnis, in dem `.ebextentions`-Unterverzeichnis mit einer

Beispielkonfiguration enthalten ist. Sie können die Protokolldateien der Beispielanwendung auch dazu verwenden, die in diesem Thema beschriebene Protokollabruffunktion zu untersuchen.

Weitere Informationen zur Verwendung von Konfigurationsdateien finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

So wie Sie Protokollfragmente und Bundle-Protokolle erweitern können, so können Sie auch Protokollrotation mit einer Konfigurationsdatei erweitern. Jedes Mal, wenn Elastic Beanstalk seine eigenen Protokolle rotiert und auf Amazon S3 hochlädt, rotiert es auch Ihre zusätzlichen Protokolle und lädt sie hoch. Die Protokollrotation-Erweiterung verhält sich abhängig vom Betriebssystem der Plattform anders. In den folgenden Abschnitten werden diese beiden Fälle beschrieben.

Erweitern der Protokollrotation auf Linux

Wie in Protokollrotations-Einstellungen auf Linux erläutert, verwendet Elastic Beanstalk `logrotate` zum Rotieren von Protokollen auf Linux-Plattformen. Wenn Sie die Protokolldateien Ihrer Anwendung für Protokollrotation konfigurieren, muss die Anwendung keine Kopien von Protokolldateien erstellen. Elastic Beanstalk konfiguriert `logrotate` so, dass bei jeder Rotation eine Kopie der Protokolldateien Ihrer Anwendung erstellt wird. Aus diesem Grund muss die Anwendung die Protokolldateien entsperrt halten, wenn sie nicht aktiv in sie schreibt.

Ausdehnen der Protokollrotation auf Windows Server

Auf Windows Server muss die Anwendung die Protokolldateien regelmäßig rotieren, wenn Sie Ihre Anwendung für Protokollrotation konfigurieren. Elastic Beanstalk sucht nach Dateien mit Namen, die mit dem von Ihnen konfigurierten Muster beginnen, und markiert sie zum Hochladen in Amazon S3. Darüber hinaus werden Punkte im Dateinamen ignoriert und Elastic Beanstalk betrachtet den Namen bis zum Punkt als Basis-Protokolldateinamen.

Elastic Beanstalk lädt alle Versionen einer Basis-Protokolldatei hoch, mit Ausnahme der neuesten, da es diese als aktive Protokolldatei der Anwendung betrachtet, die möglicherweise gesperrt sein kann. Ihre Anwendung kann daher die aktive Protokolldatei zwischen Rotationen gesperrt halten.

Beispiel: Ihre Anwendung schreibt in eine Protokolldatei mit dem Namen `my_log.log` und Sie geben diesen Namen in der `.conf`-Datei an. Die Anwendung rotiert die Datei in regelmäßigen Abständen. Während des Elastic Beanstalk-Rotationszyklus findet die Anwendung die folgenden Dateien im Ordner der Protokolldatei: `my_log.log`, `my_log.0800.log` und `my_log.0830.log`.

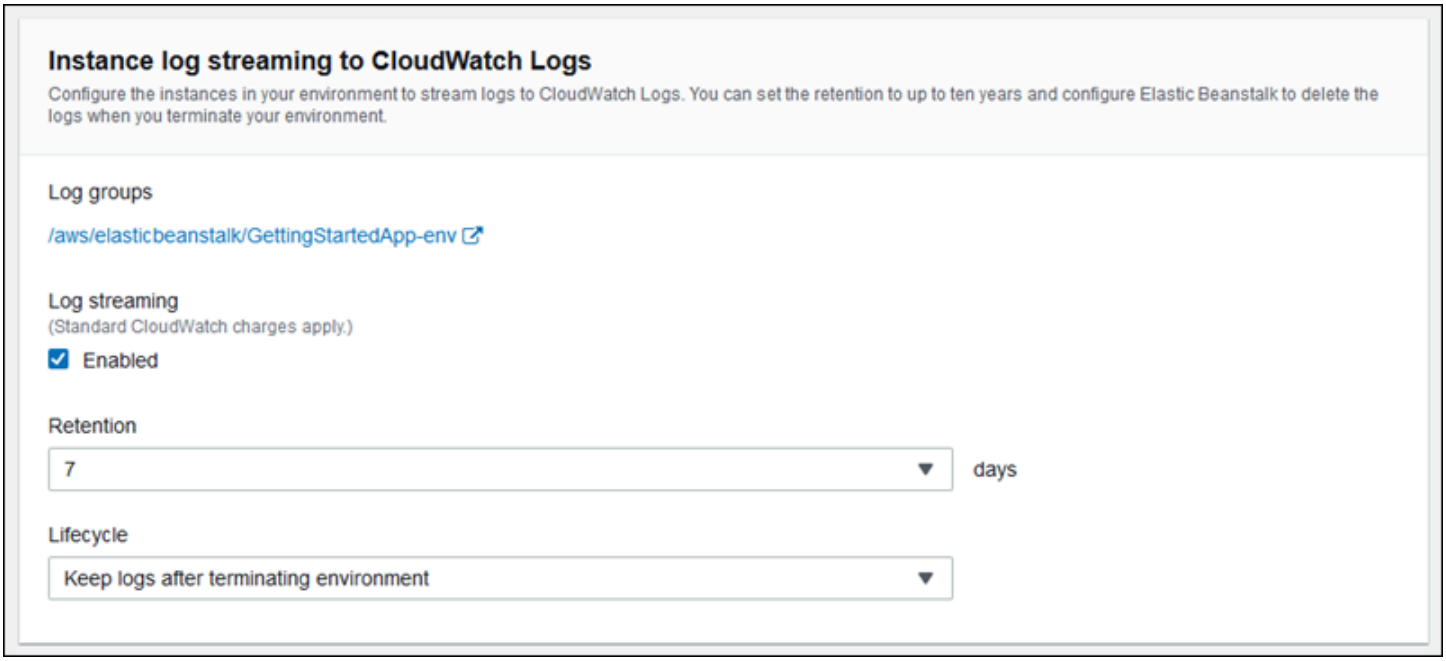
Elastic Beanstalk betrachtet sie alle als Versionen des gleichen Basisnamens `my_log`. Die Datei `my_log.log` hat die neueste Änderungszeit, daher lädt Elastic Beanstalk nur die beiden anderen Dateien, `my_log.0800.log` und `my_log.0830.log`, hoch.

Protokolldateien nach Amazon CloudWatch Logs streamen

Sie können Ihre Umgebung in der Elastic Beanstalk Beanstalk-Konsole oder mithilfe von Konfigurationsoptionen so konfigurieren, dass CloudWatch Logs zu Amazon Logs gestreamt werden. Mit CloudWatch Logs streamt jede Instance in Ihrer Umgebung Logs in Protokollgruppen, die Sie so konfigurieren können, dass sie für Wochen oder Jahre aufbewahrt werden, auch wenn Ihre Umgebung beendet wurde.

Die gestreamten Protokollsätze sind je nach Umgebung unterschiedlich, enthalten aber immer das Protokoll `eb-engine.log` sowie die Zugriffsprotokolle des nginx- oder Apache-Proxy-Servers, der vor der Anwendung ausgeführt wird.

Sie können das Protokoll-Streamen in der Elastic Beanstalk-Konsole entweder beim Erstellen der Umgebung oder für eine vorhandene Umgebung konfigurieren. Sie können die folgenden Optionen von der Konsole aus festlegen: Log-Streaming in CloudWatch Logs aktivieren/deaktivieren, die Anzahl der Aufbewahrungstage festlegen und aus Lifecyle-Optionen wählen. Im folgenden Beispiel werden Protokolle auch dann für sieben Tage gespeichert, wenn die Umgebung beendet wird.



Mit der folgenden Konfigurationsdatei wird das Protokoll-Streaming aktiviert und die Protokolle bleiben 180 Tage erhalten, auch bei beendeter Umgebung.

Example.ebextensions/log-streaming.config

```
option_settings:
  aws:elasticbeanstalk:cloudwatch:logs:
    StreamLogs: true
    DeleteOnTerminate: false
    RetentionInDays: 180
```

Bereitstellungsprotokolle für eine Elastic Beanstalk Beanstalk-Umgebung anzeigen

Elastic Beanstalk generiert für jedes Deployment in Ihrer Umgebung ein Deployment-Log. Das Deployment-Log bietet eine konsolidierte, chronologische Ansicht dessen, was während einer Bereitstellung passiert ist, einschließlich der Installation von Abhängigkeiten, der Build-Ausgabe, des Anwendungsstarts und aller aufgetretenen Fehler. Mithilfe von Bereitstellungsprotokollen können Sie fehlgeschlagene Bereitstellungen schnell diagnostizieren, ohne dass Sie per SSH auf Instanzen zugreifen oder mehrere Protokolldateien korrelieren müssen.

Bereitstellungsprotokolle werden lokal in jede Instanz geschrieben. Bei Bereitstellungen, die über die Konsole, CLI, API oder verwaltete Updates ausgelöst werden, lädt eine Instance ihr Protokoll während der Bereitstellung kontinuierlich auf Amazon S3 hoch. Die Elastic Beanstalk Beanstalk-Konsole liest das Protokoll von Amazon S3, sodass Sie den Fortschritt überwachen können, ohne eine Verbindung zur Instance herstellen zu müssen.

Die Bereitstellungsprotokolle sind so konzipiert, dass sie präzise sind. Bei Erfolg zeigt das Protokoll nur zusammenfassende Meldungen an (z. B. welche Befehle ausgeführt und abgeschlossen wurden). Bei einem Fehler enthält das Protokoll bis zu 50 Zeilen mit der Ausgabe des fehlgeschlagenen Schritts, sodass Sie den Fehler erkennen können, ohne die ausführliche Ausgabe durchsuchen zu müssen.

> **Note**
>
> Bereitstellungsprotokolle sind auf den Plattformversionen Amazon Linux 2 und Amazon Linux 2023 verfügbar, die am oder nach dem 11. März 2026 veröffentlicht wurden. Windows-Plattformen werden derzeit nicht unterstützt.

Unterstützte Vorgänge

Deployment-Logs werden für die folgenden Operationen generiert:

- Anwendungsbereitstellungen — Bereitstellung einer neuen Anwendungsversion in Ihrer Umgebung.
- Konfigurationsupdates — Änderung der Umgebungskonfigurationseinstellungen, für die Instanzupdates erforderlich sind.
- Erstellung einer Umgebung — Die erste Bereitstellung, wenn Sie eine neue Umgebung erstellen.
- App-Server neu starten — Starten Sie den Anwendungsserver auf Ihren Instanzen neu.

Bei Vorgängen, die den Anwendungs- oder Konfigurationsstatus auf Instanzen nicht ändern, wie z. B. das Anfordern von Protokollen, das Austauschen CNAMEs oder Aktualisieren von Tags, werden keine Bereitstellungsprotokolle generiert.

Inhalt des Bereitstellungsprotokolls

Ein Bereitstellungsprotokoll erfasst die folgenden Informationen während einer Bereitstellung:

- Bereitstellungslebenszyklus — Start- und Abschlussnachrichten für jede Bereitstellungsphase, z. B. `Starting Application deployment` und`Completed Application deployment`.
- .ebextensions-Ausgabe — Bei Erfolg die Namen der Befehle, die ausgeführt wurden. Bei einem Fehler die letzten 50 Zeilen der `cfn-init` Ausgabe, um das Problem zu diagnostizieren.
- Plattform-Hooks-Ausgabe — Bei Erfolg die Namen der Hook-Skripte, die ausgeführt wurden. Bei einem Fehler werden die letzten 50 Zeilen der Hook-Ausgabe ausgegeben.
- Installation von Abhängigkeiten — Ausgabe von Paketmanagern wie npm installpip install,composer install, undbundle install. Bei erfolgreicher Ausführung wird nur eine Abschlussmeldung protokolliert. Bei einem Fehler sind die letzten 50 Ausgabezeilen enthalten.
- Build-Ausgabe — Ausgabe von Build-Befehlen wie docker buildgo build, und Java-Builds. Bei einem Fehler sind die letzten 50 Ausgabezeilen enthalten.
- Ausgabe beim Start der Anwendung — Erste Ausgabe Ihrer Anwendung nach dem Start. Die Quelle hängt von Ihrer Plattform ab:
  - Docker — Container-Logs von oder docker logs docker compose logs
  - Java SE, Go, Node.js, Python, Ruby, .NET — Standardprotokolle verarbeiten
  - Tomcat — Catalina-Protokollausgabe

- PHP — PHP-FPM-Master - und Pool-Fehlerprotokolle
- ECS — Container-Protokolle aus jedem Task-Container

> **Note**
>
> Die Anwendungsausgabe wird ab 2 Sekunden nach dem Start der Anwendung erfasst. Es sind nur die ersten Startmeldungen enthalten. Wenn die Ausgabe Ihrer Anwendung länger dauert, werden sie nicht im Bereitstellungsprotokoll angezeigt. Um die vollständigen Anwendungsprotokolle einzusehen, fordern Sie Bundle-Logs an oder stellen Sie eine direkte Verbindung mit der Instance her. Weitere Informationen finden Sie unter Anzeigen von Instance-Protokollen.

Wenn ein Bereitstellungsschritt fehlschlägt, wird er im Protokoll mit `[ERROR]` bis zu 50 Zeilen mit der Ausgabe des fehlgeschlagenen Schritts gekennzeichnet. Wenn das Bereitstellungsprotokoll nicht genügend Details enthält, können Sie die vollständigen Instanzprotokolle (einschließlich `eb-engine.log`eb-hooks.log, und Anwendungsprotokolle) auf der Registerkarte Protokolle abrufen. Weitere Informationen finden Sie unter Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen.

Bereitstellungsprotokolle in der Konsole anzeigen

Die Elastic Beanstalk Beanstalk-Konsole bietet im Umgebungs-Dashboard einen Tab „Deployments“, auf dem Sie Ihren Bereitstellungsverlauf und Ihre Logs einsehen können.

Bereitstellungsverlauf anzeigen

Um den Bereitstellungsverlauf einzusehen

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Umgebungs-Dashboard den Tab Deployments aus.

   Auf der Registerkarte Bereitstellungen wird eine Tabelle mit Bereitstellungen für die Umgebung angezeigt. Jede Zeile enthält die folgenden Informationen:

- Anforderungs-ID — Die eindeutige Kennung für die Bereitstellung.
- Status — Erfolgreich, Fehlgeschlagen oder In Bearbeitung.
- Typ — Der Bereitstellungstyp, z. B. Umgebungserstellung, Anwendungsbereitstellung, Konfigurationsupdate, verwaltetes Plattform-Update, App Server neu starten, Umgebung neu aufbauen, Umgebung wiederherstellen, Umgebungsdomäne austauschen oder Umgebung beenden.
- Startzeit — Wann die Bereitstellung begann.
- Dauer — Wie lange es gedauert hat, bis die Bereitstellung abgeschlossen war.

Wenn eine Bereitstellung im Gange ist, fragt die Registerkarte automatisch nach Updates ab. Sie können auch die Schaltfläche „Aktualisieren“ wählen, um die Liste manuell neu zu laden.

Bereitstellungsdetails und -protokolle anzeigen

Um Bereitstellungsdetails anzuzeigen

1. Wählen Sie auf der Registerkarte Bereitstellungen den Link Anforderungs-ID für die Bereitstellung aus, die Sie überprüfen möchten.
2. Auf der Seite mit den Bereitstellungsdetails wird eine Zusammenfassung mit der Anforderungs-ID, dem Status, dem Bereitstellungstyp, der Startzeit, der Dauer und der Bereitstellungsrichtlinie angezeigt. Die Bereitstellungsrichtlinie (z. B. „Alles auf einmal“, „Rollend“, „Rollend“ mit zusätzlichem Batch, „Unveränderlich“ oder „Traffic Splitting“) wird angezeigt, wenn sie anhand der Bereitstellungsereignisse ermittelt werden kann.
3. Wählen Sie unter der Zusammenfassung eine der folgenden Registerkarten aus:

   - Ereignisse — Eine Zeitleiste mit Ereignissen im Zusammenhang mit dieser Bereitstellung, die so gefiltert wurde, dass nur Ereignisse für die ausgewählte Bereitstellung angezeigt werden.
   - Bereitstellungsprotokolle — Das konsolidierte Bereitstellungsprotokoll der Instance. Sie können die Protokolldatei suchen, nach Protokollebene filtern und herunterladen.

Bei laufenden Bereitstellungen wird die Registerkarte „Protokolle“ automatisch aktualisiert, sodass neue Protokolleinträge angezeigt werden, während sie geschrieben werden. Nach Abschluss einer Bereitstellung ruft die Konsole den endgültigen Protokollstatus ab, um sicherzustellen, dass Sie die vollständige Ausgabe sehen.

> **Important**
>
> Für das Anzeigen von Bereitstellungsprotokollen in der Konsole ist eine `s3:GetObject` Genehmigung für den Amazon S3 S3-Speicher-Bucket der Umgebung erforderlich (`elasticbeanstalk-`*`region`*`-`*`account-id`*). Wenn Ihre IAM-Richtlinie diese Berechtigung nicht beinhaltet, sind der Bereitstellungsverlauf und die Ereignisse weiterhin verfügbar, aber auf der Registerkarte „Protokolle“ wird ein Fehler angezeigt.

Bereitstellungsprotokolldateien auf Instanzen

Bereitstellungsprotokolle werden in das `/var/log/deployments/` Verzeichnis jeder Instance geschrieben. Der Protokolldateiname hängt davon ab, wie die Bereitstellung ausgelöst wurde:

- Workflow-gesteuerte Bereitstellungen (ausgelöst über die Konsole, CLI oder API) —`eb-deployment-`*`request-id`*`.log`, wo *`request-id`* ist die eindeutige Bereitstellungsanforderungs-ID.
- Selbststartende Bereitstellungen (Instanzstart oder Neustart des App-Servers) —. `eb-deployment-`*`unix-timestamp`*`.log`

Elastic Beanstalk rotiert diese Dateien automatisch und behält die 50 neuesten Deployment-Logs für jede Instance bei.

Für Workflow-gesteuerte Bereitstellungen wird das Protokoll unter dem folgenden Pfad in Amazon S3 hochgeladen:

```
s3://elasticbeanstalk-region-account-id/resources/environments/logs/
deployments/environment-id/log-filename
```

In Umgebungen mit mehreren Instanzen beansprucht die erste Instance, die mit dem Hochladen beginnt, die Rolle für die gesamte Bereitstellung. Diese Instance lädt ihr Protokoll für die Dauer der Bereitstellung auf Amazon S3 hoch. Alle Instances schreiben weiterhin Bereitstellungsprotokolle lokal.

> **Important**
>
> Für das Hochladen von Bereitstellungsprotokollen auf Amazon S3 ist eine `s3:PutObject` Genehmigung für den Amazon S3-Speicher-Bucket der Umgebung im Instance-Profil erforderlich, und die VPC-Konfiguration muss die Konnektivität zu Amazon S3 zulassen.

Das Hochladen von Bereitstellungsprotokollen ist auf 1 MB pro Datei begrenzt. Wenn ein Bereitstellungsprotokoll diese Größe überschreitet, wird die hochgeladene Version gekürzt und es wird eine Meldung angezeigt, dass das vollständige Protokoll auf der Instanz verfügbar ist.

S3-Protokoll-Uploads deaktivieren

Um zu verhindern, dass Bereitstellungsprotokolle auf Amazon S3 hochgeladen werden, legen Sie die folgende Umgebungseigenschaft für Ihre Umgebung fest:

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application:environment
    option_name:  EB_DEPLOYMENT_LOG_S3_DISABLED
    value:  true
```

Wenn diese Umgebungseigenschaft festgelegt ist, werden Bereitstellungsprotokolle weiterhin lokal `/var/log/deployments/` auf jeder Instance geschrieben, aber sie werden nicht auf Amazon S3 hochgeladen und sind nicht auf der Registerkarte Deployments in der Konsole verfügbar. Sie können diese Eigenschaft auch auf der Konfigurationsseite unter Software oder mithilfe der EB-CLI oder festlegen AWS CLI.

Elastic Beanstalk mit anderen Diensten verwenden AWS

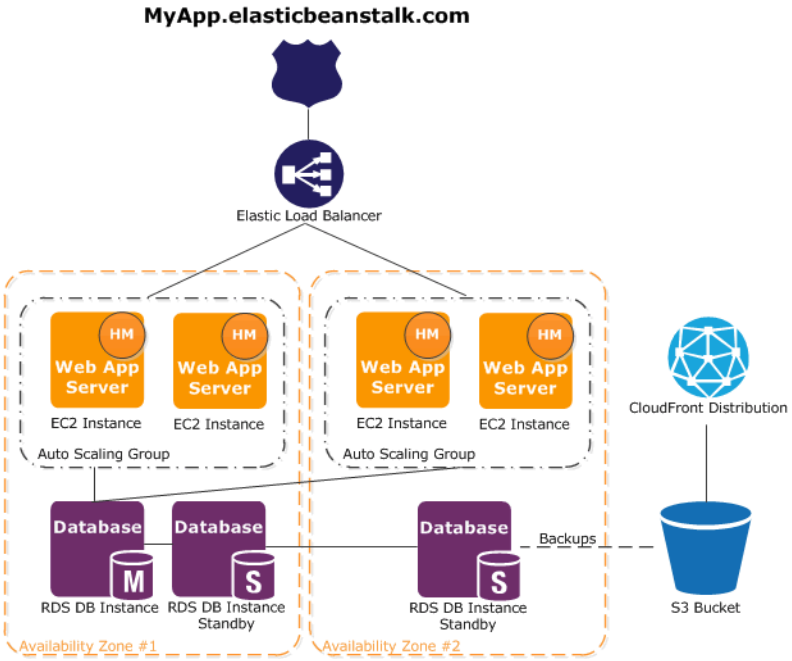
In den Themen dieses Abschnitts werden die vielen Möglichkeiten beschrieben, wie Sie zusätzliche AWS Services mit Ihrer Elastic Beanstalk Beanstalk-Anwendung nutzen können. Um die Umgebungen Ihrer Anwendung zu implementieren, verwaltet Elastic Beanstalk Ressourcen anderer AWS Dienste oder nutzt deren Funktionalität. Darüber hinaus lässt sich Elastic Beanstalk in AWS Dienste integrieren, die es nicht direkt als Teil Ihrer Umgebungen verwendet.

Themen

- Übersicht über die Architektur
- Elastic Beanstalk mit Amazon verwenden CloudFront
- Elastic Beanstalk API-Aufrufe protokollieren mit AWS CloudTrail
- Elastic Beanstalk mit Amazon verwenden CloudWatch
- Elastic Beanstalk mit Amazon Logs verwenden CloudWatch
- Elastic Beanstalk mit Amazon verwenden EventBridge
- Elastic Beanstalk Beanstalk-Ressourcen finden und verfolgen mit AWS Config
- Elastic Beanstalk mit Amazon DynamoDB verwenden
- Elastic Beanstalk mit Amazon verwenden ElastiCache
- Elastic Beanstalk mit Amazon Elastic File System verwenden
- Verwenden von Elastic Beanstalk mit AWS Identity and Access Management
- Verwenden von Elastic Beanstalk mit Amazon RDS
- Verwenden von Elastic Beanstalk mit Amazon S3
- Elastic Beanstalk mit AWS Secrets Manager und AWS Systems Manager Parameter Store verwenden
- Verwenden von Elastic Beanstalk mit Amazon VPC

Übersicht über die Architektur

Das folgende Diagramm zeigt eine Beispielarchitektur von Elastic Beanstalk in mehreren Availability Zones in Zusammenarbeit mit anderen AWS Produkten wie Amazon CloudFront, Amazon Simple Storage Service (Amazon S3) und Amazon Relational Database Service (Amazon RDS).



Um eine Fehlertoleranz einzuplanen, sollten Sie über N+1 Amazon EC2-Instances verfügen und Ihre Instances über mehrere Availability Zones verteilen. Für den unwahrscheinlichen Fall, dass eine Availability Zone ausfällt, haben Sie immer noch Ihre anderen Amazon EC2-Instances, die in einer weiteren Availability Zone ausgeführt werden. Sie können Amazon EC2 Auto Scaling so anpassen, dass eine Mindestanzahl von Instances sowie mehrere Availability Zones zulässig sind. Anweisungen dazu finden Sie unter Auto Scaling Ihrer Elastic Beanstalk Beanstalk-Umgebungsinstanzen. Weitere Informationen zum Erstellen von fehlertoleranten Anwendungen finden Sie unter Erstellung fehlertoleranter Anwendungen in AWS.

In den folgenden Abschnitten wird die Integration mit Amazon CloudFront, Amazon, Amazon DynamoDB Amazon CloudWatch, Amazon RDS ElastiCache, Amazon Route 53, Amazon Simple Storage Service, Amazon VPC und IAM ausführlicher beschrieben.

Elastic Beanstalk mit Amazon verwenden CloudFront

Amazon CloudFront ist ein Webservice, der die Verteilung Ihrer statischen und dynamischen Webinhalte, z. B. .html-, .css-, .php-, Bild- und Mediendateien, an Endbenutzer beschleunigt. CloudFront stellt Ihre Inhalte über ein weltweites Netzwerk von Edge-Standorten bereit. Wenn ein Endbenutzer Inhalte anfordert, die Sie bereitstellen CloudFront, wird der Benutzer an den Edge-Standort weitergeleitet, der die niedrigste Latenz bietet, sodass Inhalte mit der bestmöglichen Leistung bereitgestellt werden. Wenn sich die Inhalte bereits an diesem Edge-Standort befinden, stellt sie CloudFront sofort bereit. Wenn sich der Inhalt derzeit nicht an diesem Edge-Standort befindet,

CloudFront ruft er ihn von einem Amazon S3 S3-Bucket oder einem HTTP-Server (z. B. einem Webserver) ab, den Sie als Quelle für die endgültige Version Ihres Inhalts identifiziert haben.

Nachdem Sie Ihre Elastic Beanstalk Beanstalk-Anwendung erstellt und bereitgestellt haben, können Sie sich registrieren CloudFront und mit der Verbreitung Ihrer CloudFront Inhalte beginnen. Weitere Informationen CloudFront finden Sie im Amazon CloudFront Developer Guide.

Elastic Beanstalk API-Aufrufe protokollieren mit AWS CloudTrail

Elastic Beanstalk ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in Elastic Beanstalk ausgeführt wurden. CloudTrailerfasst alle API-Aufrufe für Elastic Beanstalk als Ereignisse, einschließlich Aufrufe von der Elastic Beanstalk-Konsole, von der EB-CLI und von Ihrem Code an die Elastic Beanstalk. APIs Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Elastic Beanstalk. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Event-Verlauf einsehen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Elastic Beanstalk gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

Informationen zu Elastic Beanstalk in CloudTrail

CloudTrail ist für Ihr AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn eine Aktivität in Elastic Beanstalk auftritt, wird diese Aktivität zusammen mit anderen AWS Service-Ereignissen im CloudTrail Event-Verlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für Elastic Beanstalk, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle -Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS - Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie unter:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle Elastic Beanstalk Beanstalk-Aktionen werden von der AWS Elastic Beanstalk API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Aufrufe der `DescribeApplications` `ListTagsForResource` Aktionen, und generieren beispielsweise Einträge in den CloudTrail Protokolldateien. `UpdateEnvironment`

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Gibt an, ob die Anforderung mit Root- oder IAM-Benutzer-Anmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter CloudTrail userIdentity-Element.

Grundlegendes zu den Elastic Beanstalk-Einträgen in der Protokolldatei

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `UpdateEnvironment` Aktion demonstriert, die von einem IAM-Benutzer mit dem Name`intern`, für die `sample-env` Umgebung in der `sample-app` Anwendung aufgerufen wurde.

```
{
  "Records": [{
```

```
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "AIXDAYQEXAMPLEUMLYNGL",
      "arn": "arn:aws:iam::123456789012:user/intern",
      "accountId": "123456789012",
      "accessKeyId": "ASXIAGXEXAMPLEQULKNXV",
      "userName": "intern",
      "sessionContext": {
        "attributes": {
          "mfaAuthenticated": "false",
          "creationDate": "2016-04-22T00:23:24Z"
        }
      },
      "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2016-04-22T00:24:14Z",
    "eventSource": "elasticbeanstalk.amazonaws.com",
    "eventName": "UpdateEnvironment",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "255.255.255.54",
    "userAgent": "signin.amazonaws.com",
    "requestParameters": {
      "applicationName": "sample-app",
      "environmentName": "sample-env",
      "optionSettings": []
    },
    "responseElements": null,
    "requestID": "84ae9ecf-0280-17ce-8612-705c7b132321",
    "eventID": "e48b6a08-c6be-4a22-99e1-c53139cbfb18",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
  }]
}
```


Elastic Beanstalk mit Amazon verwenden CloudWatch

CloudWatch Mit Amazon können Sie verschiedene Metriken überwachen, verwalten und veröffentlichen sowie Alarmaktionen auf der Grundlage von Messdaten konfigurieren. Mit Amazon CloudWatch Monitoring können Sie System- und Anwendungsmetriken sammeln, analysieren und anzeigen, sodass Sie betriebliche und geschäftliche Entscheidungen schneller und sicherer treffen können.

Sie können Amazon verwenden CloudWatch , um Messwerte über Ihre Amazon Web Services (AWS) -Ressourcen zu sammeln, z. B. die Leistung Ihrer Amazon-Instances. EC2 Sie können Ihre eigenen Kennzahlen auch direkt bei Amazon veröffentlichen CloudWatch. CloudWatch Amazon-Alarme helfen Ihnen dabei, Entscheidungen einfacher zu treffen, indem Sie Benachrichtigungen senden oder automatisch Änderungen an den Ressourcen vornehmen können, die Sie überwachen, basierend auf von Ihnen definierten Regeln. Sie können beispielsweise Alarme erstellen, die Amazon EC2 Auto Scaling- und Amazon Simple Notification Service (Amazon SNS) -Aktionen in Ihrem Namen auslösen.

Elastic Beanstalk verwendet Amazon automatisch, CloudWatch um Sie bei der Überwachung Ihres Anwendungs- und Umgebungsstatus zu unterstützen. Sie können zur CloudWatch Amazon-Konsole navigieren, um Ihr Dashboard aufzurufen und sich einen Überblick über all Ihre Ressourcen sowie Ihre Alarme zu verschaffen. Sie können auch mehrere Metriken anzeigen oder benutzerdefinierte Metriken hinzufügen.

Weitere Informationen zu Amazon CloudWatch finden Sie im Amazon CloudWatch Developer Guide. Ein Beispiel für die Verwendung von Amazon CloudWatch mit Elastic Beanstalk finden Sie unter. the section called “Beispiel: Verwendung benutzerdefinierter CloudWatch Amazon-Metriken”

Elastic Beanstalk mit Amazon Logs verwenden CloudWatch

In diesem Thema werden die Überwachungsfunktionen erklärt, die der Amazon CloudWatch Logs-Service Elastic Beanstalk zur Verfügung stellen kann. Es führt Sie auch durch die Konfiguration und listet die Speicherorte der Protokolle für jede Elastic Beanstalk-Plattform auf.

Durch die Implementierung von CloudWatch Logs können Sie die folgenden Monitoring-Aktivitäten durchführen:

- Überwachen und archivieren Sie Ihre Elastic Beanstalk Beanstalk-Anwendungs-, System- und benutzerdefinierten Protokolldateien aus den Amazon EC2 EC2-Instances Ihrer Umgebungen.
- Konfigurieren Sie Alarme, die es Ihnen erleichtern, auf bestimmte Logstream-Ereignisse zu reagieren, die Ihre Metrikfilter extrahieren.

Der CloudWatch Logs-Agent, der auf jeder Amazon EC2 EC2-Instance in Ihrer Umgebung installiert ist, veröffentlicht für jede von Ihnen konfigurierte Protokollgruppe metrische Datenpunkte für den CloudWatch Service. Jede Protokollgruppe wendet ihre eigenen Filtermuster an, um zu bestimmen, an welche Log-Stream-Ereignisse CloudWatch als Datenpunkte gesendet werden sollen. Protokollstreams, die zur selben Protokollgruppe gehören, besitzen dieselben Einstellungen für die

Aufbewahrung, Überwachung und Zugriffskontrolle. Sie können Elastic Beanstalk so konfigurieren, dass Logs automatisch an den CloudWatch Service gestreamt werden, wie unter beschrieben. Die Streaming-Instanz protokolliert in CloudWatch Logs Weitere Informationen zu CloudWatch Logs, einschließlich Terminologie und Konzepten, finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

Wenn Sie Enhanced Health für Ihre Umgebung aktivieren, können Sie zusätzlich zu Instance-Logs die Umgebung so konfigurieren, dass Integritätsinformationen in CloudWatch Logs gestreamt werden. Siehe Zustandsinformationen der Elastic Beanstalk Beanstalk-Umgebung an Amazon Logs streamen CloudWatch.

Themen

- Voraussetzungen für das Streaming von CloudWatch Instanzprotokollen in Logs
- So richtet Elastic Beanstalk Logs ein CloudWatch
- Die Streaming-Instanz protokolliert in CloudWatch Logs
- Fehlerbehebung bei der Integration CloudWatch von Logs
- Zustandsinformationen der Elastic Beanstalk Beanstalk-Umgebung an Amazon Logs streamen CloudWatch

Voraussetzungen für das Streaming von CloudWatch Instanzprotokollen in Logs

Um das Streaming von Protokollen von den Amazon EC2 EC2-Instances Ihrer Umgebung zu CloudWatch Logs zu aktivieren, müssen Sie die folgenden Bedingungen erfüllen.

- Platform (Plattform) – Da diese Funktion nur in Plattformversionen zur Verfügung steht, die mit oder nach diesem Release veröffentlicht wurden, müssen Sie Ihre Umgebung aktualisieren, falls Sie eine frühere Plattformversion verwenden.
- Wenn Sie die AWSElasticBeanstalkWebTieroder von AWSElasticBeanstalkWorkerTierElastic Beanstalk verwaltete Richtlinie nicht in Ihrem Elastic Beanstalk Beanstalk-Instance-Profil haben, müssen Sie Folgendes zu Ihrem Profil hinzufügen, um diese Funktion zu aktivieren.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
  {
    "Effect": "Allow",
    "Action": [
      "logs:PutLogEvents",
      "logs:CreateLogStream"
    ],
    "Resource": [
    "*"
    ]
  }
  ]
}
```

So richtet Elastic Beanstalk Logs ein CloudWatch

Elastic Beanstalk installiert auf jeder erstellten Instance einen CloudWatch Log-Agent mit den Standardkonfigurationseinstellungen. Weitere Informationen finden Sie in der CloudWatch Logs Agent Reference.

Wenn Sie das Streaming von CloudWatch Instance-Logs zu Logs aktivieren, sendet Elastic Beanstalk Logdateien von den Instances Ihrer Umgebung an CloudWatch Logs. Unterschiedliche Plattformen streamen unterschiedliche Protokolle. In der folgenden Tabelle finden Sie die Protokolle nach Plattform sortiert.

Plattform/Plattformzweig	Protokolle
Docker/ Plattformzweig: Docker auf Amazon Linux 2 mit 64 Bit	• /-engine.log var/log/eb • /var/log/eb-hooks.log • /var/log/docker • /var/log/docker-events.log • /var/log/eb-docker/containers/eb-current-app/stdouterr.log • /.log var/log/nginx/access • /.log var/log/nginx/error
Docker/ Plattformzweig: ECS auf Amazon Linux 2 mit 64 Bit	• /-events.log var/log/docker • /var/log/eb-ecs-mgr.log • /var/log/eb-engine.log

Plattform/Plattformzweig	Protokolle
	• /var/log/eb-hooks.log • /var/log/ecs/ecs-agent.log • /var/log/ecs/ecs-init.log
Go .NET Core auf Linux Java/Plattformzweig: Corretto auf Amazon Linux 2 mit 64 Bit	• /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/web.stdout.log • /var/log/nginx/access.log • /.log var/log/nginx/error
Node.js Python	• /-engine.log var/log/eb • /var/log/eb-hooks.log • /var/log/web.stdout.log • var/log/httpd/access/_Protokoll • /_Protokoll var/log/httpd/error • /.log var/log/nginx/access • /.log var/log/nginx/error
Tomcat PHP	• /-engine.log var/log/eb • /var/log/eb-hooks.log • /var/log/httpd/access_Protokoll • /_Protokoll var/log/httpd/error • /.log var/log/nginx/access • /.log var/log/nginx/error
.NET mit Windows Server	• C:\inetpub\logs\\ W3LogFiles\ u_ex*.log SVC1 • C:\Program Dateien\ Amazon\ElasticBeanstalk\ logs\ AWSDeployment .log • C:\Program Dateien\ Amazon\ElasticBeanstalk\ logs\ Hooks.log

Plattform/Plattformzweig	Protokolle
Ruby	• /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/puma/puma.log • /.stdout.log var/log/web • /var/log/nginx/access.log • /.log var/log/nginx/error

Protokolldateien auf Amazon Linux AMI-Plattformen

> **Note**
>
> Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Weitere Informationen zur Migration zu einem aktuellen und vollständig unterstützten Plattformzweig für Amazon Linux 2023 finden Sie unter Migrieren der Elastic-Beanstalk-Linux-Anwendung zu Amazon Linux 2023 oder Amazon Linux 2.

In der folgenden Tabelle werden die Protokolldateien nach Plattform aufgelistet, die aus Instances in Plattformverzweigungen auf der Basis von Amazon Linux AMI (vor Amazon Linux 2) gestreamt werden.

Plattform/Plattformzweig	Protokolle
Docker/ Plattformzweig: Docker auf Amazon Linux mit 64 Bit	• /-activity.log var/log/eb • /var/log/nginx/error.log • /-events.log var/log/docker • /var/log/docker • /var/log/nginx/access.log • /.log var/log/eb-docker/containers/eb-current-app/stdouterr
Docker/	• /-activity.log var/log/eb • /var/log/ecs/ecs-init.log

Plattform/Plattformzweig	Protokolle
Plattformzweig: Multi-Container-Docker auf Amazon Linux mit 64 Bit	• /var/log/eb-ecs-mgr.log • /var/log/ecs/ecs-agent.log • /var/log/docker-events.log
Glassfish (Docker vorkonfiguriert)	• /var/log/eb-activity.log • /var/log/nginx/error.log • /-events.log var/log/docker • /var/log/docker • /var/log/nginx/access.log
Go	• /-activity.log var/log/eb • /var/log/nginx/error.log • /.log var/log/nginx/access
Java/ Plattformzweig: Java 8 auf Amazon Linux mit 64 Bit Plattformzweig: Java 7 auf Amazon Linux mit 64 Bit	• /-activity.log var/log/eb • /var/log/nginx/access.log • /.log var/log/nginx/error • /-1.fehler.log var/log/web • var/log/web/-1.log
Tomcat	• /var/log/eb-activity.log • /var/log/httpd/error_Protokoll • /_Protokoll var/log/httpd/access • /_Protokoll var/log/nginx/error • /_Protokoll var/log/nginx/access

Plattform/Plattformzweig	Protokolle
Node.js	• /-activity.log var/log/eb • /var/log/nodejs/nodejs.log • /.log var/log/nginx/error • /.log var/log/nginx/access • /.log var/log/httpd/error • /.log var/log/httpd/access
PHP	• /-activity.log var/log/eb • /var/log/httpd/error_Protokoll • /_Protokoll var/log/httpd/access
Python	• /-activity.log var/log/eb • /var/log/httpd/error_Protokoll • /_Protokoll var/log/httpd/access • /.log opt/python/log/supervisord
Ruby/ Plattformzweig: Puma mit Ruby auf Amazon Linux mit 64 Bit	• /-activity.log var/log/eb • /var/log/nginx/error.log • /.log var/log/puma/puma • /.log var/log/nginx/access
Ruby/ Plattformzweig: Passenger mit Ruby auf Amazon Linux mit 64 Bit	• /-activity.log var/log/eb • /var/app/support/logs/passenger.log • /.log var/app/support/logs/access • /.log var/app/support/logs/error

Elastic Beanstalk konfiguriert Protokollgruppen in CloudWatch Logs für die verschiedenen Protokolldateien, die gestreamt werden. Um bestimmte Protokolldateien aus CloudWatch Logs abzurufen, müssen Sie den Namen der entsprechenden Protokollgruppe kennen. Das Namensschema der Protokollgruppe hängt vom Betriebssystem der Plattform ab.

Für Linux-Plattformen stellen Sie der Protokolldatei auf der Instance das Präfix `/aws/elasticbeanstalk/`*`environment_name`* voraus, um den Protokollgruppennamen zu erhalten. Um beispielsweise die Datei `/var/log/nginx/error.log` abzurufen, geben Sie die Protokollgruppe `/aws/elasticbeanstalk/`*`environment_name`*`/var/log/nginx/error.log` an.

Für Windows-Plattformen finden Sie in der folgenden Tabelle die Protokollgruppen für alle Protokolldateien.

Protokolldatei auf der Instance	Protokollgruppe
`C:\Program Files\Amazon\ElasticBeanstalk\logs\AWSDeployment.log`	`/aws/elasticbeanstalk/<environment-name>/EBDeploy-Log`
`C:\Program Files\Amazon\ElasticBeanstalk\logs\Hooks.log`	`/aws/elasticbeanstalk/<environment-name>/EBHooks-Log`
`C:\inetpub\logs\LogFiles` (das gesamte Verzeichnis)	`/aws/elasticbeanstalk/<environment-name>/IIS-Log`

Die Streaming-Instanz protokolliert in CloudWatch Logs

Sie können das Streaming von CloudWatch Instance-Logs in Logs über die Elastic Beanstalk Beanstalk-Konsole, die EB-CLI oder über Konfigurationsoptionen aktivieren.

Bevor Sie es aktivieren, richten Sie IAM-Berechtigungen für die Verwendung mit dem CloudWatch Logs-Agenten ein. Sie können die folgende benutzerdefinierte Richtlinien an das Instance-Profil anhängen, das Sie Ihrer Umgebung zuweisen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Streaming von Instance-Protokollen mit der Elastic Beanstalk-Konsole

Um Instanzprotokolle in Logs zu CloudWatch streamen

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Gehen Sie unter Instance-Log-Streaming zu CloudWatch Logs wie folgt vor:
   - Aktivieren Sie Log streaming (Protokoll-Streaming).
   - Setzen Sie Retention (Aufbewahrungsdauer) auf die Anzahl an Tagen, wie lange die Protokolle aufbewahrt werden sollen.
   - Wählen Sie die Einstellung Lifecycle (Lebenszyklus), die bestimmt, ob die Protokolle gespeichert werden, nachdem die Umgebung beendet wurde.
6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Nachdem Sie das Protokoll-Streaming aktiviert haben, können Sie zur Konfigurationskategorie oder -seite Software zurückkehren und dort den Link Log Groups (Protokollgruppen) suchen. Klicken Sie auf diesen Link, um Ihre Logs in der CloudWatch Konsole zu sehen.

Streaming von Instance-Protokollen mit der EB-CLI

Verwenden Sie den eb logs Befehl, um das Streaming von CloudWatch Instanzprotokollen zu Logs mithilfe der EB-CLI zu aktivieren.

```
$ eb logs --cloudwatch-logs enable
```

Sie können ihn auch verwendeneb logs, um Logs aus CloudWatch Logs abzurufen. Sie können alle Instance-Protokolle der Umgebung abrufen, oder die vielen Optionen des Befehls verwenden, um Untergruppen abzurufender Protokolle anzugeben. Der folgende Befehl beispielsweise ruft den vollständigen Satz an Instance-Protokollen für Ihre Umgebung auf und speichert diese in ein Verzeichnis unter `.elasticbeanstalk/logs`.

```
$ eb logs --all
```

Mit der Option `--log-group` können Sie Instance-Protokolle einer bestimmten Protokollgruppe abrufen, die einer bestimmten Protokolldatei einer Instance entspricht. Dazu müssen Sie den Namen der Protokollgruppe kennen, die der abzurufenden Protokolldatei entspricht. Sie finden diese Informationen in So richtet Elastic Beanstalk Logs ein CloudWatch.

Streaming von Instance-Protokollen mit Konfigurationsdateien

Wenn Sie eine Umgebung erstellen oder aktualisieren, können Sie eine Konfigurationsdatei verwenden, um das Streaming von CloudWatch Instanzprotokollen in Logs einzurichten und zu konfigurieren. Die folgende Beispiel-Konfigurationsdatei ermöglicht ein standardmäßiges Streaming der Instance-Protokolle. Elastic Beanstalk streamt den Standardsatz an Protokolldateien für die Plattform Ihrer Umgebung. Um das Beispiel zu verwenden, kopieren Sie den Text in eine Datei mit der Erweiterung `.config` im Verzeichnis `.ebextensions` auf der obersten Ebene Ihres Anwendungsquellpakets.

```
option_settings:
  - namespace: aws:elasticbeanstalk:cloudwatch:logs
    option_name: StreamLogs
    value: true
```

Benutzerdefiniertes Streaming von Protokolldateien

Die Elastic Beanstalk Beanstalk-Integration mit CloudWatch Logs unterstützt nicht direkt das Streaming von benutzerdefinierten Protokolldateien, die Ihre Anwendung generiert. Um

benutzerdefinierte Logs zu streamen, verwenden Sie eine Konfigurationsdatei, um den CloudWatch Agenten direkt zu installieren und die Dateien für die Übertragung zu konfigurieren. Eine Beispielkonfigurationsdatei finden Sie unter `logs-streamtocloudwatch-linux.config`.

> **Note**
>
> Das Beispiel funktioniert nicht auf der Windows-Plattform.

Weitere Informationen zur Konfiguration von CloudWatch Logs finden Sie in der Referenz zur CloudWatch Agentenkonfigurationsdatei im CloudWatch Amazon-Benutzerhandbuch.

Fehlerbehebung bei der Integration CloudWatch von Logs

> **Testen Sie Amazon Q Developer CLI für KI-gestützte Problembehebung**
>
> Amazon Q Developer CLI kann Ihnen helfen, Umgebungsprobleme schnell zu beheben. Die Q CLI bietet Lösungen, indem sie den Umgebungsstatus überprüft, Ereignisse überprüft, Protokolle analysiert und klärende Fragen stellt. Weitere Informationen und detaillierte Anleitungen finden Sie in den Blogs unter Troubleshooting Elastic Beanstalk Environments with Amazon Q Developer CLI. AWS

Die Protokolle der Umgebungsinstanzen konnten nicht gefunden werden

Wenn Sie einige der erwarteten Instanzprotokolle der Umgebung nicht unter Logs finden können, untersuchen Sie die folgenden häufig auftretenden Probleme: CloudWatch

- Ihre IAM-Rolle verfügt nicht über die erforderlichen IAM-Berechtigungen.
- Sie haben Ihre Umgebung in einer Umgebung gestartet AWS-Region , die CloudWatch Logs nicht unterstützt.
- Eine Ihrer benutzerdefinierten Protokolldateien existiert nicht unter dem von Ihnen angegebenen Pfad.

Anwendungsprotokolle fehlen oder treten nur sporadisch auf

Wenn die Logs Ihrer Elastic Beanstalk Beanstalk-Anwendung (`/var/log/web.stdout.log`) zu fehlen scheinen oder nur sporadisch erscheinen, kann das an den Standardeinstellungen

für die Ratenbegrenzung in rsyslog und journald liegen. Die vollständige Deaktivierung der Ratenbegrenzung kann dieses Problem zwar beheben, wird jedoch nicht empfohlen, da dies zu übermäßiger Festplattennutzung, potenziellem Denial-of-Service oder einer Verschlechterung der Systemleistung bei unerwarteten Log-Bursts führen kann. Stattdessen können Sie die Ratenbegrenzungen wie folgt anpassen. `.ebextensions configuration` Diese Konfiguration erhöht das Intervall für die Ratenbegrenzung bei höheren Burst-Grenzwerten auf 600 Sekunden und sorgt so für ein ausgewogenes Verhältnis zwischen ordnungsgemäßer Protokollierung und Systemschutz.

Probleme mit Drosselung

Wenn ein Elastic Beanstalk Beanstalk-Vorgang, der gleichzeitig eine große Anzahl von Instances startet`Error: fail to create log stream: ThrottlingException: Rate exceeded`, eine Meldung wie „Drosselung“ aufgrund zu vieler API-Aufrufe zurückgibt. CloudWatch

Um das Drosselungsproblem zu beheben, führen Sie eine der folgenden Maßnahmen durch:

- Verwenden Sie bei fortlaufenden Bereitstellungen eine kleinere Batchgröße, um die Anzahl gleichzeitiger Updates zu reduzieren.
- Beantragen Sie eine Erhöhung des Servicekontingents für Transaktionen pro Sekunde (TPS) für Ihr AWS Konto. CreateLogStream Weitere Informationen finden Sie unter CloudWatch Logs-Kontingente und Verwaltung Ihrer CloudWatch Logs-Dienstkontingente im Amazon CloudWatch Logs-Benutzerhandbuch.

Zustandsinformationen der Elastic Beanstalk Beanstalk-Umgebung an Amazon Logs streamen CloudWatch

Wenn Sie die erweiterte Statusberichterstattung für Ihre Umgebung aktivieren, können Sie die Umgebung so konfigurieren, dass Gesundheitsinformationen in Logs gestreamt werden CloudWatch . Dieses Streaming ist unabhängig vom EC2 Amazon-Instance-Log-Streaming. Dieses Thema beschreibt Informationen zum Streaming des Umgebungszustands. Weitere Informationen zum Streaming von Instance-Protokollen finden Sie unter Elastic Beanstalk mit Amazon Logs verwenden CloudWatch.

Wenn Sie Environment Health Streaming konfigurieren, erstellt Elastic Beanstalk eine CloudWatch Logs-Protokollgruppe für den Zustand der Umgebung. Der Name der Protokollgruppe ist `/aws/elasticbeanstalk/`*`environment-name`*`/environment-health.log`. Innerhalb dieser

Protokollgruppe erstellt Elastic Beanstalk Protokollstreams mit dem Namen *YYYY-MM-DD*#*<hash-suffix>* (möglicherweise mehr als einen Protokollstream pro Datum).

Wenn sich der Integritätsstatus der Umgebung ändert, fügt Elastic Beanstalk einen Datensatz in den Integritätsprotokoll-Stream ein. Der Datensatz stellt den Zustandsübergang dar – den neuen Status und eine Beschreibung der Ursache für die Änderung. Beispielsweise könnte sich der Zustand einer Umgebung auf "Severe“ (Schwerwiegend) ändern, weil der Load Balancer ausgefallen ist. Eine Beschreibung der erweiterten Zustandsstatus finden Sie unter Farben und Status in Zustandsangaben.

Voraussetzungen für das Streaming der Umgebungsintegrität in Logs CloudWatch

Um das Streaming der Umgebungsintegrität in CloudWatch Logs zu aktivieren, müssen Sie die folgenden Bedingungen erfüllen:

- Platform (Plattform) – Sie müssen eine Plattformversion verwenden, die erweiterte Integritätsberichte unterstützt.
- Permissions (Berechtigungen) – Sie müssen bestimmte Berechtigungen im Zusammenhang mit der Protokollierung an Elastic Beanstalk erteilen, sodass es in Ihrem Namen agieren kann, um die Integritätsinformationen für Ihre Umgebung zu streamen. Wenn Ihre Umgebung keine Servicerolle verwendet, die Elastic Beanstalk dafür erstellt hat (`aws-elasticbeanstalk-service-role`), und auch nicht die serviceverknüpfte Rolle Ihres Kontos (`AWSServiceRoleForElasticBeanstalk`), stellen Sie sicher, dass Sie Ihrer benutzerdefinierten Servicerolle die folgenden Berechtigungen hinzufügen.

```
{
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogStreams",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk/*:log-stream:*"
}
```

Zustandsprotokolle der Umgebung in CloudWatch Protokolle streamen

Sie können das Streaming des Umgebungszustands in CloudWatch Logs mithilfe der Elastic Beanstalk Beanstalk-Konsole, der EB-CLI oder mit Konfigurationsoptionen aktivieren.

Streamen von Umgebungsintegritätsprotokollen mit der Elastic Beanstalk-Konsole

Um Umgebungszustandsprotokolle in Logs zu streamen CloudWatch

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).
5. Stellen Sie sicher, dass unter Health reporting (Zustandsberichte) das System auf Enhanced (Erweitert) gesetzt ist.
6. Under Health Event wird in CloudWatch Logs gestreamt
   - Aktivieren Sie Log streaming (Protokoll-Streaming).
   - Setzen Sie Retention (Aufbewahrungsdauer) auf die Anzahl an Tagen, wie lange die Protokolle aufbewahrt werden sollen.
   - Wählen Sie die Einstellung Lifecycle (Lebenszyklus), die bestimmt, ob die Protokolle gespeichert werden, nachdem die Umgebung beendet wurde.
7. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Nachdem Sie das Protokoll-Streaming aktiviert haben, können Sie zur Konfigurationskategorie oder -seite Monitoring (Überwachung) zurückkehren und dort den Link Log Group (Protokollgruppe) suchen. Klicken Sie auf diesen Link, um die Integritätsprotokolle Ihrer Umgebung in der CloudWatch Konsole anzuzeigen.

Streamen von Umgebungszustandsprotokollen mit der EB-CLI

Verwenden Sie den eb logs Befehl, um das Streaming von CloudWatch Umgebungsintegritätsprotokollen in Logs mithilfe der EB-CLI zu aktivieren.

```
$ eb logs --cloudwatch-logs enable --cloudwatch-log-source environment-health
```

Sie können ihn auch verwendeneb logs, um CloudWatch Protokolle aus Logs abzurufen. Der folgende Befehl beispielsweise ruft alle Zustandsprotokolle für Ihre Umgebung auf und speichert diese in ein Verzeichnis unter `.elasticbeanstalk/logs`.

```
$ eb logs --all --cloudwatch-log-source environment-health
```

Streamen von Umgebungszustandsprotokollen mit Konfigurationsdateien

Wenn Sie eine Umgebung erstellen oder aktualisieren, können Sie eine Konfigurationsdatei verwenden, um das Streaming von CloudWatch Umgebungsintegritätsprotokollen in Logs einzurichten und zu konfigurieren. Um das nachfolgende Beispiel zu verwenden, kopieren Sie den Text in eine Datei mit der Dateinamenerweiterung `.config` im Verzeichnis `.ebextensions` auf der obersten Ebene Ihres Anwendungs-Quell-Bundles. Das Beispiel konfiguriert Elastic Beanstalk so, dass das Streaming der Umgebungsintegritätsprotokolle aktiviert ist, dass die Protokolle nach dem Beenden der Umgebung beibehalten und für 30 Tage gespeichert werden.

Example Konfigurationsdatei für das Streaming des Zustands

```
#############################################################################
##  Sets up Elastic Beanstalk to stream environment health information
##  to Amazon CloudWatch Logs.
##  Works only for environments that have enhanced health reporting enabled.
#############################################################################

option_settings:
  aws:elasticbeanstalk:cloudwatch:logs:health:
    HealthStreamingEnabled: true
    ### Settings below this line are optional.
    # DeleteOnTerminate: Delete the log group when the environment is
    # terminated. Default is false. If false, the health data is kept
    # RetentionInDays days.
    DeleteOnTerminate: false
    # RetentionInDays: The number of days to keep the archived health data
    # before it expires, if DeleteOnTerminate isn't set. Default is 7 days.
    RetentionInDays: 30
```

Standardwerte und gültige Werte für die Option finden Sie unter `aws:elasticbeanstalk:cloudwatch:logs:health`.

Elastic Beanstalk mit Amazon verwenden EventBridge

Mit Amazon können Sie ereignisgesteuerte Regeln einrichten EventBridge, die Ihre Elastic Beanstalk Beanstalk-Ressourcen überwachen und Zielaktionen initiieren, die andere Services nutzen. AWS Sie können beispielsweise eine Regel für das Versenden von E-Mail-Benachrichtigungen festlegen, indem Sie ein Amazon-SNS-Thema signalisieren, wenn sich der Status einer Produktionsumgebung in einen Warnung ändert. Oder du kannst eine Lambda-Funktion einrichten, um eine Benachrichtigung an Slack zu übergeben, wenn sich der Zustand deiner Umgebung in den Status Schwach oder Schwerwiegend ändert.

Sie können in Amazon Regeln erstellen EventBridge , um auf jedes der folgenden Elastic Beanstalk Beanstalk-Ereignisse zu reagieren:

- Statusänderungen für Umgebungsvorgänge (einschließlich Erstellen, Aktualisieren und Beenden von Vorgängen). Das Ereignis gibt an, ob die Statusänderung begonnen hat, erfolgreich war oder fehlgeschlagen ist.
- Statusänderungen für andere Ressourcen. Neben Umgebungen umfassen andere Ressourcen, die überwacht werden, Load Balancer, Auto Scaling-Gruppen und Instances.
- Integritätsübergang für Umgebungen. Das Ereignis gibt an, wo die Umgebungsintegrität von einem Integritätsstatus zu einem anderen übergegangen ist.
- Statusänderung für verwaltete Updates. Das Ereignis gibt an, ob die Statusänderung begonnen hat, erfolgreich war oder fehlgeschlagen ist.

Um bestimmte Elastic Beanstalk Beanstalk-Ereignisse zu erfassen, an denen Sie interessiert sind, definieren Sie ereignisspezifische Muster, anhand derer die Ereignisse EventBridge erkannt werden können. Ereignismuster haben dieselbe Struktur wie die Ereignisse, mit denen sie übereinstimmen. Das Muster zitiert die Felder, die Sie abgleichen möchten, und liefert die Werte, nach denen Sie suchen. Ereignisse werden auf bestmögliche Weise ausgegeben. Sie werden unter normalen Betriebsbedingungen nahezu EventBridge in Echtzeit von Elastic Beanstalk an übermittelt. Es können jedoch Situationen auftreten, die die Durchführung eines Ereignisses verzögern oder verhindern können.

Eine Liste der Felder, die in Elastic-Beanstalk-Ereignissen enthalten sind, und deren möglichen Zeichenfolgenwerten finden Sie unter Elastic Beanstalk-Ereignisfeldzuordnung. Informationen darüber, wie EventBridge Regeln mit Ereignismustern funktionieren, finden Sie unter Ereignisse und Ereignismuster in. EventBridge

Überwachen Sie eine Elastic Beanstalk Beanstalk-Ressource mit EventBridge

Mit können Sie Regeln erstellen EventBridge, die Aktionen definieren, die ergriffen werden sollen, wenn Elastic Beanstalk Ereignisse für seine Ressourcen ausgibt. Sie können beispielsweise eine Regel erstellen, die Ihnen eine E-Mail-Nachricht sendet, wenn sich der Status einer Umgebung ändert.

Die EventBridge Konsole verfügt über eine vordefinierte Musteroption zum Erstellen von Elastic Beanstalk Beanstalk-Event-Pattern. Wenn Sie diese Option in der EventBridge Konsole auswählen, wenn Sie eine Regel erstellen, können Sie schnell ein Elastic Beanstalk Beanstalk-Event-Pattern erstellen. Sie müssen nur die Ereignisfelder und -werte auswählen. Während Sie eine Auswahl treffen, erstellt die Konsole das Ereignismuster und zeigt es an. Alternativ können Sie das von Ihnen erstellte Ereignismuster manuell bearbeiten und als benutzerdefiniertes Muster speichern. Die Konsole gibt Ihnen außerdem die Option, ein detailliertes Beispielereignis anzuzeigen, das Sie kopieren und in das von Ihnen errichte Ereignismuster einfügen können.

Wenn Sie es vorziehen, ein Ereignismuster einzugeben oder zu kopieren und in die EventBridge Konsole einzufügen, können Sie die Option Benutzerdefiniertes Muster in der Konsole verwenden. Auf diese Weise müssen Sie nicht die Schritte zur Auswahl der zuvor beschriebenen Felder und Werte durchlaufen. Dieses Thema bietet Beispiele für Ereignisübereinstimmungsmuster und Elastic-Beanstalk-Ereignisse, die Sie verwenden können.

So erstellen Sie eine Regel für ein Ressourcenereignis

1. Melden Sie sich AWS mit einem Konto an, das über Nutzungsberechtigungen verfügt, EventBridge und Elastic Beanstalk.
2. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
3. Wählen Sie im Navigationsbereich Regeln aus.
4. Wählen Sie Create rule (Regel erstellen) aus.
5. Geben Sie für die Regel einen Name (Namen) und optional eine Beschreibung ein.
6. Bei Event bus (Ereignisbus) wählen Sie default (Standard) aus. Wenn ein AWS Service in Ihrem Konto ein Ereignis ausgibt, wird dieses immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.
7. Bei Regeltyp wählen Sie Regel mit einem Ereignismuster aus.
8. Wählen Sie Weiter aus.

9. Wählen Sie als Ereignisquelle AWS Ereignisse oder EventBridge Partnerereignisse aus.
10. (Optionl) Wählen Sie für Beispielereignis AWS -Ereignisse aus. Geben Sie Elastic Beanstalk im Suchfeld ein. Dadurch erhalten Sie eine Liste von Elastic Beanstalk-Beispielereignissen, aus denen Sie auswählen können, ob Sie sie anzeigen möchten. In diesem Schritt wird lediglich ein Beispielereignis angezeigt, auf das Sie verweisen können. Es hat keinen Einfluss auf das Ergebnis der Regelerstellung. Der Abschnitt Beispiel für Elastic Beanstalk-Ereignisse weiter unten in diesem Thema enthält Beispiele für die gleiche Art von Ereignissen.
11. Wählen Sie im Abschnitt Ereignismuster die Option Ereignismusterformular aus.

> **Note**
>
> Wenn Sie bereits Text für ein Ereignismuster haben und die EventBridge Konsole nicht benötigen, um ihn für Sie zu erstellen, wählen Sie Benutzerdefiniertes Muster (JSON-Editor). Sie können dann entweder manuell Text in das Feld Ereignismuster eingeben oder kopieren und einfügen. Klicken Sie auf Weiter, und fahren Sie mit dem Schritt zur Eingabe eines Ziels fort.

12. Als Event source (Ereignisquelle) wählen Sie AWS -Services aus.
13. Wählen Sie als AWS -Service Elastic Beanstalk aus.
14. Wählen Sie als Ereignistyp die Option Statusänderung aus.
15. In diesem Schritt wird beschrieben, wie Sie mit den Ereignisfeldern Detailtyp, Statusund Schweregrad für Elastic Beanstalk arbeiten können. Wenn Sie diese Felder und die übereinstimmenden Werte auswählen, erstellt die Konsole das Ereignismuster und zeigt es an.

    - Wenn Sie nur einen Wert für Bestimmte Detailtypen auswählen, können Sie einen oder mehrere Werte für das nächste Feld in der Hierarchie auswählen.
    - Wenn Sie mehr als einen Wert für Spezifische Detailtypen wählen, wählen Sie keine spezifischen Werte für die nächsten Felder in der Hierarchie aus. Dies verhindert mehrdeutige Übereinstimmungslogik über Felder in Ihrem Ereignismuster hinweg.

    Das Ereignisfeld Umgebung wird von dieser Hierarchie nicht beeinflusst und wird daher wie im nächsten Schritt beschrieben angezeigt.
16. Wählen Sie als Umgebung Beliebige Umgebung oder Bestimmte Umgebung(en) aus.

    - Wenn Sie Bestimmte Umgebung (en) auswählen, können Sie eine oder mehrere Umgebungen aus der Dropdownliste auswählen. EventBridge fügt alle Umgebungen hinzu, die Sie in der

Liste EnvironmentName[] im Detailbereich des Ereignismusters auswählen. Dann filtert Ihre Regel alle Ereignisse so, dass sie nur die von Ihnen ausgewählten spezifischen Umgebungen enthalten.

- Wenn Sie Beliebige Umgebung auswählen, werden Ihrem Ereignismuster keine Umgebungen hinzugefügt. Aus diesem Grund filtert Ihre Regel keines der Elastic-Beanstalk-Ereignisse basierend auf der Umgebung.

17. Wählen Sie Weiter aus.
18. Bei Zieltypen wählen Sie AWS -Service aus.
19. Wählen Sie unter Ziel auswählen die durchzuführende Zielaktion aus, wenn ein Ereignis zur Änderung des Ressourcenstatus von Elastic Beanstalk empfangen wird.

    Sie können beispielsweise ein Amazon Simple Notification Service (SNS)-Thema verwenden, um eine E-Mail oder SMS zu senden, wenn ein Ereignis eintritt. Dazu müssen Sie mit der Amazon SNS-Konsole ein Amazon SNS-Thema erstellen. Weitere Informationen finden Sie unter Verwenden von Amazon SNS für Benutzerbenachrichtigungen.

    > **Important**
    >
    > Einige Zielaktionen erfordern möglicherweise die Nutzung anderer Services und verursachen zusätzliche Gebühren, wie z. B. der Amazon-SNS- oder Lambda-Service. Weitere Informationen zur AWS Preisgestaltung finden Sie unterhttps://aws.amazon.com/pricing/. Einige Dienste sind Teil des AWS kostenlosen Nutzungskontingents. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter https://aws.amazon.com/free/.

20. (Optional) Wählen Sie Weiteres Ziel hinzufügen aus, um eine zusätzliche Zielaktion für die Ereignisregel anzugeben.
21. Wählen Sie Weiter aus.
22. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter EventBridge Amazon-Tags im EventBridge Amazon-Benutzerhandbuch.
23. Wählen Sie Weiter aus.
24. Überprüfen Sie die Details der Regel und wählen Sie dann Regel erstellen aus.

Beispiel für Elastic-Beanstalk-Ereignisse

Ereignismuster haben dieselbe Struktur wie die Ereignisse, mit denen sie übereinstimmen. Das Muster zitiert die Felder, die Sie abgleichen möchten, und liefert die Werte, nach denen Sie suchen.

- Änderung des Zustands für Umgebungen

```
{
   "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Health status change"
    ]
}
```

- Änderung des Zustands für die folgenden Umgebungen: `myEnvironment1` und `myEnvironment2`. Dieses Ereignismuster filtert nach diesen beiden spezifischen Umgebungen, während das vorherige Beispiel für die Änderung des Zustands, das nicht filtert, Ereignisse für alle Umgebungen sendet.

```
{"source": [
    "aws.elasticbeanstalk"
    ],
    "detail-type": [
        "Health status change"
    ],
    "detail": {
        "EnvironmentName": [
            "myEnvironment1",
            "myEnvironment2"
        ]
    }
}
```

- Änderung des Elastic Beanstalk-Ressourcenstatus für alle Umgebungen

```
{
  "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
```

```
    "Elastic Beanstalk resource status change"
    ]
}
```

- Änderung des Elastic Beanstalk-Ressourcenstatus mit `Status` Umgebungsaktualisierung fehlgeschlagen und `Severity` ERROR für die folgenden Umgebungen: `myEnvironment1` und `myEnvironment2`

```
{"source": [
    "aws.elasticbeanstalk"
    ],
    "detail-type": [
        "Elastic Beanstalk resource status change"
    ],
    "detail": {
        "Status": [
            "Environment update failed"
            ],
        "Severity": [
            "ERROR"
            ],
        "EnvironmentName": [
            "myEnvironment1",
            "myEnvironment2"
        ]
    }
}
```

- Statusänderung anderer Ressourcen für Load Balancer, Auto Scaling-Gruppen und Instances

```
{
   "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Other resource status change"
    ]
}
```

- Statusänderung verwalteter Updates für alle Umgebungen

```
{
   "source": [
```

```
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Managed update status change"
    ]
}
```

- So erfassen Sie alle Ereignisse aus Elastic Beanstalk (schließen Sie den Abschnitt `detail-type` aus)

```
{
  "source": [
    "aws.elasticbeanstalk"
  ]
}
```

Beispiel für Elastic Beanstalk-Ereignisse

Es folgt ein Beispiel eines Elastic Beanstalk-Ereignisses für eine Statusänderung der Ressourcen:

```
{
   "version":"0",
   "id":"1234a678-1b23-c123-12fd3f456e78",
   "detail-type":"Elastic Beanstalk resource status change",
   "source":"aws.elasticbeanstalk",
   "account":"111122223333",
   "time":"2020-11-03T00:31:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:was:elasticbeanstalk:us-east-1:111122223333:environment/myApplication/
myEnvironment"
   ],
   "detail":{
      "Status":"Environment creation started",
      "EventDate":1604363513951,
      "ApplicationName":"myApplication",
      "Message":"createEnvironment is starting.",
      "EnvironmentName":"myEnvironment",
      "Severity":"INFO"
   }
}
```

Es folgt ein Beispiel eines Elastic Beanstalk-Ereignisses für eine Änderung des Integritätsstatus:

```
{
   "version":"0",
   "id":"1234a678-1b23-c123-12fd3f456e78",
   "detail-type":"Health status change",
   "source":"aws.elasticbeanstalk",
   "account":"111122223333",
   "time":"2020-11-03T00:34:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:was:elasticbeanstalk:us-east-1:111122223333:environment/myApplication/
myEnvironment"
   ],
   "detail":{
      "Status":"Environment health changed",
      "EventDate":1604363687870,
      "ApplicationName":"myApplication",
      "Message":"Environment health has transitioned from Pending to Ok. Initialization
 completed 1 second ago and took 2 minutes.",
      "EnvironmentName":"myEnvironment",
      "Severity":"INFO"
   }
}
```

Elastic Beanstalk-Ereignisfeldzuordnung

In der folgenden Tabelle werden Elastic Beanstalk Beanstalk-Ereignisfelder und ihre möglichen Zeichenkettenwerte dem EventBridge `detail-type` Feld zugeordnet. Weitere Informationen darüber, wie mit Ereignismustern für einen Service EventBridge funktioniert, finden Sie unter Ereignisse und Ereignismuster in. EventBridge

EventBridge Detailtyp des Felds	Elastic Beanstalk-Feld Status	Elastic Beanstalk-Feld Severity (Schweregrad)	Elastic Beanstalk-Feld Message (Nachricht)
Statusänderung von Elastic Beanstalk-Ressourcen	Umgebungserstellung gestartet	INFO	createEnvironment wird gestartet.
	Umgebungserstellung erfolgreich	INFO	createEnvironment wurde erfolgreich abgeschlossen.
	Umgebungserstellung erfolgreich	INFO	Gestartete Umgebung: <Environment Name>. Es gab jedoch Probleme während des Starts. Einzelheiten finden Sie im Ereignisprotokoll.
	Umgebungserstellung fehlgeschlagen	ERROR	Die Umgebung konnte nicht gestartet werden.
	Umgebungs-Update gestartet	INFO	Das Umgebungs-Update wird gestartet.
	Umgebungs-Update erfolgreich	INFO	Das Umgebungs-Update wurde erfolgreich abgeschlossen.
	Umgebungs-Update fehlgeschlagen	ERROR	Die Konfiguration konnte nicht bereitgestellt werden.

EventBridge Detailtyp des Felds	Elastic Beanstalk-Feld Status	Elastic Beanstalk-Feld Severity (Schweregrad)	Elastic Beanstalk-Feld Message (Nachricht)
	Umgebungsbeendigung gestartet	INFO	terminateEnvironment wird gestartet.
	Umgebungsbeendigung erfolgreich	INFO	terminateEnvironment wurde erfolgreich abgeschlossen.
	Umgebungsbeendigung fehlgeschlagen	INFO	Der Schritt für die Umgebungsbeendigung ist fehlgeschlagen, da mindestens einer der Workflows zur Umgebungsbeendigung fehlgeschlagen ist.
Statusänderung anderer Ressourcen	Auto Scaling-Gruppe erstellt	INFO	createEnvironment wird gestartet.
	Auto Scaling-Gruppe gelöscht	INFO	createEnvironment wird gestartet.
	Instance hinzugefügt	INFO	Die Instance [i-123456789a12b1234] wurde Ihrer Umgebung hinzugefügt.
	Instance entfernt	INFO	Die Instance [i-123456789a12b1234] wurde aus Ihrer Umgebung entfernt.

EventBridge Detailtyp des Felds	Elastic Beanstalk-Feld Status	Elastic Beanstalk-Feld Severity (Schweregrad)	Elastic Beanstalk-Feld Message (Nachricht)
	Load Balancer erstellt	INFO	Erstellter Load Balancer namens: <LB Name>
	Load Balancer gelöscht	INFO	Gelöschter Load Balancer namens: <LB Name>
Statusänderung der Integrität	Umgebungsintegrität geändert	INFO/WARN	Die Umgebungsintegrität hat zu <healthStatus> gewechselt.
	Umgebungsintegrität geändert	INFO/WARN	Die Umgebungsintegrität hat von <healthStatus> zu <healthStatus> gewechselt.
Statusänderung verwalteter Updates	Verwaltetes Update gestartet	INFO	Das verwaltete Plattform-Update wird durchgeführt.
	Verwaltetes Update fehlgeschlagen	INFO	Das verwaltete Update ist fehlgeschlagen, Wiederholung in %s Minuten.

Elastic Beanstalk Beanstalk-Ressourcen finden und verfolgen mit AWS Config

AWS Configbietet eine detaillierte Ansicht der Konfiguration der AWS Ressourcen in Ihrem Konto. AWS Sie können Ressourcenbeziehungen betrachten, einen Verlauf der Konfigurationsänderungen

anzeigen und feststellen, wie sich Beziehungen und Konfigurationen im Lauf der Zeit ändern. Sie können AWS Config damit Regeln definieren, mit denen Ressourcenkonfigurationen im Hinblick auf Datenkonformität bewertet werden.

Verschiedene Elastic Beanstalk Beanstalk-Ressourcentypen sind integriert in: AWS Config

- Anwendungen
- Anwendungsversionen
- Umgebungen

Der folgende Abschnitt zeigt, wie Sie die Konfiguration für AWS Config die Aufzeichnung von Ressourcen dieser Typen konfigurieren.

Weitere Informationen zu AWS Config finden Sie im AWS Config Entwicklerhandbuch. Preisinformationen finden Sie auf der AWS Config Config-Preisinformationsseite.

Einrichten AWS Config

Informationen zur ersten Einrichtung AWS Config finden Sie in den folgenden Themen im AWS Config Entwicklerhandbuch.

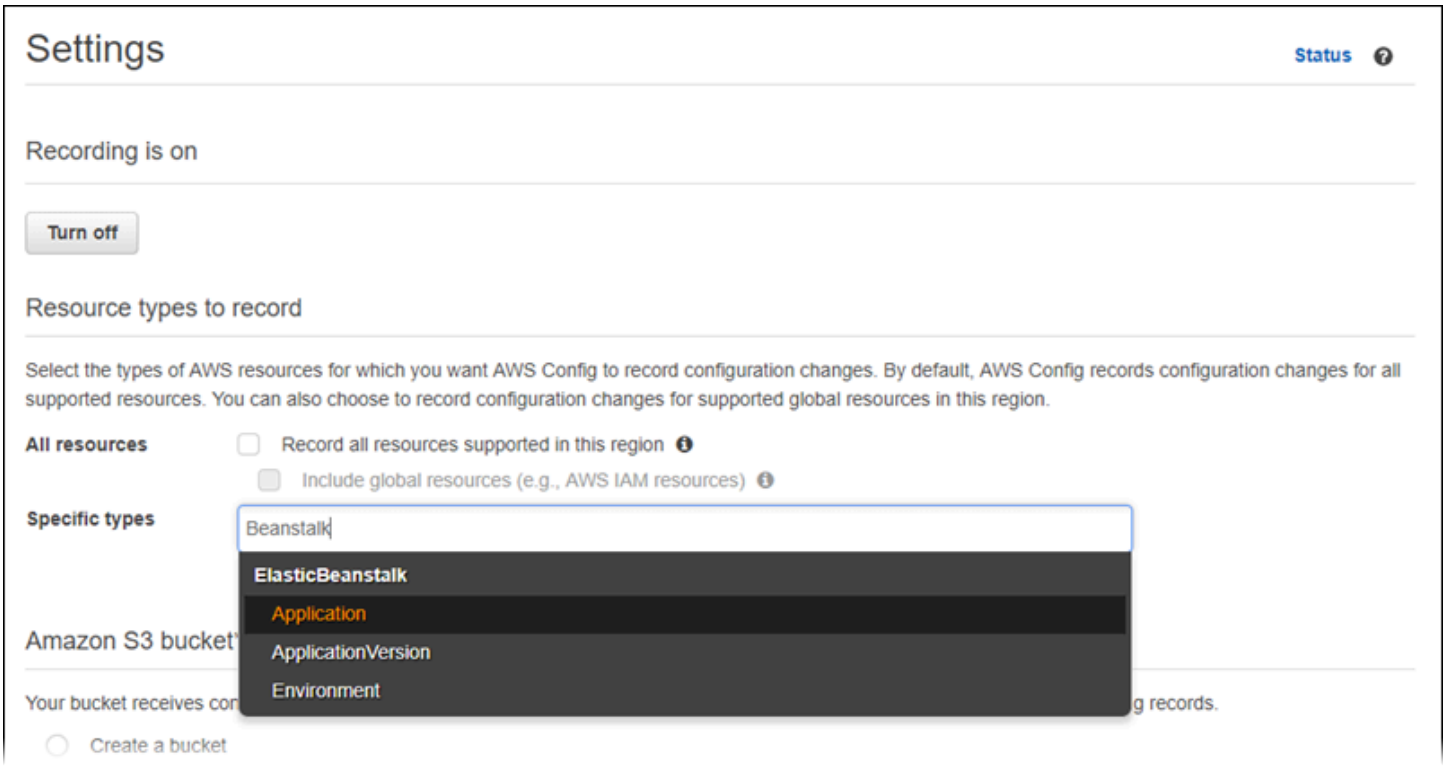
- Einrichtung AWS Config mit der Konsole
- Einrichtung AWS Config mit dem AWS CLI

Konfiguration für AWS Config die Aufzeichnung von Elastic Beanstalk Beanstalk-Ressourcen

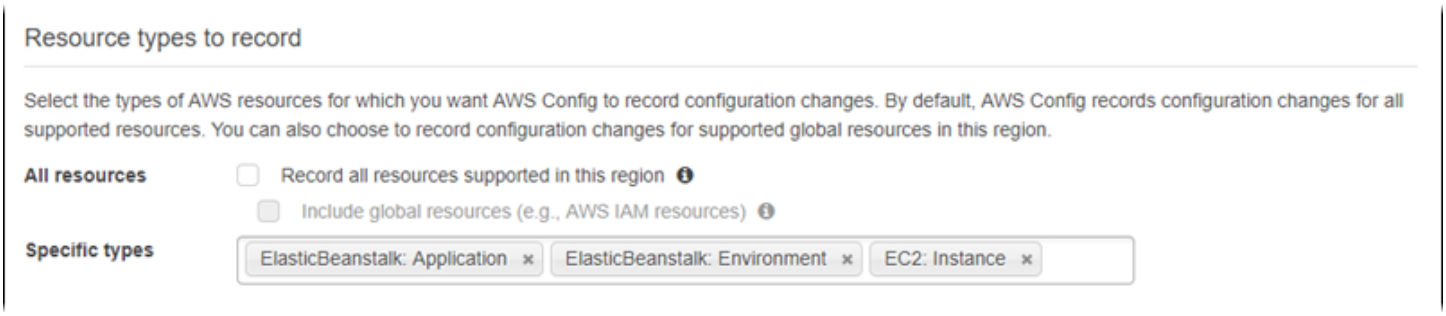
AWS Config Zeichnet standardmäßig Konfigurationsänderungen für alle unterstützten Typen von regionalen Ressourcen auf, die es in der Region entdeckt, in der Ihre Umgebung ausgeführt wird. Sie können es so AWS Config anpassen, dass Änderungen nur für bestimmte Ressourcentypen oder Änderungen an globalen Ressourcen aufgezeichnet werden.

Sie können beispielsweise so konfigurieren AWS Config , dass Änderungen für Elastic Beanstalk-Ressourcen und eine Teilmenge anderer AWS Ressourcen, die Elastic Beanstalk für Sie startet, aufgezeichnet werden. Mithilfe der AWS Config Konsole können Sie Elastic Beanstalk auf der Seite AWS Config Einstellungen im Feld Spezifische Typen als Ressource auswählen. Von dort aus können Sie wählen, ob Sie einen der Elastic Beanstalk Beanstalk-Ressourcentypen aufzeichnen möchten: Anwendung ApplicationVersion, und Umgebung.

Die folgende Abbildung zeigt die Seite AWS Config „Einstellungen“ mit Elastic Beanstalk Beanstalk-Ressourcentypen, die Sie aufzeichnen können: Anwendung ApplicationVersion, und Umgebung.



Nachdem Sie einige Ressourcentypen ausgewählt haben, sieht die Liste Specific types (Spezifische Typen) folgendermaßen aus.



Informationen zu regionalen und globalen Ressourcen und zum vollständigen Anpassungsverfahren finden Sie unter Auswahl der AWS Config Ressourceneinträge.

Elastic Beanstalk Beanstalk-Konfigurationsdetails in der Konsole anzeigen AWS Config

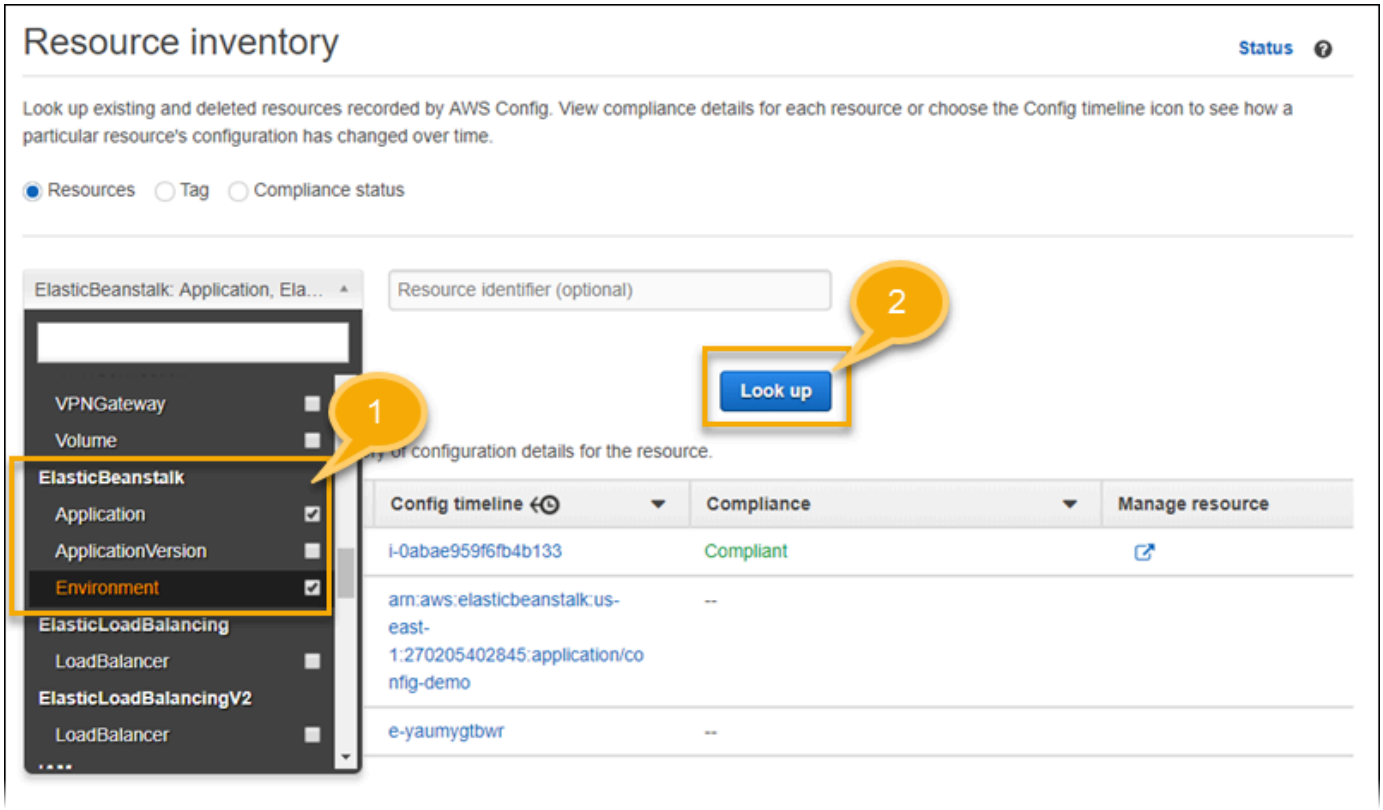
Sie können die AWS Config Konsole verwenden, um nach Elastic Beanstalk Beanstalk-Ressourcen zu suchen und aktuelle und historische Details zu deren Konfigurationen abzurufen. Das folgende Beispiel zeigt, wie Sie Informationen über eine Elastic Beanstalk-Umgebung finden.

So finden Sie eine Elastic Beanstalk Beanstalk-Umgebung in der Konsole AWS Config

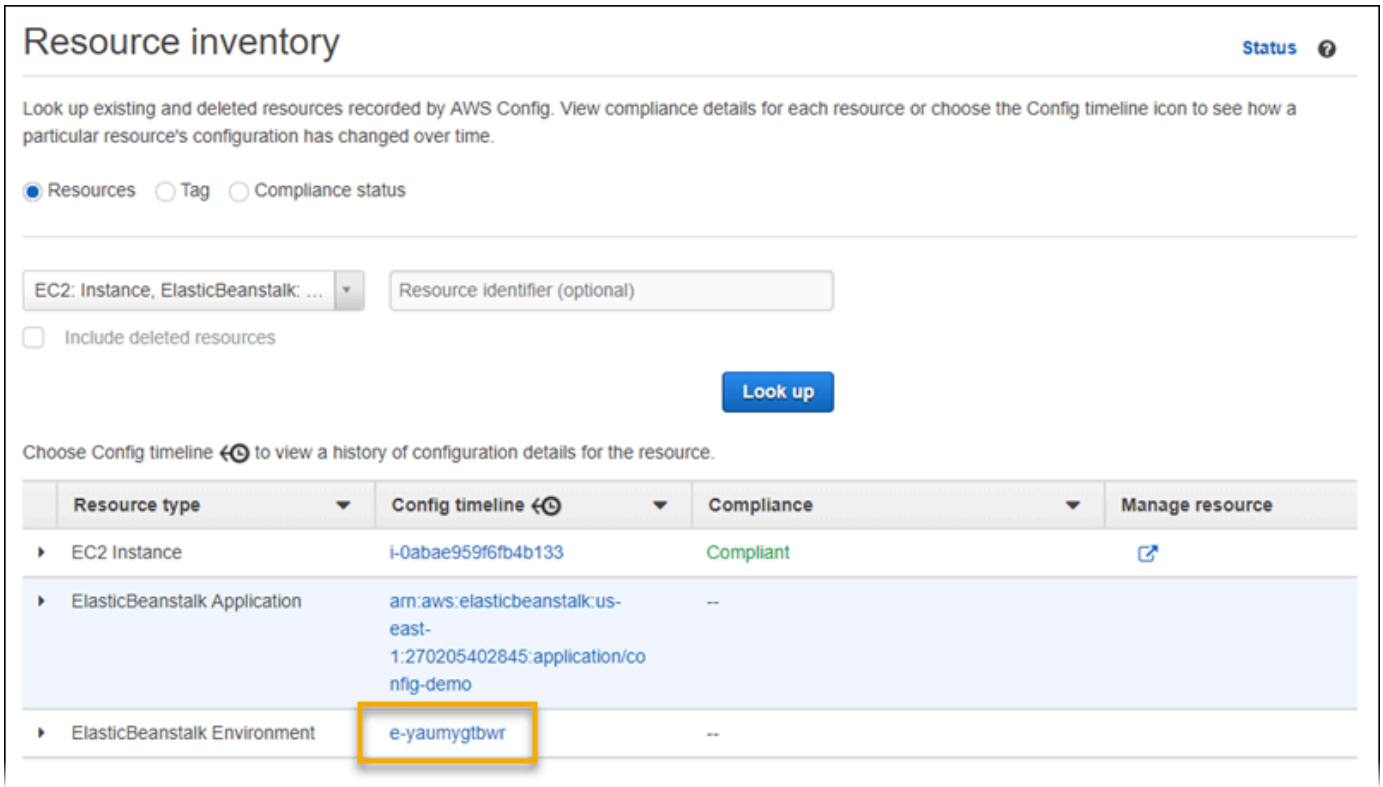
1. Öffnen Sie die AWS Config -Konsole.
2. Wählen Sie Resources aus.
3. Wählen Sie auf der Seite Resource inventory (Ressourcenbestand) die Option Resources (Ressourcen) aus.
4. Öffnen Sie das Menü Ressourcentyp, scrollen Sie zu ElasticBeanstalkund wählen Sie dann einen oder mehrere Elastic Beanstalk Beanstalk-Ressourcentypen aus.

   > **Note**
   >
   > Wählen Sie weitere Ressourcentypen, um Konfigurationsdetails für andere als die von Elastic Beanstalk für die Anwendung erstellten Ressourcen anzuzeigen. Sie können beispielsweise Instance unter auswählen. EC2

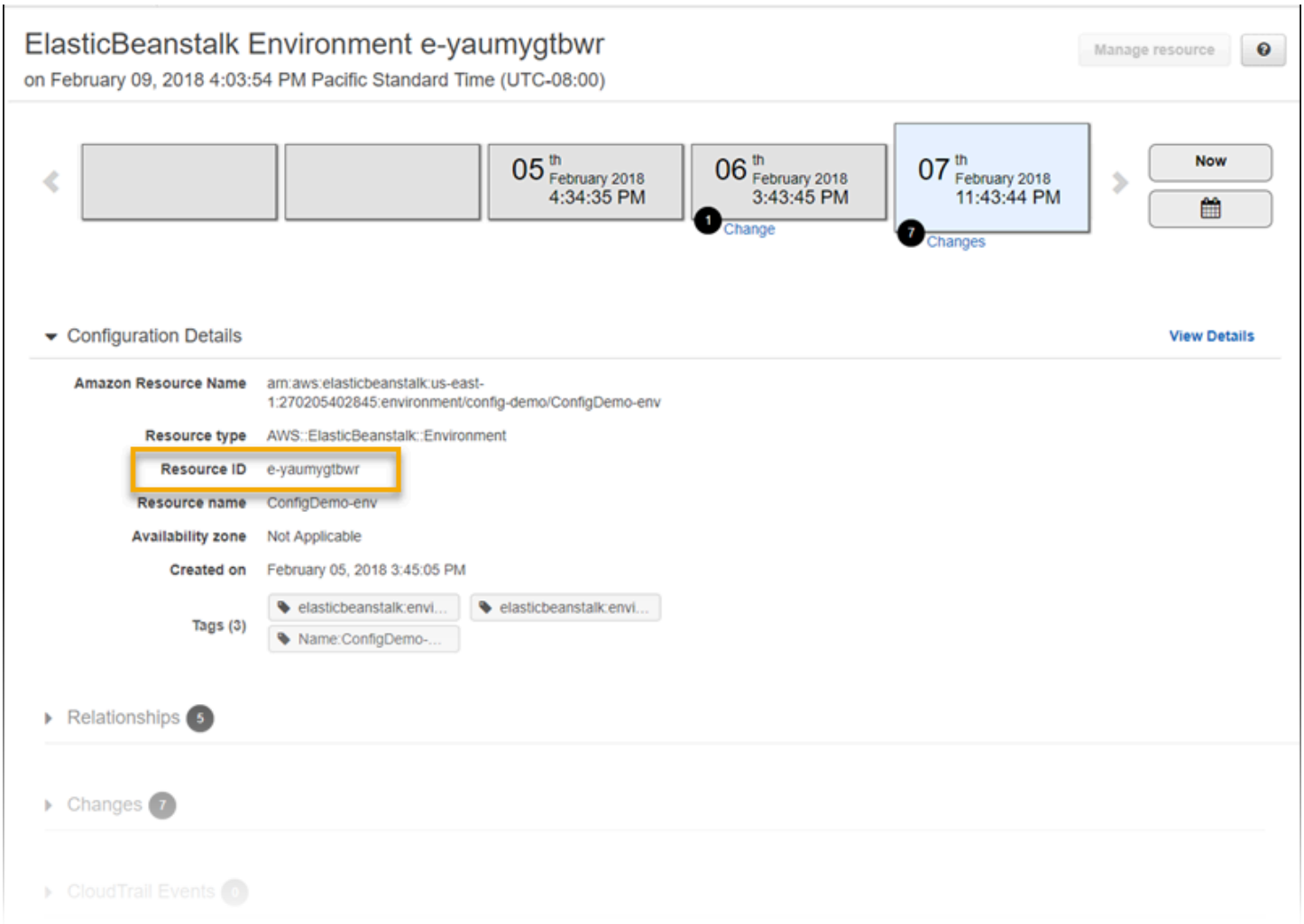
5. Wählen Sie Look up. Siehe 2 in der folgenden Abbildung.



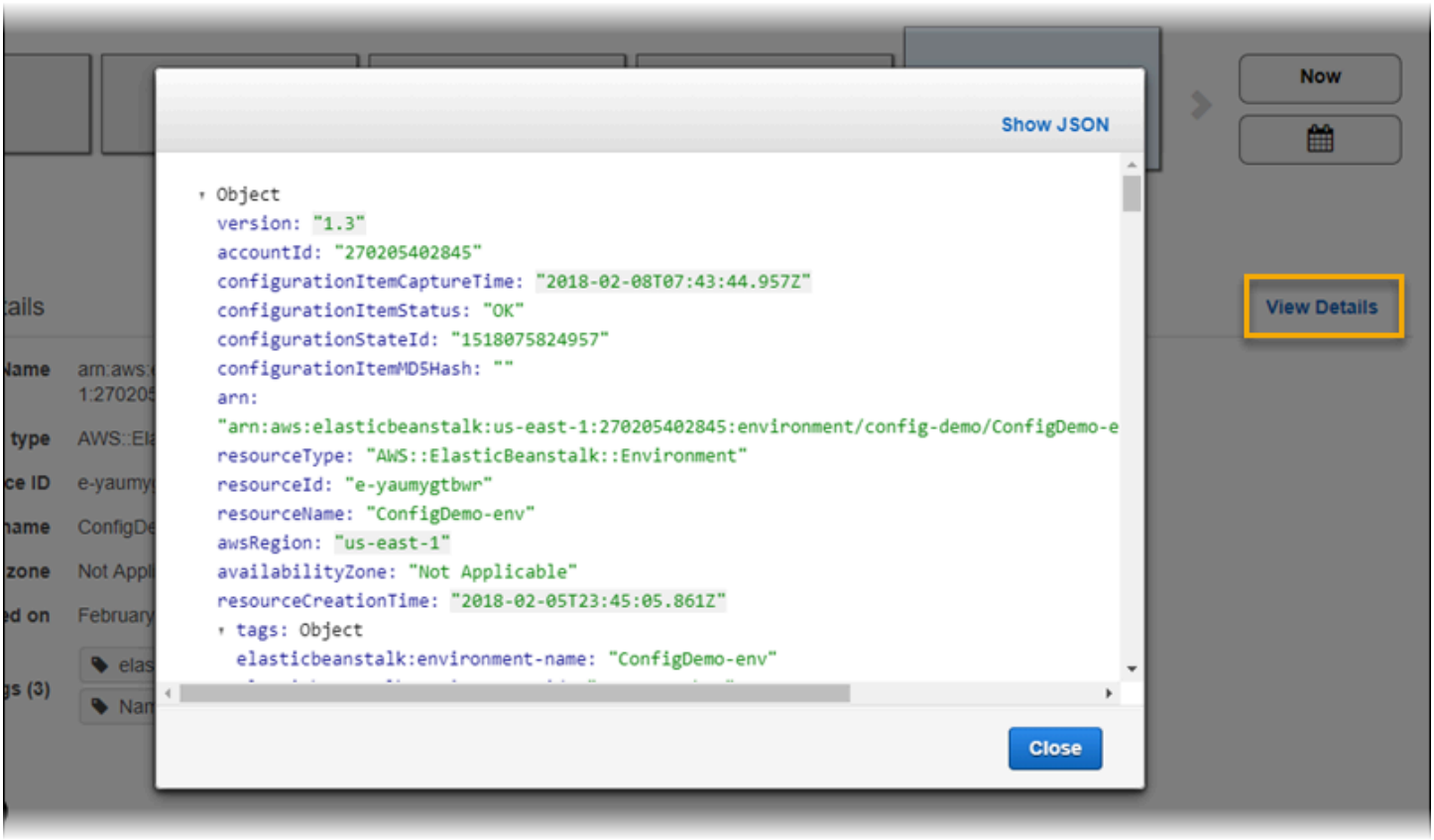
6. Wählen Sie in der angezeigten Ressourcenliste eine Ressourcen-ID aus. AWS Config



AWS Config zeigt Konfigurationsdetails und andere Informationen zu der ausgewählten Ressource an.



Die vollständigen Details der aufgezeichneten Konfiguration können Sie mit View Details (Details anzeigen) anzeigen.



Weitere Möglichkeiten, eine Ressource zu finden und Informationen auf dieser Seite einzusehen, finden Sie unter AWS Ressourcenkonfigurationen und Verlauf anzeigen im AWS Config Entwicklerhandbuch.

Evaluierung von Elastic Beanstalk Beanstalk-Ressourcen mithilfe von Regeln AWS Config

Sie können AWS Config Regeln erstellen, die die idealen Konfigurationseinstellungen für Ihre Elastic Beanstalk Beanstalk-Ressourcen darstellen. Sie können vordefinierte AWS Managed Config Rules verwenden oder benutzerdefinierte Regeln definieren. AWS Config verfolgt kontinuierlich Änderungen an der Konfiguration Ihrer Ressourcen, um festzustellen, ob diese Änderungen gegen eine der Bedingungen in Ihren Regeln verstoßen. Die AWS Config Konsole zeigt den Konformitätsstatus Ihrer Regeln und Ressourcen an.

Wenn eine Ressource gegen eine Regel verstößt und als nicht konform gekennzeichnet wird, AWS Config kann ich Sie mithilfe eines Amazon Simple Notification Service (Amazon SNS) -Themas benachrichtigen. Um die Daten in diesen AWS Config Benachrichtigungen programmgesteuert zu verarbeiten, verwenden Sie eine Amazon Simple Queue Service (Amazon SQS) -Warteschlange als Benachrichtigungsendpunkt für das Amazon SNS SNS-Thema. Sie können beispielsweise Code

schreiben, der einen Workflow startet, wenn die Auto Scaling-Gruppenkonfiguration der Umgebung geändert wird.

Weitere Informationen zum Einrichten und Verwenden von Regeln finden Sie unter Ressourcen mithilfe von Regeln bewerten im AWS Config Entwicklerhandbuch.AWS Config

Elastic Beanstalk mit Amazon DynamoDB verwenden

Amazon DynamoDB ist ein vollständig verwalteter NoSQL-Datenbank-Service, der schnelle und planbare Leistung mit nahtloser Skalierbarkeit bereitstellt. Als Entwickler können Sie mit DynamoDB eine Datenbanktabelle erstellen, um beliebige Datenmengen zu speichern und abzurufen und jeglichen Anforderungsdatenverkehr zu verarbeiten. DynamoDB verteilt die Daten und den Datenverkehr für die Tabelle automatisch auf eine ausreichende Anzahl von Servern, um die vom Kunden angegebene Anforderungskapazität zu erreichen und die gespeicherte Datenmenge zu verarbeiten und dabei gleichzeitig eine konsistente und schnelle Leistung beizubehalten. Alle Datenelemente werden auf Solid-State-Laufwerken (SSDs) gespeichert und automatisch über mehrere Availability Zones in einer AWS Region repliziert, um eine integrierte Hochverfügbarkeit und Datenbeständigkeit zu gewährleisten.

Wenn Sie regelmäßige Aufgaben in einer Worker-Umgebung verwenden, erstellt Elastic Beanstalk eine DynamoDB-Tabelle und verwendet diese, um die Leader-Wahl durchzuführen und Informationen über die Aufgabe zu speichern. Jede Instance in der Umgebung versucht alle paar Sekunden in die Tabelle zu schreiben, um Leader zu werden und die Aufgabe wie geplant durchzuführen.

Sie können Konfigurationsdateien zum Erstellen einer DynamoDB-Tabelle für Ihre Anwendung verwenden. Unter eb-node-express-sampleweiter finden Sie eine Beispielanwendung GitHub für Node.js, die eine Tabelle mit einer Konfigurationsdatei erstellt und mit dem AWS SDK für JavaScript in Node.js eine Verbindung zu dieser Tabelle herstellt. Eine Beispielanleitung für die Verwendung von DynamoDB mit PHP finden Sie unter Beispiel: DynamoDB CloudWatch, und SNS. Ein Beispiel, das den verwendet AWS SDK für Java, finden Sie unter Manage Tomcat Session State with DynamoDB in der Dokumentation. AWS SDK für Java

Beim Erstellen einer DynamoDB-Tabelle mit Konfigurationsdateien wird die Tabelle nicht mit dem Lebenszyklus der Umgebung verknüpft und nicht gelöscht, wenn Sie Ihre Umgebung beenden. Um sicherzustellen, dass personenbezogene Informationen nicht unnötig beibehalten werden, löschen Sie alle Datensätze, die Sie nicht mehr benötigen, oder löschen die Tabelle.

Weitere Informationen über DynamoDB finden Sie im Entwicklerhandbuch von DynamoDB.

Elastic Beanstalk mit Amazon verwenden ElastiCache

Amazon ElastiCache ist ein Webservice, der die Einrichtung, Verwaltung und Skalierung verteilter In-Memory-Cache-Umgebungen in der Cloud ermöglicht. Er bietet einen leistungsstarken, skalierbaren und kostengünstigen In-Memory-Cache und beseitigt gleichzeitig die Komplexität, die mit der Bereitstellung und Verwaltung einer verteilten Cache-Umgebung verbunden ist. ElastiCache ist protokollkonform mit Redis und Memcached, sodass der Code, die Anwendungen und die beliebtesten Tools, die Sie heute in Ihren bestehenden Redis- und Memcached-Umgebungen verwenden, problemlos mit dem Service zusammenarbeiten. Weitere Informationen zu ElastiCache finden Sie auf der ElastiCacheAmazon-Produktseite.

Um Elastic Beanstalk mit Amazon zu verwenden ElastiCache

1. Erstellen Sie einen Cluster ElastiCache .

   - Anweisungen zum Erstellen eines ElastiCache Clusters mit Redis finden Sie unter Getting Started with Amazon ElastiCache for Redis im ElastiCachefor Redis-Benutzerhandbuch.
   - Anweisungen zum Erstellen eines ElastiCache Clusters mit Memcached finden Sie unter Getting Started with Amazon ElastiCache for Memcached im for Memcached User GuideElastiCache .

2. Konfigurieren Sie Ihre ElastiCache Sicherheitsgruppe so, dass der Zugriff von der EC2 Amazon-Sicherheitsgruppe aus möglich ist, die von Ihrer Elastic Beanstalk Beanstalk-Anwendung verwendet wird. Anweisungen, wie Sie den Namen Ihrer EC2 Sicherheitsgruppe mithilfe der AWS Management Console ermitteln können, finden Sie EC2 Sicherheitsgruppen auf der Dokumentseite EC2 Instances.

   - Weitere Informationen zu Redis finden Sie unter Authorize Access im ElastiCache for Redis User Guide.
   - Weitere Informationen zu Memcached finden Sie unter Authorize Access im for Memcached User Guide. ElastiCache

Sie können Konfigurationsdateien verwenden, um Ihre Elastic Beanstalk Beanstalk-Umgebung an Ihre Bedürfnisse anzupassen. ElastiCache Beispiele für Konfigurationsdateien, die sich in Elastic ElastiCache Beanstalk integrieren lassen, finden Sie unter. Beispiel: ElastiCache

Elastic Beanstalk mit Amazon Elastic File System verwenden

Mit Amazon Elastic File System (Amazon EFS) können Sie Netzwerk-Dateisysteme erstellen, die von Instances über mehrere Availability Zones hinweg gemountet werden können. Ein Amazon EFS-Dateisystem ist eine AWS Ressource, die Sicherheitsgruppen verwendet, um den Zugriff über das Netzwerk zu kontrollieren, das sich in Ihrer Standard- oder benutzerdefinierten VPC befindet.

In einer Elastic-Beanstalk-Umgebung können Sie Amazon EFS verwenden, um ein gemeinsames Verzeichnis zu erstellen, in dem Dateien für Ihre Anwendung gespeichert werden, die Benutzer hochladen und ändern. Ihre Anwendung kann ein gemountetes Amazon-EFS-Volume wie lokalen Speicher behandeln. Auf diese Weise müssen Sie Ihren Anwendungscode nicht ändern, um auf mehrere Instances zu erweitern.

Weitere Informationen über Amazon EFS finden Sie im Amazon Elastic File System-Benutzerhandbuch .

> **Note**
>
> Elastic Beanstalk erstellt einen Webapp-Benutzer, den Sie als Eigentümer für Anwendungsverzeichnisse auf Amazon-Instances einrichten können. EC2 Weitere Informationen finden Sie unter Persistenter Speicher im Thema Überlegungen zum Entwurf dieses Handbuchs.

Sections

- Konfigurationsdateien
- Verschlüsselte Dateisysteme
- Beispielanwendungen
- Bereinigen von Dateisystemen

Konfigurationsdateien

Elastic Beanstalk bietet Konfigurationsdateien, die Sie zum Erstellen und Einbinden von Amazon EFS-Dateisystemen verwenden können. Sie können ein Amazon EFS-Volume als Teil Ihrer Umgebung erstellen oder ein Amazon EFS-Volume mounten, welches Sie unabhängig von Elastic Beanstalk erstellt haben.

- storage-efs-createfilesystem.config — Verwendet den `Resources` Schlüssel, um ein neues Dateisystem und Mount-Punkte in Amazon EFS zu erstellen. Alle Instances in Ihrer Umgebung können eine Verbindung zum selben Dateisystem herstellen, um einen gemeinsamen, skalierbaren Speicher zu erhalten. Das Mounting des Dateisystems auf der jeweiligen Instance erfolgt über `storage-efs-mountfilesystem.config`.

  > **Interne Ressourcen**
  >
  > Alle Ressourcen, die Sie mit Konfigurationsdateien erstellen, sind an den Lebenszyklus Ihrer Umgebung gebunden. Wenn Sie Ihre Umgebung beenden oder die Konfigurationsdatei entfernen, gehen diese Ressourcen verloren.

- storage-efs-mountfilesystem.config — Hängen Sie ein Amazon EFS-Dateisystem in einem lokalen Pfad auf den Instances in Ihrer Umgebung ein. Sie können das Volume als Teil der Umgebung mit `storage-efs-createfilesystem.config` erstellen. Oder Sie können es mit der Amazon EFS-Konsole oder dem AWS SDK in Ihre Umgebung einbinden. AWS CLI

Um die Konfigurationsdateien zu verwenden, beginnen Sie mit dem Erstellen Ihres Amazon EFS-Dateisystems mit `storage-efs-createfilesystem.config`. Befolgen Sie die Anweisungen in der Konfigurationsdatei und fügen Sie sie zum .ebextensions-Verzeichnis in Ihrem Quellcode hinzu, um das Dateisystem in Ihrer VPC zu erstellen.

Stellen Sie den aktualisierte Quellcode für Ihre Elastic Beanstalk-Umgebung bereit. Dies soll bestätigen, dass das Dateisystem erfolgreich erstellt wurde. Fügen Sie anschließend die `storage-efs-mountfilesystem.config` hinzu, um das Dateisystem für die Instances in Ihrer Umgebung bereitzustellen. Erfolgt dies in zwei separaten Bereitstellungen, wird sichergestellt, dass das Dateisystem intakt bleibt, wenn der Bereitstellungsvorgang fehlschlägt. Wenn Sie beides in derselben Bereitstellung durchführen, verursacht ein Problem in einem der Schritte, dass das Dateisystem beendet wird wenn die Bereitstellung fehlschlägt.

Verschlüsselte Dateisysteme

Amazon EFS unterstützt verschlüsselte Dateisysteme. Die in diesem Thema beschriebene Konfigurationsdatei `storage-efs-createfilesystem.config` definiert zwei benutzerdefinierte Optionen. Mit diesen Optionen können Sie ein Amazon-EFS-verschlüsseltes Dateisystem erstellen. Weitere Informationen erhalten Sie in den Anweisungen in der Konfigurationsdatei.

Beispielanwendungen

Elastic Beanstalk bietet auch Beispielanwendungen, die Amazon EFS für gemeinsam genutzten Speicher verwenden. Die beiden Projekte verfügen über Konfigurationsdateien, die Sie mit einem Standard WordPress - oder Drupal-Installationsprogramm verwenden können, um einen Blog oder ein anderes Content-Management-System in einer Umgebung mit Lastenausgleich auszuführen. Wenn ein Benutzer ein Foto oder anderen Medien hochlädt, wird die Datei auf einem Amazon-EFS-Dateisystem gespeichert. Dies vermeidet die Verwendung der Alternative, die ein Plugin verwendet, um hochgeladene Dateien in Amazon S3 zu speichern.

- Lastenausgleich WordPress — Dazu gehören die Konfigurationsdateien für die WordPress sichere Installation und Ausführung in einer Elastic Beanstalk Beanstalk-Umgebung mit Lastenausgleich.
- Lastenverteiltes Drupal – Dazu gehören Konfigurationsdateien und Anweisungen für die sichere Installation und Ausführung von Drupal in einer lastenverteilten Elastic-Beanstalk-Umgebung.

Bereinigen von Dateisystemen

Wenn Sie ein Amazon-EFS-Dateisystem unter Verwendung einer Konfigurationsdatei als Teil Ihrer Elastic-Beanstalk-Umgebung erstellt haben, entfernt Elastic Beanstalk das Dateisystem, wenn Sie die Umgebung terminieren. Um die Speicherkosten einer laufenden Anwendung zu minimieren, löschen Sie regelmäßig Dateien, die Ihre Anwendung nicht benötigt. Oder stellen Sie sicher, dass der Anwendungscode den Dateilebenszyklus korrekt beibehält.

> **Important**
>
> Wenn Sie darüber hinaus ein Amazon-EFS-Dateisystem außerhalb einer Elastic-Beanstalk-Umgebung erstellt und in den Instances der Umgebung gemountet haben, entfernt Elastic Beanstalk das Dateisystem nicht, wenn Sie die Umgebung beenden. Um sicherzustellen, dass Ihre persönlichen Daten nicht aufbewahrt werden und um Speicherkosten zu vermeiden, löschen Sie die Dateien, die Ihre Anwendung gespeichert hat, wenn Sie sie nicht mehr benötigen. Alternativ können Sie auch das gesamte Dateisystem entfernen.

Verwenden von Elastic Beanstalk mit AWS Identity and Access Management

AWS Identity and Access Management (IAM) hilft Ihnen dabei, den Zugriff auf Ihre AWS Ressourcen sicher zu kontrollieren. In diesem Abschnitt finden Sie Referenzmaterialien für die Nutzung der Richtlinien, Instance-Profile und Servicerollen von IAM.

Eine Übersicht über die Berechtigungen finden Sie unter Elastic Beanstalk Service-Rollen, Instanzprofile und Benutzerrichtlinien. In den meisten Umgebungen verfügen die Service-Rolle und das Instance-Profil, zu deren Erstellung Sie von der Elastic Beanstalk-Konsole beim Starten der ersten Umgebung aufgefordert werden, über alle erforderlichen Berechtigungen. Ebenso enthalten auch die von Elastic Beanstalk bereitgestellten verwalteten Richtlinien für Vollzugriff und schreibgeschützten Zugriff alle Benutzerberechtigungen, die bei der täglichen Arbeit benötigt werden.

Das IAM-Benutzerhandbuch bietet ausführliche Informationen zu Berechtigungen. AWS

Themen

- Elastic Beanstalk Instance-Profile verwalten
- Rollen von Elastic Beanstalk Service verwalten
- Servicegebundene Rollen für Elastic Beanstalk verwenden
- Elastic Beanstalk-Benutzerrichtlinien verwalten
- Amazon-Ressourcennamen-Format für Elastic Beanstalk
- Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen
- Tags zur Steuerung des Zugriffs auf Elastic Beanstalk-Ressourcen verwenden
- Beispielrichtlinien basierend auf verwalteten Richtlinien
- Beispielrichtlinien auf Basis von Ressourcenberechtigungen
- Verhinderung des umgebungsübergreifenden Amazon S3 S3-Bucket-Zugriffs

Elastic Beanstalk Instance-Profile verwalten

Ein Instance-Profil ist ein Container für eine AWS Identity and Access Management (IAM-) Rolle, den Sie verwenden können, um Rolleninformationen an eine Amazon EC2 EC2-Instance zu übergeben, wenn die Instance gestartet wird.

Wenn Ihr AWS Konto kein EC2-Instance-Profil hat, müssen Sie eines mit dem IAM-Service erstellen. Anschließend können Sie das EC2-Instance-Profil neuen Umgebungen zuweisen, die Sie erstellen. Die Schritte Umgebung erstellen in der Elastic Beanstalk Beanstalk-Konsole bieten Ihnen Zugriff auf die IAM-Konsole, sodass Sie ein EC2-Instance-Profil mit den erforderlichen Berechtigungen erstellen können.

> **Note**
>
> Zuvor hat Elastic Beanstalk ein standardmäßiges EC2-Instance-Profil erstellt, `aws-elasticbeanstalk-ec2-role` das beim ersten Erstellen einer Umgebung durch ein AWS Konto benannt wurde. Dieses Instance-Profil enthielt verwaltete Standardrichtlinien. Wenn Ihr Konto bereits über dieses Instance-Profil verfügt, können Sie es weiterhin Ihren Umgebungen zuweisen.
> Aktuelle AWS Sicherheitsrichtlinien erlauben es einem AWS Service jedoch nicht, automatisch Rollen mit Vertrauensrichtlinien für andere AWS Dienste, in diesem Fall EC2, zu erstellen. Aufgrund dieser Sicherheitsrichtlinien erstellt Elastic Beanstalk kein `aws-elasticbeanstalk-ec2-role`-Standard-Instance-Profil mehr.

Verwaltete Richtlinien

Elastic Beanstalk bietet mehrere verwaltete Richtlinien, damit Ihre Umgebung unterschiedlichen Anwendungsfällen gerecht wird. Um die Standardanwendungsfälle für eine Umgebung zu erfüllen, müssen diese Richtlinien der Rolle für das EC2-Instance-Profil zugeordnet werden.

- AWSElasticBeanstalkWebTier— Erteilt der Anwendung Berechtigungen zum Hochladen von Protokollen auf Amazon S3 und zum Debuggen von Informationen. AWS X-Ray Informationen zum Inhalt der verwalteten Richtlinien finden Sie AWSElasticBeanstalkWebTierim Referenzhandbuch für AWS verwaltete Richtlinien.
- AWSElasticBeanstalkWorkerTier— Erteilt Berechtigungen für Protokoll-Uploads, Debugging, Veröffentlichung von Metriken und Worker-Instance-Aufgaben, einschließlich Warteschlangenverwaltung, Wahl des Leiters und periodische Aufgaben. Informationen zum Inhalt der verwalteten Richtlinien finden Sie AWSElasticBeanstalkWorkerTierim Referenzhandbuch für AWS verwaltete Richtlinien.
- AWSElasticBeanstalkMulticontainerDocker— Erteilt dem Amazon Elastic Container Service Berechtigungen zur Koordination von Cluster-Aufgaben für Docker-Umgebungen. Informationen

zum Inhalt der verwalteten Richtlinien finden Sie AWSElasticBeanstalkMulticontainerDockerim Referenzhandbuch für AWS verwaltete Richtlinien.

> **Important**
>
> Von Elastic Beanstalk verwaltete Richtlinien bieten keine granularen Berechtigungen – sie gewähren alle Berechtigungen, die möglicherweise für die Arbeit mit Elastic-Beanstalk-Anwendungen erforderlich sind. In einigen Fällen möchten Sie möglicherweise die Berechtigungen unserer verwalteten Richtlinien weiter einschränken. Ein Beispiel für einen Anwendungsfall finden Sie unterVerhinderung des umgebungsübergreifenden Amazon S3 S3-Bucket-Zugriffs.
>
> Unsere verwalteten Richtlinien decken auch keine Berechtigungen für benutzerdefinierte Ressourcen ab, die Sie möglicherweise zu Ihrer Lösung hinzufügen und die nicht von Elastic Beanstalk verwaltet werden. Verwenden Sie benutzerdefinierte Richtlinien, um detailliertere Berechtigungen, mindestens erforderliche Berechtigungen oder benutzerdefinierte Ressourcenberechtigungen zu implementieren.

Richtlinie für Vertrauensbeziehungen für EC2

Damit die EC2-Instances in der Umgebung die eforderliche Rolle übernehmen können, muss das Instance-Profil Amazon EC2 wie folgt als vertrauenswürdige Entität in der Vertrauensstellungsrichtlinie angeben:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Wenn Sie die Berechtigungen anpassen möchten, können Sie entweder Richtlinien zur Rolle hinzufügen, die dem Instance-Standardprofil angefügt ist, oder ein eigenes Instance-Profil mit eingeschränkten Berechtigungen erstellen.

Um die KI-gestützte Umgebungsanalysefunktion zu verwenden, fügen Sie Ihrem Instanzprofil die `bedrock:InvokeModel` `elasticbeanstalk:DescribeEnvironmentHealth` Berechtigungen`elasticbeanstalk:DescribeEvents`,, und hinzu. `bedrock:ListFoundationModels` Diese Berechtigungen ermöglichen es Elastic Beanstalk, Amazon Bedrock zu verwenden und auf Umgebungsdaten zuzugreifen, um Protokolle, Ereignisse und den Zustand der Instance zu analysieren.

Sections

- Erstellen eines Instance-Profils
- Hinzufügen von Berechtigungen zum Instance-Standardprofil
- Überprüfen der dem Instance-Profil zugeordneten Berechtigungen
- Ein out-of-date Standard-Instanzprofil aktualisieren

Erstellen eines Instance-Profils

Ein Instance-Profil ist ein Wrapper für eine IAM-Standardrolle, damit eine EC2-Instance die Rolle übernehmen kann. Sie können ein Instanzprofil mit den standardmäßigen verwalteten Elastic Beanstalk Beanstalk-Richtlinien erstellen. Sie können auch zusätzliche Instanzprofile erstellen, um die Berechtigungen für verschiedene Anwendungen anzupassen. Oder Sie können ein Instanzprofil erstellen, das die beiden verwalteten Richtlinien, die Berechtigungen für Docker-Umgebungen auf Worker-Ebene oder ECS-verwaltete Docker-Umgebungen gewähren, nicht enthält, wenn Sie diese Funktionen nicht verwenden.

Um ein Instanzprofil mit den standardmäßigen verwalteten Richtlinien zu erstellen

1. Öffnen Sie die Seite Roles (Rollen)in der IAM-Konsole.
2. Wählen Sie Rolle erstellen aus.
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS -Service aus.
4. Wählen Sie für Service oder Anwendungsfall Elastic Beanstalk.
5. Wählen Sie als Anwendungsfall Elastic Beanstalk — Compute.
6. Wählen Sie Weiter aus.
7. Geben Sie einen Role name ein.

Sie können den Namen der Standardrolle eingeben, die die Elastic Beanstalk Beanstalk-Konsole vorschlägt:. `aws-elasticbeanstalk-ec2-role`

8. Vergewissern Sie sich, dass die Berechtigungsrichtlinien Folgendes beinhalten, und wählen Sie dann Weiter aus:

   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`

9. Wählen Sie Rolle erstellen aus.

Um ein Instanzprofil mit einer bestimmten Auswahl an verwalteten Richtlinien zu erstellen

1. Öffnen Sie die Seite Roles (Rollen)in der IAM-Konsole.
2. Wählen Sie Rolle erstellen aus.
3. Wählen Sie unter Typ der vertrauenswürdigen Entität die Option AWS -Service aus.
4. Wählen Sie unter Use case (Anwendungsfall) die Option EC2 aus.
5. Wählen Sie Weiter aus.
6. Fügen Sie die entsprechenden von Elastic Beanstalk bereitgestellten verwalteten Richtlinien sowie weitere Richtlinien hinzu, um der Anwendung die erforderlichen Berechtigungen zu erteilen.
7. Wählen Sie Weiter aus.
8. Geben Sie einen Namen für die Rolle ein.
9. (Optional) Fügen Sie der Rolle Tags hinzu.
10. Wählen Sie Rolle erstellen aus.

Hinzufügen von Berechtigungen zum Instance-Standardprofil

Wenn Ihre Anwendung auf Ressourcen zugreift AWS APIs oder für die im Standard-Instanzprofil keine Berechtigungen gewährt wurden, fügen Sie Richtlinien hinzu, die Berechtigungen in der IAM-Konsole gewähren.

Hinzufügen von Richtlinien zur Rolle, die dem Instance-Standardprofil angefügt ist

1. Öffnen Sie die Seite Roles (Rollen) in der IAM-Konsole.

2. Wählen Sie die Rolle, die Ihnen als EC2-Instance-Profil zugewiesen wurde.
3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Attach policies (Richtlinien anfügen) aus.
4. Wählen Sie die verwaltete Richtlinie für die zusätzlichen Services aus, die von der Anwendung verwendet werden. Zum Beispiel `AmazonS3FullAccess` oder `AmazonDynamoDBFullAccess`.
5. Wählen Sie Richtlinie anfügen aus.

Überprüfen der dem Instance-Profil zugeordneten Berechtigungen

Die Berechtigungen, die dem Instance-Standardprofil zugeordnet sind, können je nach Erstellungsdatum, Datum des letzten Umgebungsstarts und verwendetem Client unterschiedlich sein. Die Berechtigungen für das Instance-Standardprofil können Sie in der IAM-Konsole überprüfen.

So überprüfen Sie die Berechtigungen des Instance-Standardprofils

1. Öffnen Sie die Seite Roles (Rollen)in der IAM-Konsole.
2. Wählen Sie die Rolle, die Ihnen als EC2-Instance-Profil zugewiesen wurde.
3. Prüfen Sie auf der Registerseite Permissions (Berechtigungen) die der Rolle angefügte Liste von Richtlinien.
4. Um die von einer Richtlinie erteilten Berechtigungen anzuzeigen, wählen Sie die Richtlinie aus.

Ein out-of-date Standard-Instanzprofil aktualisieren

Falls benötigte Berechtigungen für das Instance-Standardprofil fehlen, können Sie die verwalteten Richtlinien manuell zu der Ihrem EC2-Instance-Profil zugewiesen Rolle hinzufügen.

Hinzufügen von verwalteten Richtlinien zur Rolle, die dem Instance-Standardprofil angefügt ist

1. Öffnen Sie die Seite Roles (Rollen)in der IAM-Konsole.
2. Wählen Sie die Rolle, die Ihnen als EC2-Instance-Profil zugewiesen wurde.
3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Attach policies (Richtlinien anfügen) aus.
4. Geben Sie **`AWSElasticBeanstalk`** ein, um die Richtlinien zu filtern.
5. Wählen Sie die folgenden Richtlinien und anschließend Attach Policies (Richtlinien anfügen) aus:

   - `AWSElasticBeanstalkWebTier`

- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

Rollen von Elastic Beanstalk Service verwalten

AWS Elastic Beanstalk Führt in Ihrem Namen Aktionen mit Umgebungsressourcen durch, um Ihre Umgebung zu verwalten und zu überwachen. Elastic Beanstalk benötigt bestimmte Berechtigungen, um diese Aktionen auszuführen, und nimmt AWS Identity and Access Management (IAM-) Servicerollen an, um diese Berechtigungen zu erhalten.

Beim Annehmen einer Servicerolle muss Elastic Beanstalk stets temporäre Sicherheitsanmeldeinformationen verwenden. Um diese Anmeldeinformationen abzurufen, sendet Elastic Beanstalk eine Anforderung an AWS -Security-Token-Service (AWS STS) auf einem regionsspezifischen Endpunkt. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen im IAM-Benutzerhandbuch.

> **Note**
>
> Wenn der AWS STS Endpunkt für die Region, in der sich Ihre Umgebung befindet, deaktiviert ist, sendet Elastic Beanstalk die Anfrage an einen alternativen Endpunkt, der nicht deaktiviert werden kann. Dieser Endpunkt ist mit einer anderen Region verknüpft. Daher ist die Anfrage eine regionsübergreifende Anfrage. Weitere Informationen finden Sie unter Aktivierung und Deaktivierung AWS STS in einer AWS Region im IAM-Benutzerhandbuch.

Servicerollen mit der Elastic Beanstalk-Konsole und EB CLI verwalten

Sie können die Elastic Beanstalk-Konsole und die EB-Befehlszeilenschnittstelle verwenden, um Servicerollen mit für Ihre Umgebung ausreichenden Berechtigungen einzurichten. Sie erstellen eine Standardservicerolle und verwenden darin verwaltete Richtlinien.

Richtlinien zu verwalteten Servicerollen

Elastic Beanstalk bietet eine verwaltete Richtlinie für verbesserte Statusüberwachung und eine weitere mit für verwaltete Plattformaktualisierungen erforderlichen zusätzlichen Berechtigungen. Die Konsole und die EB-Befehlszeilenschnittstelle weisen beide Richtlinien der Standardservicerolle zu, die sie für Sie erstellen. Diese Richtlinien sollten nur für diese Standard-Servicerolle verwendet werden. Sie sollten nicht mit anderen Benutzern oder Rollen in Ihren Konten verwendet werden.

`AWSElasticBeanstalkEnhancedHealth`

Diese Richtlinie erteilt Berechtigungen für die Überwachung der Instance- und Umgebungsintegrität durch Elastic Beanstalk. Sie umfasst auch Amazon-SQS-Aktionen, damit Elastic Beanstalk die Warteschlangenaktivität für Worker-Umgebungen überwachen kann. Den Inhalt dieser verwalteten Richtlinie finden Sie auf der AWSElasticBeanstalkEnhancedHealthSeite im Referenzhandbuch für AWS verwaltete Richtlinien.

`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Diese Richtlinie erteilt Elastic Beanstalk Berechtigungen, um Umgebungen in Ihrem Namen zu aktualisieren, um verwaltete Plattformaktualisierungen durchzuführen. Den Inhalt dieser verwalteten Richtlinie finden Sie AWSElasticBeanstalkManagedUpdatesCustomerRolePolicyauf der Seite im Referenzhandbuch für AWS verwaltete Richtlinien.

Service Level-Berechtigungsgruppierungen

Diese Richtlinie ist in Anweisungen gruppiert, die auf den bereitgestellten Berechtigungen basieren.

- *`ElasticBeanstalkPermissions`*— Diese Gruppe von Berechtigungen dient zum Aufrufen der Elastic Beanstalk-Serviceaktionen (Elastic APIs Beanstalk).
- *`AllowPassRoleToElasticBeanstalkAndDownstreamServices`* – Diese Berechtigungsgruppe ermöglicht es, jede Rolle an Elastic Beanstalk und an andere nachgelagerte Services wie CloudFormation zu übergeben.
- *`ReadOnlyPermissions`* — Diese Berechtigungsgruppe dient zum Sammeln von Informationen über die laufende Umgebung.
- *`*OperationPermissions`* — Gruppen mit diesem Benennungsmuster dienen zum Aufrufen der notwendigen Vorgänge zur Durchführung von Plattformaktualisierungen.
- *`*BroadOperationPermissions`* — Gruppen mit diesem Benennungsmuster dienen zum Aufrufen der notwendigen Vorgänge zur Durchführung von Plattformaktualisierungen. Sie enthalten auch umfassende Berechtigungen zur Unterstützung von Legacy-Umgebungen.
- *`*TagResource`*— Gruppen mit diesem Benennungsmuster sind für Aufrufe vorgesehen, die Tags tag-on-create APIs an Ressourcen anhängen, die in einer Elastic Beanstalk Beanstalk-Umgebung erstellt werden.

Verwenden Sie die Seite Richtlinien in der IAM-Konsole, um den Inhalt einer verwalteten Richtlinie anzuzeigen.

> **Important**
>
> Von Elastic Beanstalk verwaltete Richtlinien bieten keine granularen Berechtigungen – sie gewähren alle Berechtigungen, die möglicherweise für die Arbeit mit Elastic-Beanstalk-Anwendungen erforderlich sind. In einigen Fällen möchten Sie möglicherweise die Berechtigungen unserer verwalteten Richtlinien weiter einschränken. Ein Beispiel für einen Anwendungsfall finden Sie unterVerhinderung des umgebungsübergreifenden Amazon S3 S3-Bucket-Zugriffs.
> Unsere verwalteten Richtlinien decken auch keine Berechtigungen für benutzerdefinierte Ressourcen ab, die Sie möglicherweise zu Ihrer Lösung hinzufügen und die nicht von Elastic Beanstalk verwaltet werden. Verwenden Sie benutzerdefinierte Richtlinien, um detailliertere Berechtigungen, mindestens erforderliche Berechtigungen oder benutzerdefinierte Ressourcenberechtigungen zu implementieren.

Veraltete, von verwaltete Richtlinien

In der Vergangenheit unterstützte Elastic Beanstalk die Rollenrichtlinie für AWSElasticBeanstalkServiceverwaltete Dienste. Diese Richtlinie wurde ersetzt durch AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy. Möglicherweise können Sie die frühere Richtlinie weiterhin in der IAM-Konsole sehen und verwenden.

Den Inhalt der verwalteten Richtlinien finden Sie AWSElasticBeanstalkServiceim Referenzhandbuch für AWS verwaltete Richtlinien.

Wir empfehlen jedoch, zur Verwendung der neuen verwalteten Richtlinie (AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy) überzugehen. Fügen Sie benutzerdefinierte Richtlinien hinzu, um Berechtigungen für benutzerdefinierte Ressourcen zu erteilen, falls vorhanden.

Elastic-Beanstalk-Konsole verwenden

Wenn Sie in der Elastic Beanstalk-Konsole eine Umgebung starten, erstellt die Konsole eine Standard-Servicerolle mit dem Namen `aws-elasticbeanstalk-service-role` und fügt dieser Servicerolle verwaltete Richtlinien mit Standardberechtigungen an.

Damit Elastic Beanstalk die Rolle `aws-elasticbeanstalk-service-role` übernehmen kann, legt die Servicerolle Elastic Beanstalk als vertrauenswürdige Einheit in der Vertrauensbeziehungsrichtlinie fest.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
          "Service": "elasticbeanstalk.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
          "StringEquals": {
            "sts:ExternalId": "elasticbeanstalk"
          }
        }
      }
    ]
}
```

Wenn Sie für Ihre Umgebung verwaltete Plattformaktualisierungen aktivieren, übernimmt Elastic Beanstalk eine separate Servicerolle für verwaltete Updates, um verwaltete Updates auszuführen. Standardmäßig verwendet die Elastic Beanstalk-Konsole die gleiche generierte Servicerolle, `aws-elasticbeanstalk-service-role`, als Servicerolle für verwaltete Updates. Bei einer Änderung der Standard-Servicerolle legt die Konsole die Servicerolle für verwaltete Updates auf die Verwendung der serviceverknüpften Rolle für verwaltete Updates fest, `AWSServiceRoleForElasticBeanstalkManagedUpdates`. Weitere Informationen zu serviceverknüpften Rollen finden Sie unter the section called “Verwenden von servicegebundenen Rollen”.

> **Note**
>
> Aufgrund von Berechtigungsproblemen gelingt es dem Elastic Beanstalk-Service nicht immer, diese serviceverknüpfte Rolle für Sie zu erstellen. Daher versucht die Konsole, sie explizit zu erstellen. Um sicherzustellen, dass Ihr Konto über diese servicebezogene Rolle verfügt, erstellen Sie mindestens einmal mithilfe der Konsole eine Umgebung und konfigurieren Sie verwaltete Updates, die aktiviert werden, bevor Sie die Umgebung erstellen.

Verwenden der EB-CLI

Wenn Sie eine Umgebung mit dem the section called “eb create”-Befehl der Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) starten und keine Servicerolle über die `--service-role`-Option angeben, erstellt Elastic Beanstalk die Standard-Servicerolle `aws-elasticbeanstalk-service-role`. Wenn die standardmäßige Servicerolle bereits vorhanden ist, verwendet Elastic Beanstalk diese für die neue Umgebung. Die Elastic Beanstalk-Konsole führt in diesen Situationen ähnliche Aktionen aus.

Im Gegensatz zur Konsole können Sie über eine EB-CLI-Befehlsoption keine Servicerolle für verwaltete Updates angeben. Wenn Sie für Ihre Umgebung verwaltete Updates aktivieren, müssen Sie die Servicerolle für verwaltete Updates über Konfigurationsoptionen einstellen. Im folgenden Beispiel werden verwaltete Updates aktiviert und die Standardservicerolle als Servicerolle für verwaltete Updates verwendet.

Example.ebeextensions/ .config managed-platform-update

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
    ServiceRoleForManagedUpdates: "aws-elasticbeanstalk-service-role"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: patch
    InstanceRefreshEnabled: true
```

Servicerollen mit der Elastic Beanstalk API verwalten

Wenn Sie eine Umgebung mithilfe der `CreateEnvironment`-Aktion der Elastic Beanstalk-API erstellen, geben Sie eine Servicerolle mit der `ServiceRole`-Konfigurationsoption im `aws:elasticbeanstalk:environment`-Namespace an. Einzelheiten zur Verwendung einer erweiterten Zustandsüberwachung mit der Elastic Beanstalk-API finden Sie unter Verwenden der erweiterten Integritätsberichte mit der Elastic-Beanstalk-API.

Wenn Sie für Ihre Umgebung Verwaltete Plattformaktualisierungen aktivieren, können Sie über die Option `ServiceRoleForManagedUpdates` des Namespace `aws:elasticbeanstalk:managedactions` eine Servicerolle für verwaltete Updates angeben.

Verwenden von servicegebundenen Rollen

Eine serviceverknüpfte Rolle ist eine einzigartige Art von Servicerolle, die von Elastic Beanstalk so vordefiniert wurde, dass sie alle Berechtigungen beinhaltet, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen. Die serviceverknüpfte Rolle ist Ihrem Konto zugeordnet. Sie wird von Elastic Beanstalk einmalig erstellt und dann zum Erstellen weiterer Umgebungen wiederverwendet. Weitere Informationen zur Verwendung von serviceverknüpften Rollen mit Elastic Beanstalk-Umgebungen finden Sie unter Servicegebundene Rollen für Elastic Beanstalk verwenden.

Wenn Sie mit der Elastic Beanstalk-API eine Umgebung erstellen und keine Servicerolle angeben, erstellt Elastic Beanstalk eine überwachende serviceverknüpfte Rolle für Ihr Konto, sofern diese nicht bereits vorhanden ist. Elastic Beanstalk nutzt diese Rolle für die neue Umgebung. Sie können die serviceverknüpfte Rolle Ihres Kontos auch im Voraus mit IAM erstellen. Nachdem Ihr Konto über diese Rolle verfügt, können Sie damit eine Umgebung mit der Elastic Beanstalk-API, der Elastic Beanstalk-Konsole oder der EB CLI erstellen.

Wenn Sie verwaltete Plattform-Updates für die Umgebung aktivieren und `AWSServiceRoleForElasticBeanstalkManagedUpdates` als Wert für die `ServiceRoleForManagedUpdates`-Option des `aws:elasticbeanstalk:managedactions`-Namespace angeben, erstellt Elastic Beanstalk eine serviceverknüpfte Rolle für verwaltete Updates für Ihr Konto, falls diese nicht schon vorhanden ist. Elastic Beanstalk nutzt die Rolle, um verwaltete Updates für die neue Umgebung durchzuführen.

> **Note**
>
> Wenn Elastic Beanstalk versucht, die serviceverknüpften Rollen für verwaltete Updates und Überwachung für Ihr Konto zu erstellen, wenn Sie eine Umgebung erstellen, müssen Sie die Berechtigung `iam:CreateServiceLinkedRole`-besitzen. Wenn Sie nicht über diese Genehmigung verfügen, schlägt das Erstellen der Umgebung fehl und es wird eine Meldung angezeigt, die das Problem erläutert.
> Alternativ kann ein anderer Benutzer, der über die Berechtigung zum Erstellen von serviceverknüpften Rollen verfügt, IAM verwenden, um die serviceverknüpfte Rolle vorab zu erstellen. Mit dieser Methode benötigen Sie keine `iam:CreateServiceLinkedRole` Berechtigung, um Ihre Umgebung zu erstellen.

Überprüfung der Berechtigungen der Standard-Servicerolle

Die der Standard/Servicerolle gewährten Berechtigungen können je nach Erstellungsdatum, Datum des letzten Umgebungsstarts und verwendetem Client unterschiedlich sein. in der IAM-Konsole können Sie die von der Standard-Servicerolle gewährten Berechtigungen überprüfen.

So überprüfen Sie die Berechtigungen der Standard-Servicerolle

1. Öffnen Sie in der IAM-Konsole die Seite Roles (Rollen).
2. Wählen Sie aws-elasticbeanstalk-service-role.
3. Prüfen Sie auf der Registerseite Permissions (Berechtigungen) die der Rolle angefügte Liste von Richtlinien.
4. Um die von einer Richtlinie erteilten Berechtigungen anzuzeigen, wählen Sie die Richtlinie aus.

Aktualisierung einer Standard-Servicerolle out-of-date

Falls die benötigten Berechtigungen für die Standard-Servicerolle fehlen, können Sie sie aktualisieren, indem Sie in der Elastic Beanstalk-Environment Management Console eine neue Umgebung erstellen.

Alternativ können Sie die verwalteten Richtlinien manuell zur Standard-Servicerolle hinzufügen.

So fügen Sie verwaltete Richtlinien zur Standard-Servicerolle hinzu

1. Öffnen Sie in der IAM-Konsole die Seite Roles (Rollen).
2. Wählen Sie aws-elasticbeanstalk-service-role.
3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Attach policies (Richtlinien anfügen) aus.
4. Geben Sie **`AWSElasticBeanstalk`** ein, um die Richtlinien zu filtern.
5. Wählen Sie die folgenden Richtlinien und anschließend Attach Policies (Richtlinien anfügen) aus:
   - `AWSElasticBeanstalkEnhancedHealth`
   - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Hinzufügen von Berechtigungen zur Standard-Servicerolle

Wenn Ihre Anwendung Konfigurationsdateien enthält, die sich auf AWS Ressourcen beziehen, für die Berechtigungen nicht in der Standard-Servicerolle enthalten sind, benötigt Elastic Beanstalk möglicherweise zusätzliche Berechtigungen. Diese zusätzlichen Berechtigungen sind erforderlich, um diese Verweise aufzulösen, wenn die Konfigurationsdateien während eines verwalteten Updates verarbeitet werden. Wenn Berechtigungen fehlen, schlägt die Aktualisierung fehl, und Elastic Beanstalk gibt eine Meldung aus, in der die erforderliche Berechtigung angegeben wird. Fügen Sie in der IAM-Konsole mit den folgenden Schritten Berechtigungen für zusätzliche Services zur Standard-Servicerolle hinzu.

So fügen Sie zusätzliche Richtlinien zur Standard-Servicerolle hinzu

1. Öffnen Sie in der IAM-Konsole die Seite Roles (Rollen).
2. Wählen Sie aws-elasticbeanstalk-service-role.
3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Attach policies (Richtlinien anfügen) aus.
4. Wählen Sie die verwaltete Richtlinie für die zusätzlichen Services aus, die von der Anwendung verwendet werden. Zum Beispiel `AmazonAPIGatewayAdministrator` oder `AmazonElasticFileSystemFullAccess`.
5. Wählen Sie Richtlinie anfügen aus.

Erstellen einer -Servicerolle

Wenn Sie die Standard-Servicerolle nicht verwenden können, erstellen Sie eine Servicerolle.

So erstellen Sie eine Servicerolle

1. Öffnen Sie in der IAM-Konsole die Seite Roles (Rollen).
2. Wählen Sie Rolle erstellen aus.
3. Wählen Sie in AWS -Service die Option AWS Elastic Beanstalk und anschließend Ihren Anwendungsfall aus.
4. Wählen Sie Weiter: Berechtigungen aus.
5. Fügen Sie die verwalteten Richtlinien `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` und

`AWSElasticBeanstalkEnhancedHealth` sowie weitere Richtlinien hinzu, um der Anwendung die erforderlichen Berechtigungen zu erteilen.

6. Wählen Sie Weiter: Tags aus.
7. (Optional) Fügen Sie der Rolle Tags hinzu.
8. Wählen Sie Weiter: Prüfen aus.
9. Geben Sie einen Namen für die Rolle ein.
10. Wählen Sie Rolle erstellen aus.

Wenden Sie die benutzerdefinierte Servicerolle an, wenn Sie eine Umgebung im Assistenten zum Erstellen einer Umgebung oder mit der Option `--service-role` für den `eb create`-Befehl erstellen.

Servicegebundene Rollen für Elastic Beanstalk verwenden

AWS Elastic Beanstalk verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Elastic Beanstalk verknüpft ist. Servicebezogene Rollen sind von Elastic Beanstalk vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Elastic Beanstalk definiert einige Arten von servicegebundenen Rollen:

- Servicegebundene Überwachungsrolle – Ermöglicht es Elastic Beanstalk, den Status laufender Umgebungen zu überwachen und Benachrichtigungen über statusrelevante Ereignisse zu veröffentlichen.
- Servicegebundene Wartungsrolle – Ermöglicht es Elastic Beanstalk, regelmäßige Wartungsarbeiten für Ihre laufenden Umgebungen durchzuführen.
- Servicegebundene Rolle für verwaltete Updates – Ermöglicht es Elastic Beanstalk, geplante Plattformaktualisierungen Ihrer laufenden Umgebungen durchzuführen.

Themen

- Überwachen serviceverknüpfter Rollen
- Wartung serviceverknüpfter Rollen
- Serviceverknüpfte Rolle für verwaltete Updates

Überwachen serviceverknüpfter Rollen

AWS Elastic Beanstalk verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Elastic Beanstalk verknüpft ist. Servicebezogene Rollen sind von Elastic Beanstalk vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von Elastic Beanstalk, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Elastic Beanstalk definiert die Berechtigungen seiner servicegebundenen Rollen. Sofern nicht anders definiert, kann nur Elastic Beanstalk diese Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Elastic Beanstalk-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die supportservicegebundene Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Servicegebundene Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk verwendet die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForElasticBeanstalk— Ermöglicht Elastic Beanstalk, den Zustand laufender Umgebungen zu überwachen und Benachrichtigungen über Integritätsereignisse zu veröffentlichen.

Die AWSService RoleForElasticBeanstalk serviceverknüpfte Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `elasticbeanstalk.amazonaws.com`

Die Berechtigungsrichtlinie der AWSService RoleForElasticBeanstalk serviceverknüpften Rolle enthält alle Berechtigungen, die Elastic Beanstalk benötigt, um Aktionen in Ihrem Namen durchzuführen:

AllowCloudformationReadOperationsOnElasticBeanstalkStacks

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudformationReadOperationsOnElasticBeanstalkStacks",
            "Effect": "Allow",
            "Action": [
                "cloudformation:DescribeStackResource",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/awseb-*",
                "arn:aws:cloudformation:*:*:stack/eb-*"
            ]
        },
        {
            "Sid": "AllowOperations",
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeNotificationConfigurations",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:PutNotificationConfiguration",
                "ec2:DescribeInstanceStatus",
                "ec2:AssociateAddress",
                "ec2:DescribeAddresses",
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "elasticloadbalancing:DescribeInstanceHealth",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeTargetHealth",
                "elasticloadbalancing:DescribeTargetGroups",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sns:Publish"
            ],
            "Resource": [
```

```
                "*"
            ]
        }
    ]
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Alternativ können Sie eine AWS verwaltete Richtlinie verwenden, um vollen Zugriff auf Elastic Beanstalk zu gewähren.

Servicegebundene Rolle für Elastic Beanstalk erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk API erstellen und keine Servicerolle angeben, erstellt Elastic Beanstalk die servicegebundene Rolle für Sie.

> **Important**
>
> Wenn Sie den Elastic Beanstalk-Service vor dem 27. September 2017 genutzt haben, als er begann, die AWSService RoleForElasticBeanstalk serviceverknüpfte Rolle zu unterstützen, und Ihr Konto ihn benötigte, dann hat Elastic Beanstalk die AWSService RoleForElasticBeanstalk Rolle in Ihrem Konto erstellt. Weitere Informationen finden Sie unter Eine neue Rolle ist in meinem IAM-Konto erschienen.

Wenn Elastic Beanstalk versucht, die AWSService RoleForElasticBeanstalk serviceverknüpfte Rolle für Ihr Konto zu erstellen, wenn Sie eine Umgebung erstellen, benötigen Sie die entsprechende Genehmigung. `iam:CreateServiceLinkedRole` Wenn Sie nicht über diese Genehmigung verfügen, schlägt das Erstellen der Umgebung fehl und es wird eine Meldung angezeigt, die das Problem erklärt.

Alternativ kann ein anderer Benutzer, der über die Berechtigung zum Erstellen von serviceverknüpften Rollen verfügt, IAM verwenden, um die serviceverknüpfte Rolle vorab zu erstellen. Anschließend können Sie Ihre Umgebung auch ohne die `iam:CreateServiceLinkedRole`-Berechtigung selbst erstellen.

Sie (oder ein anderer Benutzer) können die IAM-Konsole verwenden, um eine servicegebundene Rolle mit dem Anwendungsfall Elastic Beanstalk zu erstellen. Erstellen Sie in der IAM-CLI oder der IAM-API eine serviceverknüpfte Rolle mit dem Servicenamen `elasticbeanstalk.amazonaws.com`. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpfte Rolle im IAM-Leitfaden. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen und keine Servicerolle angeben, erstellt Elastic Beanstalk die servicegebundene Rolle erneut für Sie.

Servicegebundene Rolle für Elastic Beanstalk bearbeiten

Elastic Beanstalk erlaubt es Ihnen nicht, die AWSService RoleForElasticBeanstalk serviceverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Servicegebundenen Rolle für Elastic Beanstalk bearbeiten

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie zunächst sicherstellen, dass alle Elastic Beanstalk-Umgebungen entweder eine andere Servicerolle verwenden oder beendet werden.

> **Note**
>
> Wenn der Elastic Beanstalk-Service die Rolle verwendet, wenn Sie versuchen, die Umgebungen zu beenden, tritt möglicherweise ein Fehler auf. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um eine Elastic Beanstalk Beanstalk-Umgebung zu beenden, die die AWSService RoleForElasticBeanstalk (Konsole) verwendet

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Lesen Sie für Einzelheiten über das Terminieren einer Elastic Beanstalk-Umgebung mit der EB CLI eb terminate.

Weitere Informationen TerminateEnvironmentzum Beenden einer Elastic Beanstalk Beanstalk-Umgebung mithilfe der API finden Sie unter.

Manuelles Löschen der serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die IAM-CLI oder die IAM-API, um die serviceverknüpfte Rolle zu löschen. AWSService RoleForElasticBeanstalk Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Leitfaden.

Unterstützte Regionen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Elastic Beanstalk - Endpunkte und -Kontingente.

Wartung serviceverknüpfter Rollen

AWS Elastic Beanstalk verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Elastic Beanstalk verknüpft ist. Servicebezogene Rollen sind von Elastic Beanstalk vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von Elastic Beanstalk, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Elastic Beanstalk definiert die

Berechtigungen seiner servicegebundenen Rollen. Sofern nicht anders definiert, kann nur Elastic Beanstalk diese Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Elastic Beanstalk-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die supportservicegebundene Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Servicegebundene Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk verwendet die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForElasticBeanstalkMaintenance— Ermöglicht Elastic Beanstalk, regelmäßige Wartungsaktivitäten für Ihre laufenden Umgebungen durchzuführen.

Die AWSService RoleForElasticBeanstalkMaintenance serviceverknüpfte Rolle vertraut darauf, dass die folgenden Services die Rolle übernehmen:

- `maintenance.elasticbeanstalk.amazonaws.com`

Die Berechtigungsrichtlinie der AWSService RoleForElasticBeanstalkMaintenance serviceverknüpften Rolle enthält alle Berechtigungen, die Elastic Beanstalk benötigt, um Aktionen in Ihrem Namen durchzuführen:

JSON

```
{
    "Version":"2012-10-17",
    "Statement":
        {
            "Sid":
 "AllowCloudformationChangeSetOperationsOnElasticBeanstalkStacks",
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateChangeSet",
```

```
                "cloudformation:DescribeChangeSet",
                "cloudformation:ExecuteChangeSet",
                "cloudformation:DeleteChangeSet",
                "cloudformation:ListChangeSets",
                "cloudformation:DescribeStacks"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/awseb-*",
                "arn:aws:cloudformation:*:*:stack/eb-*"
            ]
        }
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Alternativ können Sie eine AWS verwaltete Richtlinie verwenden, um vollen Zugriff auf Elastic Beanstalk zu gewähren.

Servicegebundene Rolle für Elastic Beanstalk erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen und kein Instance-Profil angeben, erstellt Elastic Beanstalk die servicegebundene Rolle für Sie.

> **Important**
>
> Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Features verwendet. Wenn Sie den Elastic Beanstalk-Service vor dem 18. April 2019 genutzt haben, als er begann, die AWSService RoleForElasticBeanstalkMaintenance serviceverknüpfte Rolle zu unterstützen, und Ihr Konto ihn benötigte, dann hat Elastic Beanstalk die AWSService RoleForElasticBeanstalkMaintenance Rolle in Ihrem Konto erstellt. Weitere Informationen finden Sie unter Eine neue Rolle ist in meinem IAM-Konto erschienen.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine Elastic

Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen und kein Instance-Profil angeben, erstellt Elastic Beanstalk die servicegebundene Rolle erneut für Sie.

Servicegebundene Rolle für Elastic Beanstalk bearbeiten

Elastic Beanstalk erlaubt es Ihnen nicht, die AWSService RoleForElasticBeanstalkMaintenance serviceverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Servicegebundenen Rolle für Elastic Beanstalk bearbeiten

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie IAM zum Löschen einer serviceverknüpften Rolle verwenden können, müssen Sie zuerst alle Elastic Beanstalk-Umgebungen beenden, die die Rolle verwenden.

> **Note**
>
> Wenn der Elastic Beanstalk-Service die Rolle verwendet, wenn Sie versuchen, die Umgebungen zu beenden, tritt möglicherweise ein Fehler auf. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um eine Elastic Beanstalk Beanstalk-Umgebung zu beenden, die die AWSService RoleForElasticBeanstalkMaintenance (Konsole) verwendet

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Lesen Sie für Einzelheiten über das Terminieren einer Elastic Beanstalk-Umgebung mit der EB CLI eb terminate.

Weitere Informationen TerminateEnvironmentzum Beenden einer Elastic Beanstalk Beanstalk-Umgebung mithilfe der API finden Sie unter.

Manuelles Löschen der serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die IAM-CLI oder die IAM-API, um die serviceverknüpfte Rolle zu löschen. AWSService RoleForElasticBeanstalkMaintenance Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Leitfaden.

Unterstützte Regionen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Elastic Beanstalk - Endpunkte und -Kontingente.

Serviceverknüpfte Rolle für verwaltete Updates

AWS Elastic Beanstalk verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Elastic Beanstalk verknüpft ist. Servicebezogene Rollen sind von Elastic Beanstalk vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von Elastic Beanstalk, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Elastic Beanstalk definiert die Berechtigungen seiner servicegebundenen Rollen. Sofern nicht anders definiert, kann nur Elastic Beanstalk diese Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Elastic Beanstalk-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die supportservicegebundene Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in

der Spalte Servicegebundene Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk verwendet die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForElasticBeanstalkManagedUpdates— Ermöglicht Elastic Beanstalk, geplante Plattform-Updates Ihrer laufenden Umgebungen durchzuführen.

Die AWSService RoleForElasticBeanstalkManagedUpdates serviceverknüpfte Rolle vertraut darauf, dass die folgenden Services die Rolle übernehmen:

- `managedupdates.elasticbeanstalk.amazonaws.com`

Die verwaltete Richtlinie AWSElasticBeanstalkManagedUpdatesServiceRolePolicygewährt der AWSService RoleForElasticBeanstalkManagedUpdates serviceverknüpften Rolle alle Berechtigungen, die Elastic Beanstalk benötigt, um verwaltete Aktualisierungsaktionen in Ihrem Namen durchzuführen. Den Inhalt der verwalteten Richtlinien finden Sie auf der AWSElasticBeanstalkManagedUpdatesServiceRolePolicySeite im Referenzhandbuch für AWS verwaltete Richtlinien.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Alternativ können Sie eine AWS verwaltete Richtlinie verwenden, um vollen Zugriff auf Elastic Beanstalk zu gewähren.

Servicegebundene Rolle für Elastic Beanstalk erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen, verwaltete Aktualisierungen aktivieren und `AWSServiceRoleForElasticBeanstalkManagedUpdates` als Wert für die `ServiceRoleForManagedUpdates`-Option des `aws:elasticbeanstalk:managedactions`-Namespaces angeben, erstellt Elastic Beanstalk die servicegebundene Rolle für Sie.

Wenn Elastic Beanstalk versucht, die AWSService RoleForElasticBeanstalkManagedUpdates serviceverknüpfte Rolle für Ihr Konto zu erstellen, wenn Sie eine Umgebung erstellen, benötigen Sie die entsprechende Genehmigung. `iam:CreateServiceLinkedRole` Wenn Sie nicht über

diese Genehmigung verfügen, schlägt das Erstellen der Umgebung fehl und es wird eine Meldung angezeigt, die das Problem erklärt.

Alternativ kann ein anderer Benutzer, der über die Berechtigung zum Erstellen von serviceverknüpften Rollen verfügt, IAM verwenden, um die serviceverknüpfte Rolle vorab zu erstellen. Anschließend können Sie Ihre Umgebung auch ohne die `iam:CreateServiceLinkedRole`-Berechtigung selbst erstellen.

Sie (oder ein anderer Benutzer) können die IAM-Konsole verwenden, um eine servicegebundene Rolle mit dem Anwendungsfall Elastic Beanstalk Managed Updates zu erstellen. Erstellen Sie in der IAM-CLI oder der IAM-API eine serviceverknüpfte Rolle mit dem Servicenamen `managedupdates.elasticbeanstalk.amazonaws.com`. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpfte Rolle im IAM-Leitfaden. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen, verwaltete Updates aktivieren und `AWSServiceRoleForElasticBeanstalkManagedUpdates` als Wert für die `ServiceRoleForManagedUpdates`-Option des `aws:elasticbeanstalk:managedactions`-Namespaces angeben, erstellt Elastic Beanstalk die servicegebundene Rolle erneut für Sie.

Servicegebundene Rolle für Elastic Beanstalk bearbeiten

Elastic Beanstalk erlaubt es Ihnen nicht, die AWSService RoleForElasticBeanstalkManagedUpdates serviceverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Servicegebundenen Rolle für Elastic Beanstalk bearbeiten

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie zunächst sicherstellen, dass Elastic Beanstalk-Umgebungen mit aktivierten verwalteten Updates entweder eine andere Servicerolle verwenden oder beendet werden.

> **Note**
>
> Wenn der Elastic Beanstalk-Service die Rolle verwendet, wenn Sie versuchen, die Umgebungen zu beenden, tritt möglicherweise ein Fehler auf. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um eine Elastic Beanstalk Beanstalk-Umgebung zu beenden, die die AWSService RoleForElasticBeanstalkManagedUpdates (Konsole) verwendet

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Lesen Sie für Einzelheiten über das Terminieren einer Elastic Beanstalk-Umgebung mit der EB CLI eb terminate.

Weitere Informationen TerminateEnvironmentzum Beenden einer Elastic Beanstalk Beanstalk-Umgebung mithilfe der API finden Sie unter.

Manuelles Löschen der serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die IAM-CLI oder die IAM-API, um die serviceverknüpfte Rolle zu löschen. AWSService RoleForElasticBeanstalkManagedUpdates Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Leitfaden.

Unterstützte Regionen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Elastic Beanstalk - Endpunkte und -Kontingente.

Elastic Beanstalk-Benutzerrichtlinien verwalten

AWS Elastic Beanstalk bietet zwei verwaltete Richtlinien, mit denen Sie allen Ressourcen, die Elastic Beanstalk verwaltet, vollen Zugriff oder schreibgeschützten Zugriff zuweisen können. Sie können die Richtlinien an AWS Identity and Access Management (IAM-) Benutzer oder Gruppen oder an Rollen anhängen, die von Ihren Benutzern übernommen wurden.

Verwaltete Benutzerrichtlinien

- AdministratorAccess- AWSElastic Beanstalk — Gibt dem Benutzer volle Administratorrechte zum Erstellen, Ändern und Löschen von Elastic Beanstalk Beanstalk-Anwendungen, Anwendungsversionen, Konfigurationseinstellungen, Umgebungen und den zugrunde liegenden Ressourcen. Den Inhalt der verwalteten Richtlinien finden Sie auf der Seite AdministratorAccess— AWSElastic Beanstalk im Referenzhandbuch für verwaltete Richtlinien.AWS
- AWSElasticBeanstalkReadOnly— Ermöglicht dem Benutzer das Anzeigen von Anwendungen und Umgebungen, jedoch nicht das Ausführen von Vorgängen, durch die diese geändert werden. Es bietet schreibgeschützten Zugriff auf alle Elastic Beanstalk Beanstalk-Ressourcen und auf andere AWS Ressourcen, die von der Elastic Beanstalk Beanstalk-Konsole abgerufen werden. Beachten Sie, dass mit schreibgeschütztem Zugriff keine Aktionen – wie z. B. Elastic Beanstalk-Protokolle zum Lesen herunterladen – möglich sind. Das liegt daran, dass die Protokolle im Amazon S3-Bucket bereitgestellt werden, wo Elastic Beanstalk eine Schreibberechtigung erfordern würde. Weitere Informationen zum Einrichten des Zugriffs auf Elastic Beanstalk-Protokolle finden Sie im Beispiel am Ende dieses Themas. Den Inhalt der verwalteten Richtlinien finden Sie auf der AWSElasticBeanstalkReadOnlySeite im Referenzhandbuch für verwaltete Richtlinien.AWS

> **Important**
>
> Von Elastic Beanstalk verwaltete Richtlinien bieten keine granularen Berechtigungen – sie gewähren alle Berechtigungen, die möglicherweise für die Arbeit mit Elastic-Beanstalk-Anwendungen erforderlich sind. In einigen Fällen möchten Sie möglicherweise die Berechtigungen unserer verwalteten Richtlinien weiter einschränken. Ein Beispiel für einen

Anwendungsfall finden Sie unterVerhinderung des umgebungsübergreifenden Amazon S3 S3-Bucket-Zugriffs.

Unsere verwalteten Richtlinien decken auch keine Berechtigungen für benutzerdefinierte Ressourcen ab, die Sie möglicherweise zu Ihrer Lösung hinzufügen und die nicht von Elastic Beanstalk verwaltet werden. Verwenden Sie benutzerdefinierte Richtlinien, um detailliertere Berechtigungen, mindestens erforderliche Berechtigungen oder benutzerdefinierte Ressourcenberechtigungen zu implementieren.

Veraltete, von verwaltete Richtlinien

Zuvor unterstützte Elastic Beanstalk zwei weitere verwaltete Benutzerrichtlinien, und AWSElasticBeanstalkFullAccess. AWSElasticBeanstalkReadOnlyAccess Wir planen, diese bisherigen Richtlinien außer Betrieb zu nehmen. Möglicherweise können Sie sie immer noch in der IAM-Konsole sehen und verwenden. Nichtsdestotrotz empfehlen wir Ihnen, auf die Verwendung der neuen Richtlinien für verwaltete Benutzer zu wechseln und benutzerdefinierte Richtlinien hinzuzufügen, um ggf. Berechtigungen für benutzerdefinierte Ressourcen zu erteilen.

Richtlinien für die Integration mit anderen Services

Wir bieten auch detailliertere Richtlinien, die es Ihnen ermöglichen, Ihre Umgebung in andere Services zu integrieren, falls Sie diese bevorzugen.

- AWSElasticBeanstalkRoleCWL — Ermöglicht einer Umgebung die Verwaltung von Amazon CloudWatch Logs-Protokollgruppen.
- AWSElasticBeanstalkRoleRDS — Ermöglicht einer Umgebung die Integration einer Amazon RDS-Instance.
- AWSElasticBeanstalkRoleWorkerTier— Ermöglicht einer Worker-Umgebungsebene, eine Amazon DynamoDB-Tabelle und eine Amazon SQS SQS-Warteschlange zu erstellen.
- AWSElasticBeanstalkRoleECS — Ermöglicht einer Docker-Umgebung mit mehreren Containern die Verwaltung von Amazon ECS-Clustern.
- AWSElasticBeanstalkRoleCore— Ermöglicht Kernoperationen einer Webservice-Umgebung.
- AWSElasticBeanstalkRoleSNS — Ermöglicht einer Umgebung, die Amazon SNS SNS-Themenintegration zu aktivieren.

Die JSON-Quelle für eine bestimmte verwaltete Richtlinie finden Sie im Referenzhandbuch für AWS verwaltete Richtlinien.

Steuern des Zugriffs mit verwalteten Richtlinien

Mit verwalteten Richtlinien können Sie Vollzugriff oder schreibgeschützten Zugriff auf Elastic Beanstalk gewähren. Diese Richtlinien werden von Elastic Beanstalk automatisch aktualisiert, wenn weitere Berechtigungen für den Zugriff auf neue Funktionen benötigt werden.

So wenden Sie eine verwaltete Richtlinie auf IAM-Benutzer oder -Gruppen an

1. Öffnen Sie die Seite Policies (Richtlinien) in der IAM-Konsole.
2. Geben Sie im Suchfeld **`AWSElasticBeanstalk`** ein, um die Richtlinien zu filtern.
3. Aktivieren Sie in der Liste der Richtlinien das Kontrollkästchen neben AWSElasticBeanstalkReadOnlyoder AdministratorAccess— AWSElastic Beanstalk.
4. Klicken Sie auf Policy actions und anschließend auf Attach.
5. Wählen Sie einen oder mehrere Benutzer und Gruppen aus, denen Sie die Richtlinie anfügen möchten. Über das Menü Filter und das Suchfeld können Sie die Liste der Prinzipal-Entitäten filtern.
6. Wählen Sie Richtlinie anfügen aus.

Erstellen einer benutzerdefinierten Benutzerrichtlinie

Sie können Ihre eigene IAM-Richtlinie erstellen, um bestimmte Elastic Beanstalk-API-Aktionen für bestimmte Elastic-Beanstalk-Ressourcen zuzulassen oder abzulehnen und den Zugriff auf benutzerdefinierte Ressourcen zu steuern, die nicht von Elastic Beanstalk verwaltet werden. Weitere Informationen zum Anfügen einer Richtlinie an einen Benutzer oder eine Gruppe finden Sie unter Arbeiten mit Richtlinien im IAM-Benutzerhandbuch. Weitere Informationen zum Erstellen einer benutzerdefinierten Richtlinie finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

> **Note**
>
> Sie können einschränken, wie Benutzer mit Elastic Beanstalk-APIs interagieren können. Jedoch gibt es derzeit keine effektive Möglichkeit, mit der Sie Benutzer mit Berechtigungen zum Erstellen der benötigten zugrunde liegenden Ressourcen daran hindern können, weitere Ressourcen in Amazon EC2 und anderen Services zu erstellen.
> Betrachten Sie diese Richtlinien als ein effektives Mittel zur Verteilung von Elastic Beanstalk-Verantwortlichkeiten, nicht als ein Mittel zur Sicherung aller zugrundeliegenden Ressourcen.

> **Important**
>
> Wenn Sie einer Elastic Beanstalk-Servicerolle benutzerdefinierte Richtlinien zugewiesen haben, ist es wichtig, dass Sie ihr die richtigen Berechtigungen für Startvorlagen zuweisen. Andernfalls verfügen Sie möglicherweise nicht über die erforderlichen Berechtigungen, um eine Umgebung zu aktualisieren oder eine neue zu starten. Weitere Informationen finden Sie unter Erforderliche Berechtigungen für -Startvorlagen.

Eine IAM-Richtlinie enthält Richtlinienanweisungen, mit denen die zu erteilenden Berechtigungen beschrieben werden. Um eine Richtlinienanweisung für Elastic Beanstalk zu erstellen, müssen Sie wissen, wie die folgenden vier Elemente einer Richtlinienanweisung verwendet werden:

- Effect gibt an, ob die Aktionen in der Anweisung zugelassen oder verweigert werden.
- Action gibt die zu steuernden API-Operationen an. Beispielsweise können Sie mit `elasticbeanstalk:CreateEnvironment` den Vorgang `CreateEnvironment` angeben. Bestimmte Vorgänge, wie z. B. das Erstellen einer Umgebung, erfordern zusätzliche Berechtigungen für die Aktionsausführung. Weitere Informationen finden Sie unter Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen.

  > **Note**
  >
  > Geben Sie zur Verwendung der API-Operation `UpdateTagsForResource` anstelle des Namens der API-Operation eine (oder beide) der folgenden beiden virtuellen Aktionen an:
  >
  > `elasticbeanstalk:AddTags`
  >
  > Steuert die Berechtigung zum Aufrufen von `UpdateTagsForResource` und übergeben einer Liste mit Tags, die dem Parameter `TagsToAdd` hinzugefügt werden.
  >
  > `elasticbeanstalk:RemoveTags`
  >
  > Steuert die Berechtigung zum Aufrufen von `UpdateTagsForResource` und übergeben einer Liste mit Tag-Schlüsseln, die aus dem Parameter `TagsToRemove` entfernt werden.

- Resource gibt die Ressourcen an, bei denen der Zugriff gesteuert werden soll. Um Elastic Beanstalk-Ressourcen anzugeben, führen Sie die Amazon-Ressourcenname (ARN) jeder Ressource auf.

- (Optional) Condition gibt Einschränkungen für die in der Anweisung erteilten Berechtigungen an. Weitere Informationen finden Sie unter Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen.

Die folgenden Abschnitte zeigen einige Fälle, in denen Sie eine benutzerdefinierte Benutzerrichtlinie in Betracht ziehen könnten.

Begrenzte Erstellung einer Elastic Beanstalk-Umgebung ermöglichen

Die Richtlinie im folgenden Beispiel ermöglicht einem Benutzer, die `CreateEnvironment`-Aktion aufzurufen, um eine Umgebung zu erstellen, deren Name mit **`Test`** beginnt, mit der angegebenen Anwendung und Anwendungsversion.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"CreateEnvironmentPerm",
      "Action": [
        "elasticbeanstalk:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My First
 Elastic Beanstalk Application/Test*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My First Elastic Beanstalk Application"],
          "elasticbeanstalk:FromApplicationVersion":
 ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My First
 Elastic Beanstalk Application/First Release"]
        }
      }
    },
    {
      "Sid":"AllNonResourceCalls",
      "Action":[
```

```
        "elasticbeanstalk:CheckDNSAvailability",
        "elasticbeanstalk:CreateStorageLocation"
      ],
      "Effect":"Allow",
      "Resource":[
        "*"
      ]
    }
  ]
}
```

Die obige Richtlinie zeigt, wie man beschränkten Zugang zu Elastic Beanstalk-Operationen gewährt. Um eine Umgebung tatsächlich zu starten, muss der Benutzer über die Berechtigung verfügen, die AWS Ressourcen zu erstellen, die die Umgebung ebenfalls mit Strom versorgen. Beispielsweise wird mit der folgenden Richtlinie Zugriff auf den Standardsatz der Ressourcen für eine Webserverumgebung gewährt:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "ecs:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "cloudformation:*",
        "sqs:*"
        ],
      "Resource": "*"
    }
  ]
}
```

Zugriff auf Elastic Beanstalk-Protokolle in Amazon S3 ermöglichen

Die Richtlinie im folgenden Beispiel ermöglicht es einem Benutzer, Elastic Beanstalk-Protokolle abzurufen, sie in Amazon S3 bereitzustellen und abzurufen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:DeleteObject",
        "s3:GetObjectAcl",
        "s3:PutObjectAcl"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::elasticbeanstalk-*"
    }
  ]
}
```

> **Note**
>
> Um diese Berechtigungen nur auf die Protokollpfade einzuschränken, verwenden Sie das folgende Ressourcenformat.
>
> ```
> "arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/resources/environments/
> logs/*"
> ```

Verwaltung einer bestimmten Elastic Beanstalk-Anwendung ermöglichen

Die Richtlinie im folgenden Beispiel ermöglicht einem Benutzer das Verwalten von Umgebungen und anderen Ressourcen innerhalb einer bestimmten Elastic Beanstalk-Anwendung. Die Richtlinie verweigert Elastic Beanstalk-Aktionen auf Ressourcen anderer Anwendungen und verweigert auch die Erstellung und Löschung von Elastic Beanstalk-Anwendungen.

> **Note**
>
> Die Richtlinie verweigert nicht den Zugriff auf Ressourcen durch andere Services. Es zeigt eine effektive Möglichkeit zum Verteilen von Verantwortlichkeiten für die Verwaltung von Elastic Beanstalk-Anwendungen über verschiedene Benutzer, nicht als Möglichkeit zum Sichern der zugrundeliegenden Ressourcen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:CreateApplication",
        "elasticbeanstalk:DeleteApplication"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:CreateApplicationVersion",
        "elasticbeanstalk:CreateConfigurationTemplate",
        "elasticbeanstalk:CreateEnvironment",
        "elasticbeanstalk:DeleteApplicationVersion",
        "elasticbeanstalk:DeleteConfigurationTemplate",
        "elasticbeanstalk:DeleteEnvironmentConfiguration",
        "elasticbeanstalk:DescribeApplicationVersions",
        "elasticbeanstalk:DescribeConfigurationOptions",
        "elasticbeanstalk:DescribeConfigurationSettings",
        "elasticbeanstalk:DescribeEnvironmentResources",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticbeanstalk:DescribeEvents",
        "elasticbeanstalk:DeleteEnvironmentConfiguration",
        "elasticbeanstalk:RebuildEnvironment",
        "elasticbeanstalk:RequestEnvironmentInfo",
```

```
        "elasticbeanstalk:RestartAppServer",
        "elasticbeanstalk:RetrieveEnvironmentInfo",
        "elasticbeanstalk:SwapEnvironmentCNAMEs",
        "elasticbeanstalk:TerminateEnvironment",
        "elasticbeanstalk:UpdateApplicationVersion",
        "elasticbeanstalk:UpdateConfigurationTemplate",
        "elasticbeanstalk:UpdateEnvironment",
        "elasticbeanstalk:RetrieveEnvironmentInfo",
        "elasticbeanstalk:ValidateConfigurationSettings"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringNotEquals": {
          "elasticbeanstalk:InApplication": [
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
myapplication"
          ]
        }
      }
    }
  ]
}
```

Amazon-Ressourcennamen-Format für Elastic Beanstalk

Sie geben eine Ressource für eine IAM-Richtlinie an, wobei Sie den Amazon-Ressourcenname (ARN) der Ressource verwenden. Für Elastic Beanstalk hat der ARN das folgende Format.

```
arn:aws:elasticbeanstalk:region:account-id:resource-type/resource-path
```

Wobei Folgendes gilt:

- *region* ist die Region, in der sich die Ressource befindet (z. B. **us-west-2**).
- *account-id*ist die AWS Konto-ID ohne Bindestriche (zum Beispiel) **123456789012**
- *resource-type* identifiziert den Typ der Elastic Beanstalk-Ressource, z. B. `environment`. In der nachstehenden Tabelle finden Sie eine Liste aller Ressourcentypen von Elastic Beanstalk.
- *resource-path* ist der Teil, der die jeweilige Ressource über den Namen identifiziert. Eine Ressource vom Typ Elastic Beanstalk hat einen Pfad, der diese Ressource eindeutig identifiziert.

In der folgenden Tabelle finden Sie das Format des Ressourcenpfads für jeden Ressourcentyp. Eine Umgebung ist beispielsweise immer einer Anwendung zugeordnet. Der Ressourcenpfad für die Umgebung **myEnvironment** in der Anwendung **myApp** würde wie folgt aussehen:

```
myApp/myEnvironment
```

Elastic Beanstalk verfügt über mehrere Arten von Ressourcen, die Sie in einer Richtlinie angeben können. In der folgenden Tabelle sind die jeweiligen ARN-Formate für jeden Ressourcentyp sowie ein Beispiel aufgeführt.

Ressourcentyp	Format für ARN
application	arn:aws:elasticbeanstalk: *region*:*account-id* :application/ *application-name* Beispiel: **arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App**
applicationversion	arn:aws:elasticbeanstalk: *region*:*account-id* :applicationversion/ *application-name* /*version-label* Beispiel: **arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version**
configurationtemplate	arn:aws:elasticbeanstalk: *region*:*account-id* :configurationtemplate/ *application-name* /*template-name* Beispiel: **arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template**
environment	arn:aws:elasticbeanstalk: *region*:*account-id* :environment/ *application-name* /*environment-name* Beispiel: **arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/MyEnvironment**

Ressourcentyp	Format für ARN
platform	`arn:aws:elasticbeanstalk:` *region*`:`*account-id*` :platform/`*platform-name*`/`*platform-version* Beispiel: **arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/MyPlatform/1.0**
solutionstack	`arn:aws:elasticbeanstalk:` *region*`::solutionstack/`*solutionstack-name* Beispiel: **arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7**

Eine Umgebung, Anwendungsversion und eine Konfigurationsvorlage sich immer in einer bestimmten Anwendung vorhanden. Sie werden feststellen, dass diese Ressourcen alle über einen Anwendungsnamen in ihrem Ressourcenpfad verfügen, damit sie eindeutig über ihren Ressourcennamen und die Anwendung, die sie enthält, identifiziert werden können. Lösungsstapel werden zwar von Konfigurationsvorlagen und Umgebungen verwendet, Lösungsstapel sind jedoch nicht anwendungs- oder AWS kontospezifisch und enthalten auch nicht die Anwendung oder AWS das Konto. ARNs

Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen

In diesem Abschnitt werden die Ressourcen und Bedingungen beschrieben, die Sie in Richtlinienanweisungen zum Erteilen von Berechtigungen verwenden können, mit denen spezifische Elastic Beanstalk-Aktionen auf bestimmten Elastic Beanstalk-Ressourcen ausgeführt werden können.

Mit Bedingungen können Sie Berechtigungen für Ressourcen festlegen, die die Aktion ausführen muss. Wenn Sie beispielsweise die `CreateEnvironment`-Aktion aufrufen, müssen Sie auch die bereitzustellende Anwendungsversion und die Anwendung angeben, die den Anwendungsnamen enthält. Wenn Sie Berechtigungen für die `CreateEnvironment`-Aktion festlegen, geben Sie die Anwendung und die Anwendungsversion an, auf die die Aktion einwirken soll, indem Sie die `InApplication`- und `FromApplicationVersion`-Bedingungen verwenden.

Außerdem können Sie die Umgebungskonfiguration mit einem Lösungs-Stack (`FromSolutionStack`) oder einer Konfigurationsvorlage (`FromConfigurationTemplate`)

verwenden. Mit der folgenden Richtlinienanweisung kann die `CreateEnvironment`-Aktion eine Umgebung mit Namen **`myenv`** (angegeben durch `Resource`) in der Anwendung **`My App`** (durch die `InApplication`-Bedingung angegeben) erstellen. Dabei wird die Anwendungsversion **`My Version`** (`FromApplicationVersion`) mit einer **`32bit Amazon Linux running Tomcat 7`**-Konfiguration (`FromSolutionStack`) verwendet:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
myenv"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
          "elasticbeanstalk:FromApplicationVersion":
 ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My
 Version"],
          "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-
east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]
        }
      }
    }
  ]
}
```

> **Note**
>
> Die meisten Bedingungsschlüssel in diesem Thema gelten speziell für Elastic Beanstalk und ihre Namen enthalten das Präfix `elasticbeanstalk:`. Aus Gründen der Übersichtlichkeit wird dieses Präfix der Bedingungsschlüsselnamen in den folgenden Abschnitten

ausgelassen. Wir geben also zum Beispiel `InApplication` anstelle des vollständigen Namens `elasticbeanstalk:InApplication` an.
Im Gegensatz dazu erwähnen wir einige Bedingungsschlüssel, die AWS dienstübergreifend verwendet werden, und wir fügen ihr `aws:` Präfix hinzu, um die Ausnahme hervorzuheben.
In Richtlinienbeispielen werden die stets die vollständigen Bedingungsschlüssel mit Präfix angezeigt.

Sections

- Richtlinieninformationen für Elastic Beanstalk-Aktionen
- Bedingungsschlüssel für Elastic Beanstalk-Aktionen

Richtlinieninformationen für Elastic Beanstalk-Aktionen

In der folgenden Tabelle werden alle Elastic Beanstalk-Aktionen, die Ressource, auf die jede Aktion einwirkt, und zusätzliche Kontextinformationen aufgelistet, die mithilfe von Bedingungen bereitgestellt werden können.

Richtlinieninformationen für Elastic Beanstalk-Aktionen, einschließlich Ressourcen, Bedingungen, Beispiele und Abhängigkeiten

Ressource	Bedingungen	Beispielanweisung
Aktion: `AbortEnvironmentUpdate`		

`application`
 	
`environment`
	`aws:ResourceTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Mit der folgenden Richtlinie können Benutzer Vorgänge zur Aktualisierung einer Umgebung auf Umgebungen in einer Anwendung namens `My App` abbrechen.
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:AbortEnvironmentUpdate"`
		`  ],`

Ressource	Bedingungen	Beispielanweisung
		<pre>"Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] }</pre>
Aktion: CheckDNSAvailability		

"*"	–	<pre>{
		"Version":"2012-10-17",
		"Statement": [
		{
		"Action": [
		"elasticbeanstalk:CheckDNSAvailability"
		],
		"Effect": "Allow",
		"Resource": "*"
		}
		]
		}</pre>

Aktion: ComposeEnvironments		

Ressource	Bedingungen	Beispielanweisung
`application`	`aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt einem Benutzer, Umgebungen aufzubauen, die zu einer Anwendung mit dem Namen `My App` gehören. **{** **"Version":"2012-10-17",** **"Statement": [** **{** **"Action": [** **"elasticbeanstalk:ComposeEnvironments"** **],** **"Effect": "Allow",** **"Resource": [** **"arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/*"** **]** **}** **]** **}**
Aktion: `CreateApplication`		

Ressource	Bedingungen	Beispielanweisung
`application`	`aws:RequestTag/` *`key-name`* (Optional) `aws:TagKeys` (Optional)	In diesem Beispiel kann die `CreateApplication` -Aktion Anwendungen erstellen, deren Namen mit **`DivA`** beginnen: `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:CreateApplication" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/DivA*" ] } ] }`
Aktion: `CreateApplicationVersion`		

Ressource	Bedingungen	Beispielanweisung
`applicationversion`	`InApplication` `aws:RequestTag/` *`key-name`* (Optional) `aws:TagKeys` (Optional)	In diesem Beispiel kann die `CreateApplicationVersion` -Aktion Anwendungsversionen mit einem beliebigen Namen (*****) in der Anwendung **`My App`** erstellen: `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateApplicationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/*"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`

Aktion: `CreateConfigurationTemplate`

Ressource	Bedingungen	Beispielanweisung

`configurationtemplate`	`InApplication`
	
	`FromApplication`
	
	`FromApplicationVersion`
	
	`FromConfigurationTemplate`
	
	`FromEnvironment`
	
	`FromSolutionStack`
	
	`aws:RequestTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Mit der folgenden Richtlinie kann die `CreateConfigurationTemplate` -Aktion Konfigurationsvorlagen, deren Name mit **`My Template`** (`My Template*`) beginnt, in der Anwendung **`My App`** erstellen:
		
		`{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateConfigurationTemplate"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template*"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"], "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"] } } }] }`

Aktion: `CreateEnvironment`		

Ressource	Bedingungen	Beispielanweisung

`environment`	`InApplication`
	
	`FromApplicationVersion`
	
	`FromConfigurationTemplate`
	
	`FromSolutionStack`
	
	`aws:RequestTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Die folgende Richtlinie erlaubt der `CreateEnvironment` -Aktion die Erstellung einer Umgebung mit dem Namen **`myenv`** in der Anwendung **`My App`** sowie die Verwendung des Lösungs-Stacks **`32bit Amazon Linux running Tomcat 7`**
		
		`{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"], "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"], "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"] } } }]`

Ressource	Bedingungen	Beispielanweisung
		`}`
Aktion: `CreatePlatformVersion`		
`platform`	`aws:RequestTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	In diesem Beispiel darf die `CreatePlatformVersion` -Aktion Plattformversionen erstellen, die auf die `us-east-2` -Region ausgerichtet sind und deren Name mit **`us-east-2_`** beginnt: `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:CreatePlatformVersion" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*" ] } ] }`
Aktion: `CreateStorageLocation`		

Ressource	Bedingungen	Beispielanweisung
`"*"`	–	`{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:CreateStorageLocation" ], "Effect": "Allow", "Resource": "*" } ] }`
Aktion: `DeleteApplication`		
`application`	`aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Mit dieser Richtlinie kann die `DeleteApplication` -Aktion die Anwendung **`My App`** löschen: `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DeleteApplication" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Aktion: `DeleteApplicationVersion`		

Ressource	Bedingungen	Beispielanweisung
`applicationversion`	`InApplication` `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Mit der folgenden Richtlinie kann die `DeleteApplicationVersion` -Aktion eine Anwendungsversion, deren Name **`My Version`** lautet, in der Anwendung **`My App`** löschen: `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DeleteApplicationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Aktion: `DeleteConfigurationTemplate`		

Ressource	Bedingungen	Beispielanweisung
`configurationtemplate`	`InApplication` (Optional) `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Mit der folgenden Richtlinie kann die `DeleteConfigurationTemplate` -Aktion eine Konfigurationsvorlage, deren Name **`My Template`** lautet, in der Anwendung **`My App`** löschen. Die Angabe des Anwendungsnamens als Bedingung ist optional. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DeleteConfigurationTemplate" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template" ] } ] }`

Aktion: `DeleteEnvironmentConfiguration`

Ressource	Bedingungen	Beispielanweisung
`environment`	`InApplication` (Optional)	Die folgende Richtlinie ermöglicht der `DeleteEnvironmentConfiguration` -Aktion das Löschen eines Konfigurationsentwurfs für die Umgebung **`myenv`** in der Anwendung **`My App`**. Die Angabe des Anwendungsnamens als Bedingung ist optional. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DeleteEnvironmentConfiguration" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`
Aktion: `DeletePlatformVersion`		

Ressource	Bedingungen	Beispielanweisung
`platform`	`aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Mit der folgenden Richtlinie kann die `DeletePlatformVersion` -Aktion Plattformversionen löschen, die auf die `us-east-2` -Region ausgerichtet sind und deren Name mit **`us-east-2_`** beginnt: `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DeletePlatformVersion" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*" ] } ] }`
Aktion: `DescribeApplications`		

Ressource	Bedingungen	Beispielanweisung
`application`	`aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Mit dieser Richtlinie kann die `DescribeApplications` -Aktion die Anwendung "My App" beschreiben. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DescribeApplications" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Aktion: `DescribeApplicationVersions`		

Ressource	Bedingungen	Beispielanweisung
`applicationversion`	`InApplication` (Optional) `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt der `DescribeApplicationVersions` -Aktion, die Anwendungsversion **`My Version`** in der Anwendung **`My App`** zu beschreiben. Die Angabe des Anwendungsnamens als Bedingung ist optional. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DescribeApplicationVersions" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version" ] } ] }`
Aktion: `DescribeConfigurationOptions`		

Ressource	Bedingungen	Beispielanweisung

`environment`
 	
`configurationtemplate`
 	
`solutionstack`
	`InApplication` (Optional)
	
	`aws:ResourceTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Die folgende Richtlinie ermöglicht der `DescribeConfigurationOptions` -Aktion das Beschreiben der Konfigurationsoptionen für die Umgebung **`myenv`** in der Anwendung **`My App`**. Die Angabe des Anwendungsnamens als Bedingung ist optional.
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": "elasticbeanstalk:DescribeConfigurationOptions",`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"`
		`  ]`
		`  }`
		`  ]`
		`}`

Aktion: `DescribeConfigurationSettings`		

Ressource	Bedingungen	Beispielanweisung

`environment`
 	
`configurationtemplate`
	`InApplication` (Optional)
	
	`aws:ResourceTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Die folgende Richtlinie ermöglicht der `DescribeConfigurationSettings` -Aktion das Beschreiben der Konfigurationseinstellungen für die Umgebung **`myenv`** in der Anwendung **`My App`**. Die Angabe des Anwendungsnamens als Bedingung ist optional.
		
		`{ "Version":"2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeConfigurationSettings", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`

Aktion: `DescribeEnvironmentHealth`		

Ressource	Bedingungen	Beispielanweisung
`environment`	`aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Mit der folgenden Richtlinie wird die Verwendung von `DescribeEnvironmentHealth` zum Abrufen von Zustandsinformationen für eine Umgebung namens **myenv** zugelassen. `{ "Version":"2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeEnvironmentHealth", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`

Aktion: `DescribeEnvironmentResources`

Ressource	Bedingungen	Beispielanweisung
`environment`	`InApplication` (Optional) `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie ermöglicht es der `DescribeEnvironmentResources` Aktion, eine Liste von AWS Ressourcen für die Umgebung **`myenv`** in der Anwendung zurückzugeben**`My App`**. Die Angabe des Anwendungsnamens als Bedingung ist optional. `{ "Version":"2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeEnvironmentResources", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`
Aktion: `DescribeEnvironments`		

Ressource	Bedingungen	Beispielanweisung
`environment`	`InApplication` (Optional) `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt der `DescribeEnvironments` -Aktion, die Umgebungen **`myenv`** und **`myotherenv`** in der Anwendung **`My App`** zu beschreiben. Die Angabe des Anwendungsnamens als Bedingung ist optional. `{ "Version":"2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeEnvironments", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv", "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App2/myotherenv" ] } ] }`
Aktion: `DescribeEvents`		

Ressource	Bedingungen	Beispielanweisung

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
	`InApplication`
	
	`aws:ResourceTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Die folgende Richtlinie ermöglicht der `DescribeEvents` -Aktion das Auflisten von Ereignisbeschreibungen für die Umgebung **`myenv`** und die Anwendungsversion **`My Version`** in der Anwendung **`My App`**.
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`    {`
		`      "Action": "elasticbeanstalk:DescribeEvents",`
		`      "Effect": "Allow",`
		`      "Resource": [`
		`        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv",`
		`        "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"`
		`      ],`
		`      "Condition": {`
		`        "StringEquals": {`
		`          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]`
		`        }`
		`      }`
		`    }`
		`  ]`
		`}`

Aktion: `DescribeInstancesHealth`		

Ressource	Bedingungen	Beispielanweisung

`environment`	–	Mit der folgenden Richtlinie wird die Verwendung von `DescribeInstancesHealth` zum Abrufen von Zustandsinformationen für Instances in einer Umgebung namens **`myenv`** zugelassen.
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": "elasticbeanstalk:DescribeInstancesHealth",`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"`
		`  ]`
		`  }`
		`  ]`
		`}`

Aktion: `DescribePlatformVersion`		

Ressource	Bedingungen	Beispielanweisung

`platform`	`aws:ResourceTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Die folgende Richtlinie ermöglicht der `DescribePlatformVersion` -Aktion, Plattformversionen mit Ausrichtung auf die `us-east-2` -Region zu beschreiben, deren Name mit **`us-east-2_`** beginnt:
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:DescribePlatformVersion"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"`
		`  ]`
		`  }`
		`  ]`
		`}`

Aktion: `ListAvailableSolutionStacks`		

Ressource	Bedingungen	Beispielanweisung
`solutionstack`	–	Mit der folgenden Richtlinie kann die `ListAvailableSolutionStacks` -Aktion nur den Lösungs-Stack **`32bit Amazon Linux running Tomcat 7`** zurückgeben. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:ListAvailableSolutionStacks" ], "Effect": "Allow", "Resource": "arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7" } ] }`
Aktion: `ListPlatformVersions`		

Ressource	Bedingungen	Beispielanweisung

`platform`	`aws:RequestTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		In diesem Beispiel darf die `CreatePlatformVersion` -Aktion Plattformversionen erstellen, die auf die `us-east-2` -Region ausgerichtet sind und deren Name mit **`us-east-2_`** beginnt:
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:ListPlatformVersions"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"`
		`  ]`
		`  }`
		`  ]`
		`}`

Aktion: `ListTagsForResource`		

Ressource	Bedingungen	Beispielanweisung

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
 	
`platform`
	`aws:ResourceTag/` *`key-name`* (Optional)
	
	`aws:TagKeys` (Optional)
		Die folgende Richtlinie ermöglicht der `ListTagsForResource` -Aktion das Auflisten von Tags von vorhandenen Umgebungen nur, wenn sie über ein Tag mit Namen `stage` mit dem Wert `test` verfügen:
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:ListTagsForResource"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": "*",`
		`  "Condition": {`
		`  "StringEquals": {`
		`  "aws:ResourceTag/stage": ["test"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Aktion: `RebuildEnvironment`		

Ressource	Bedingungen	Beispielanweisung
`environment`	`InApplication` `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt der `RebuildEnvironment` -Aktion, die Umgebung **`myenv`** in der Anwendung **`My App`** neu zu erstellen. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RebuildEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`

Aktion: `RequestEnvironmentInfo`

Ressource	Bedingungen	Beispielanweisung
`environment`	`InApplication` `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt der `RequestEnvironmentInfo` -Aktion, Informationen zur Umgebung **`myenv`** in der Anwendung **`My App`** zusammenzustellen. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RequestEnvironmentInfo"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Aktion: `RestartAppServer`		

Ressource	Bedingungen	Beispielanweisung

`environment`	`InApplication`	Die folgende Richtlinie ermöglicht der `RestartAppServer` -Aktion, den Anwendungscontainer-Server für die Umgebung **`myenv`** in der Anwendung **`My App`** neu zu starten.
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:RestartAppServer"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"`
		`  ],`
		`  "Condition": {`
		`  "StringEquals": {`
		`  "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Aktion: `RetrieveEnvironmentInfo`		

Ressource	Bedingungen	Beispielanweisung
`environment`	`InApplication` `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt der `RetrieveEnvironmentInfo` -Aktion, die zusammengestellten Informationen über die Umgebung **`myenv`** in der Anwendung **`My App`** abzurufen. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RetrieveEnvironmentInfo"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Aktion: `SwapEnvironmentCNAMEs`		

Ressource	Bedingungen	Beispielanweisung
`environment`	`InApplication` (Optional) `FromEnvironment` (Optional)	Die folgende Richtlinie ermöglicht es der `SwapEnvironmentCNAMEs` Aktion, die CNAMEs gegen die Umgebungen **`mysrcenv`** und auszutauschen **`mydestenv`** . `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:SwapEnvironmentCNAMEs" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/mysrcenv", "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/mydestenv" ] } ] }`
Aktion: `TerminateEnvironment`		

Ressource	Bedingungen	Beispielanweisung
`environment`	`InApplication` `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt der `TerminateEnvironment` -Aktion, die Umgebung **`myenv`** in der Anwendung **`My App`** zu beenden. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:TerminateEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`

Aktion: UpdateApplication

Ressource	Bedingungen	Beispielanweisung
`application`	`aws:ResourceTag/` *`key-name`* (Optional) `aws:TagKeys` (Optional)	Mit dieser Richtlinie kann die `UpdateApplication` -Aktion Eigenschaften der Anwendung **`My App`** aktualisieren. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:UpdateApplication" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Aktion: UpdateApplicationResourceLifecycle		

Ressource	Bedingungen	Beispielanweisung
`application`	`aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt der `UpdateApplicationResourceLifecycle` -Aktion das Aktualisieren von Lebenszykluseinstellungen der Anwendung **`My App`**. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:UpdateApplicationResourceLifecycle" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Aktion: `UpdateApplicationVersion`		

Ressource	Bedingungen	Beispielanweisung
`applicationversion`	`InApplication` `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt der `UpdateApplicationVersion` -Aktion, die Eigenschaften der Anwendungsversion **`My Version`** in der Anwendung **`My App`** zu aktualisieren. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateApplicationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Aktion: `UpdateConfigurationTemplate`		

Ressource	Bedingungen	Beispielanweisung
`configurationtemplate`	`InApplication` `aws:ResourceTag/` *key-name* (Optional) `aws:TagKeys` (Optional)	Die folgende Richtlinie erlaubt der `UpdateConfigurationTemplate` -Aktion, die Eigenschaften oder Optionen der Konfigurationsvorlage **`My Template`** in der Anwendung **`My App`** zu aktualisieren. **`{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateConfigurationTemplate"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`**

Aktion: `UpdateEnvironment`

Ressource	Bedingungen	Beispielanweisung

`environment`	`InApplication`
	
	`FromApplicationVersion`
	
	`FromConfigurationTemplate`
	
	`aws:ResourceTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Die folgende Richtlinie ermöglicht der `UpdateEnvironment` -Aktion das Aktualisieren der Umgebung **`myenv`** in der Anwendung **`My App`**, indem die Anwendungsversion **`My Version`** bereitgestellt wird.
		
		`{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"], "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"] } } }] }`

Aktion: **`UpdateTagsForResource`** — `AddTags`		

Ressource	Bedingungen	Beispielanweisung

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
 	
`platform`
	`aws:ResourceTag/` *key-name* (Optional)
	
	`aws:RequestTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Die `AddTags`-Aktion ist eine von zwei virtuellen Aktionen, die mit der `UpdateTagsForResource` -API verknüpft sind.
		
		Die folgende Richtlinie ermöglicht der `AddTags`-Aktion das Ändern von Tags von vorhandenen Umgebungen nur, wenn sie über ein Tag mit Namen `stage` mit dem Wert `test` verfügen:
		
		`{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:AddTags" ], "Effect": "Allow", "Resource": "*", "Condition": { "StringEquals": { "aws:ResourceTag/stage": ["test"] } } } ] }`

Aktion: **`UpdateTagsForResource`** — `RemoveTags`		

Ressource	Bedingungen	Beispielanweisung

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
 	
`platform`
	`aws:ResourceTag/` *`key-name`* (Optional)
	
	`aws:TagKeys` (Optional)
		Die `RemoveTags` -Aktion ist eine von zwei virtuellen Aktionen, die mit der `UpdateTagsForResource` -API verknüpft sind.
		
		Die folgende Richtlinie verweigert der `RemoveTags` -Aktion, die Entfernung eines Tags mit Namen `stage` aus bestehenden Umgebungen anzufordern:
		
		{
		"Version":"2012-10-17",
		"Statement": [
		{
		"Action": [
		"elasticbeanstalk:RemoveTags"
		],
		"Effect": "Deny",
		"Resource": "*",
		"Condition": {
		"ForAnyValue:StringEquals": {
		"aws:TagKeys": ["stage"]
		}
		}
		}
		]
		}

Aktion: `ValidateConfigurationSettings`		

Ressource	Bedingungen	Beispielanweisung

`template`
 	
`environment`
	`InApplication`
	
	`aws:ResourceTag/` *key-name* (Optional)
	
	`aws:TagKeys` (Optional)
		Mit dieser Richtlinie kann die `ValidateConfigurationSettings` -Aktion die Konfigurationseinstellungen für die Umgebung **`myenv`** in der Anwendung **`My App`** überprüfen.
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:ValidateConfigurationSettings"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"`
		`  ],`
		`  "Condition": {`
		`  "StringEquals": {`
		`  "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Bedingungsschlüssel für Elastic Beanstalk-Aktionen

Mit Schlüsseln können Sie Bedingungen festlegen, die Abhängigkeiten ausdrücken, Berechtigungen beschränken oder Einschränkungen bei den Eingabeparametern für eine Aktion angeben. Elastic Beanstalk unterstützt die folgenden Schlüssel.

`InApplication`

Gibt die Anwendung an, die die Ressource enthält, auf der die Aktion ausgeführt wird.

Im folgenden Beispiel kann die `UpdateApplicationVersion`-Aktion die Eigenschaften der Anwendungsversion **`My Version`** aktualisieren. Die `InApplication`-Bedingung gibt **`My App`** als Container für **`My Version`** an.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:UpdateApplicationVersion"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My
 App/My Version"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"]
        }
      }
    }
  ]
}
```

`FromApplicationVersion`

Gibt eine Anwendungsversion als eine Abhängigkeit oder eine Einschränkung auf einem Eingabeparameter an.

Das folgende Beispiel erlaubt der `UpdateEnvironment`-Aktion, die Umgebung **`myenv`** in der Anwendung **`My App`** zu aktualisieren. Die `FromApplicationVersion`-Bedingung schränkt den `VersionLabel`-Parameter so ein, dass nur die Anwendungsversion **`My Version`** die Umgebung aktualisieren kann.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:UpdateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
myenv"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
          "elasticbeanstalk:FromApplicationVersion":
 ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/
My Version"]
        }
      }
    }
  ]
}
```

`FromConfigurationTemplate`

Gibt eine Konfigurationsvorlage als eine Abhängigkeit oder eine Einschränkung auf einem Eingabeparameter an.

Das folgende Beispiel erlaubt der `UpdateEnvironment`-Aktion, die Umgebung **`myenv`** in der Anwendung **`My App`** zu aktualisieren. Die `FromConfigurationTemplate`-Bedingung schränkt den `TemplateName`-Parameter so ein, dass nur die Konfigurationsvorlage **`My Template`** die Umgebung aktualisieren kann.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Action": [
        "elasticbeanstalk:UpdateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
myenv"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
          "elasticbeanstalk:FromConfigurationTemplate":
 ["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My
 App/My Template"]
        }
      }
    }
  ]
}
```

`FromEnvironment`

Gibt eine Umgebung als eine Abhängigkeit oder eine Einschränkung auf einem Eingabeparameter an.

Das folgende Beispiel ermöglicht es der `SwapEnvironmentCNAMEs` Aktion, die Eingabe CNAMEs **`My App`** für alle Umgebungen auszutauschen, deren Namen mit **`mysrcenv`** und beginnen, **`mydestenv`** aber nicht für Umgebungen, deren Namen mit **`mysrcenvPROD*`** und beginnen**`mydestenvPROD*`**.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:SwapEnvironmentCNAMEs"
      ],
      "Effect": "Allow",
      "Resource": [
```

```
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
mysrcenv*",
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
mydestenv*"
      ],
      "Condition": {
        "ArnNotLike": {
          "elasticbeanstalk:FromEnvironment": [
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My
 App/mysrcenvPROD*",
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My
 App/mydestenvPROD*"
          ]
        }
      }
    }
  ]
}
```

`FromSolutionStack`

Gibt einen Lösungs-Stack als eine Abhängigkeit oder eine Einschränkung auf einem Eingabeparameter an.

Mit der folgenden Richtlinie kann die `CreateConfigurationTemplate`-Aktion Konfigurationsvorlagen, deren Name mit **`My Template`** (`My Template*`) beginnt, in der Anwendung **`My App`** erstellen. Die `FromSolutionStack`-Bedingung schränkt den `solutionstack`-Parameter so ein, dass nur der Lösungs-Stack **`32bit Amazon Linux running Tomcat 7`** als Eingabewert für diesen Parameter zulässig ist.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:CreateConfigurationTemplate"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-
east-2:123456789012:configurationtemplate/My App/My Template*"
```

```
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
          "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-
east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]
        }
      }
    }
  ]
}
```

`aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, `aws:TagKeys`

Geben Sie Tag-basierte Bedingungen an. Details hierzu finden Sie unter Tags zur Steuerung des Zugriffs auf Elastic Beanstalk-Ressourcen verwenden.

Tags zur Steuerung des Zugriffs auf Elastic Beanstalk-Ressourcen verwenden

In diesem Thema wird erklärt, wie die tagbasierte Zugriffskontrolle Sie bei der Erstellung und Verwaltung von IAM-Richtlinien unterstützen kann.

Wir können Bedingungen in IAM-Benutzerrichtlinienerklärungen verwenden, um die Berechtigungen für den Zugriff von Elastic Beanstalk auf Ressourcen zu konfigurieren. Weitere Informationen zu den Bedingungen für Richtlinienerklärungen finden Sie unter. Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen Das Verwenden von Tags in Bedingungen ist eine Möglichkeit zur Kontrolle des Zugriffs auf Ressourcen und Anfragen. Informationen zum Tagging von Elastic Beanstalk-Ressourcen finden Sie unter Markieren von Elastic Beanstalk-Anwendungsressourcen.

Wenn Sie IAM-Richtlinien entwerfen, können Sie präzise abgestufte Berechtigungen festlegen, indem Sie Zugriff auf bestimmte Ressourcen gewähren. Mit zunehmender Anzahl der Ressourcen, die Sie verwalten, wird diese Aufgabe erschwert. Durch Markieren von Ressourcen und Verwenden von Tags in Richtlinienanweisungsbedingungen lässt sich diese Aufgabe vereinfachen. Sie erteilen Massenzugriff auf eine beliebige Ressource mit einem bestimmten Tag. Anschließend wenden Sie dieses Tag während der Erstellung oder zu einem späteren Zeitpunkt wiederholt auf relevante Ressourcen an.

Markierungen können an die Ressource angehängt oder in der Anfrage an Services übergeben werden, die das Markieren unterstützen. In Elastic Beanstalk können Ressourcen Tags haben, und einige Aktionen können Tags enthalten. Wenn Sie eine IAM-Richtlinie erstellen, können Sie Tag-Bedingungsschlüssel verwenden, um die folgenden Bedingungen zu steuern:

- Welche Benutzer Aktionen für eine Umgebung ausführen können, basierend auf bereits vorhandenen Tags.
- Welche Tags in der Anforderung einer Aktion übergeben werden können.
- Ob bestimmte Tag-Schlüssel in einer Anforderung verwendet werden können.

Die vollständige Syntax und Semantik der Tag-Bedingungsschlüssel finden Sie unter Steuern des Zugriffs mit Tags im IAM-Benutzerhandbuch.

Beispiele für Tag-Bedingungen in Richtlinien

Die folgenden Beispiele zeigen, wie Sie Tag-Bedingungen in Richtlinien für Elastic Beanstalk-Benutzer festlegen können.

Example 1: Einschränken von Aktionen basierend auf Tags in der Anforderung

Die Richtlinie für verwaltete Benutzer von Elastic Beanstalk AdministratorAccess— AWSElastic Beanstalk gibt Benutzern uneingeschränkte Rechte, jede Elastic Beanstalk-Aktion auf jeder von Elastic Beanstalk verwalteten Ressource auszuführen.

Mit der folgenden Richtlinie wird nicht autorisierten Benutzern die Berechtigung verweigert, Elastic Beanstalk-Produktionsumgebungen zu erstellen. Dazu verweigert es die `CreateEnvironment`-Aktion, wenn die Anforderung ein Tag mit dem Namen `stage` und einem der Werte `gamma` oder `prod` festlegt. Darüber hinaus verhindert diese Richtlinie, dass nicht autorisierte Benutzer die Phase der Produktionsumgebungen manipulieren, indem sie Tag-Änderungsaktionen verhindert, bei denen die gleichen Tag-Werte betroffen sind oder bei denen das `stage`-Tag vollständig entfernt wird. Der Administrator eines Kunden muss diese IAM-Richtlinie nicht autorisierten IAM-Benutzern hinzufügen, zusätzlich zu der verwalteten Benutzerrichtlinie.

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:CreateEnvironment",
        "elasticbeanstalk:AddTags"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stage": ["gamma", "prod"]
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:RemoveTags"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": ["stage"]
        }
      }
    }
  ]
}
```

Example 2: Einschränken von Aktionen basierend auf Ressourcen-Tags

Die Richtlinie für verwaltete Benutzer von Elastic Beanstalk AdministratorAccess— AWSElastic Beanstalk gibt Benutzern uneingeschränkte Rechte, jede Elastic Beanstalk-Aktion auf jeder von Elastic Beanstalk verwalteten Ressource auszuführen.

Mit der folgenden Richtlinie wird Benutzern die Berechtigung verweigert, Aktionen für Elastic Beanstalk-Produktionsumgebungen auszuführen. Dazu verweigert es bestimmte Aktionen, wenn die Umgebung über ein Tag mit dem Namen `stage` und einem der Werte `gamma` oder `prod` verfügt. Der Administrator eines Kunden muss diese IAM-Richtlinie nicht autorisierten IAM-Benutzern hinzufügen, zusätzlich zu der verwalteten Benutzerrichtlinie.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:AddTags",
        "elasticbeanstalk:RemoveTags",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticbeanstalk:TerminateEnvironment",
        "elasticbeanstalk:UpdateEnvironment",
        "elasticbeanstalk:ListTagsForResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stage": ["gamma", "prod"]
        }
      }
    }
  ]
}
```

Example 3: Zulassen von Aktionen basierend auf Tags in der Anforderung

Mit der folgenden Richtlinie wird Benutzern die Berechtigung zum Erstellen von Elastic Beanstalk-Entwicklungsanwendungen erteilt.

Dazu werden die Aktionen `CreateApplication` und `AddTags` zugelassen, wenn die Anforderung ein Tag mit dem Namen `stage` mit dem Wert `development` angibt. Die `aws:TagKeys`-Bedingung stellt sicher, dass der Benutzer keine anderen Tag-Schlüssel hinzufügen kann. Insbesondere wird sichergestellt, dass die Groß-/Kleinschreibung des `stage`-Tag-Schlüssels berücksichtigt wird. Beachten Sie, dass diese Richtlinie für IAM-Benutzer nützlich ist, denen die Richtlinie für verwaltete Benutzer von Elastic Beanstalk AdministratorAccess— AWSElastic Beanstalk nicht angehängt ist. Die verwaltete Richtlinie gibt Benutzern unbegrenzte Berechtigung, jede Elastic Beanstalk-Aktion für jede von Elastic BeanStalk-verwaltete Ressource durchzuführen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:CreateApplication",
        "elasticbeanstalk:AddTags"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stage": "development"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["stage"]
        }
      }
    }
  ]
}
```

Example 4: Zulassen von Aktionen basierend auf Ressourcen-Tags

Mit der folgenden Richtlinie wird Benutzern die Berechtigung zum Ausführen von Aktionen auf und Abrufen von Informationen zu Elastic Beanstalk-Entwicklungsanwendungen gewährt.

Dazu werden bestimmte Aktionen zugelassen, wenn die Anwendung ein Tag mit dem Namen `stage` mit dem Wert `development` enthält. Die `aws:TagKeys`-Bedingung stellt sicher, dass der Benutzer keine anderen Tag-Schlüssel hinzufügen kann. Insbesondere wird sichergestellt, dass die Groß-/Kleinschreibung des `stage`-Tag-Schlüssels berücksichtigt wird. Beachten Sie, dass diese Richtlinie für IAM-Benutzer nützlich ist, denen die Richtlinie für verwaltete Benutzer von Elastic Beanstalk AdministratorAccess— AWSElastic Beanstalk nicht angehängt ist. Die verwaltete Richtlinie gibt Benutzern unbegrenzte Berechtigung, jede Elastic Beanstalk-Aktion für jede von Elastic BeanStalk-verwaltete Ressource durchzuführen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:UpdateApplication",
        "elasticbeanstalk:DeleteApplication",
        "elasticbeanstalk:DescribeApplications"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stage": "development"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["stage"]
        }
      }
    }
  ]
}
```

Beispielrichtlinien basierend auf verwalteten Richtlinien

In diesem Abschnitt wird gezeigt, wie der Benutzerzugriff gesteuert werden kann, AWS Elastic Beanstalk und enthält Beispielrichtlinien, die den erforderlichen Zugriff für allgemeine Szenarien bereitstellen. Diese Richtlinien sind von den von Elastic Beanstalk verwalteten Richtlinien abgeleitet. Weitere Informationen zum Anfügen von verwalteten Richtlinien für Benutzer und Gruppen finden Sie unter Elastic Beanstalk-Benutzerrichtlinien verwalten.

In diesem Szenario ist Example Corp. ein Software-Unternehmen mit drei Teams, die für die Unternehmens-Website verantwortlich sind: Administratoren, die die Infrastruktur verwalten, Entwickler, die die Software für die Website erstellen, und ein QA-Team, das die Website testet. Damit die Berechtigungen für ihre Elastic Beanstalk-Ressourcen einfacher verwaltet werden können, erstellt Example Corp. drei Gruppen, zu denen Mitglieder der jeweiligen Teams gehören: Administratoren, Entwickler und Tester. Example Corp. möchte, dass die Admins-Gruppe

vollständigen Zugriff auf alle Anwendungen, Umgebungen und ihre zugrundeliegenden Ressourcen hat, damit sie alle Elastic Beanstalk-Komponenten erstellen, beheben und löschen können. Entwickler benötigen Berechtigungen zur Anzeige aller Elastic Beanstalk-Komponenten und zum Erstellen und Bereitstellen von Anwendungsversionen. Entwickler sollten nicht in der Lage sein, neue Anwendungen oder Umgebungen zu erstellen oder laufende Umgebungen zu beenden. Tester müssen alle Elastic Beanstalk-Ressourcen anzeigen, um Anwendungen überwachen und testen zu können. Die Tester sollten nicht in der Lage sein, Änderungen an Elastic Beanstalk-Ressourcen vorzunehmen.

Die folgenden Beispielrichtlinien bieten die erforderlichen Berechtigungen für jede Gruppe.

Beispiel 1: Gruppe Admins — All Elastic Beanstalk und zugehörige Dienste APIs

Mit der folgenden Richtlinie erhalten Benutzer Berechtigungen für alle Aktionen, die zur Verwendung von Elastic Beanstalk erforderlich sind. Mit dieser Richtlinie kann Elastic Beanstalk auch Ressourcen in Ihrem Namen in den folgenden Services bereitstellen und verwalten. Elastic Beanstalk nutzt diese zusätzlichen Services zur Bereitstellung zugrunde liegender Ressourcen für die Umgebungserstellung.

- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- Auto Scaling
- Amazon CloudWatch
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon Relational Database Service
- CloudFormation

Beachten Sie, dass diese Richtlinie als Beispiel dient. Sie bietet umfangreiche Berechtigungen für AWS -Services, die von Elastic Beanstalk zur Anwendungs- und Umgebungsverwaltung herangezogen werden. `ec2:*`Ermöglicht beispielsweise einem AWS Identity and Access Management (IAM-) Benutzer, jede Aktion auf einer beliebigen Amazon EC2 EC2-Ressource im AWS Konto auszuführen. Diese Berechtigungen sind nicht auf Ressourcen beschränkt, die mit Elastic Beanstalk verwendet werden. Als bewährte Methode sollten Sie Personen nur die Berechtigungen erteilen, die sie für ihre Aufgaben benötigen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticbeanstalk:*",
        "ec2:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "rds:*",
        "cloudformation:*"
      ],
      "Resource" : "*"
    }
  ]
}
```

Beispiel 2: Entwicklergruppe – Alle Operationen außer solchen mit hoher Privilegierung

Mit der folgenden Richtlinie wird die Berechtigung zum Erstellen von Anwendungen und Umgebungen abgelehnt und alle anderen Elastic Beanstalk-Aktionen werden zugelassen.

Beachten Sie, dass diese Richtlinie als Beispiel dient. Sie bietet umfangreiche Berechtigungen für AWS -Produkte, die von Elastic Beanstalk zur Anwendungs- und Umgebungsverwaltung herangezogen werden. Beispielsweise kann ein IAM-Benutzer mit `ec2:*` alle Aktionen für alle Amazon-EC2-Ressourcen des AWS -Kontos ausführen. Diese Berechtigungen sind nicht auf Ressourcen beschränkt, die mit Elastic Beanstalk verwendet werden. Als bewährte Methode sollten Sie Personen nur die Berechtigungen erteilen, die sie für ihre Aufgaben benötigen.

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement" : [
    {
      "Action" : [
        "elasticbeanstalk:CreateApplication",
        "elasticbeanstalk:CreateEnvironment",
        "elasticbeanstalk:DeleteApplication",
        "elasticbeanstalk:RebuildEnvironment",
        "elasticbeanstalk:SwapEnvironmentCNAMEs",
        "elasticbeanstalk:TerminateEnvironment"],
      "Effect" : "Deny",
      "Resource" : "*"
    },
    {
      "Action" : [
        "elasticbeanstalk:*",
        "ec2:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "rds:*",
        "cloudformation:*"],
      "Effect" : "Allow",
      "Resource" : "*"
    }
  ]
}
```

Beispiel 3: Tester – Nur anzeigen

Mit der folgenden Richtlinie wird Lesezugriff für alle Anwendungen, Anwendungsversionen, Veranstaltungen und Umgebungen zugelassen. Das Ausführen von Aktionen wird nicht zugelassen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
```

```
      "Action" : [
        "elasticbeanstalk:Check*",
        "elasticbeanstalk:Describe*",
        "elasticbeanstalk:List*",
        "elasticbeanstalk:RequestEnvironmentInfo",
        "elasticbeanstalk:RetrieveEnvironmentInfo",
        "ec2:Describe*",
        "elasticloadbalancing:Describe*",
        "autoscaling:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:List*",
        "cloudwatch:Get*",
        "s3:Get*",
        "s3:List*",
        "sns:Get*",
        "sns:List*",
        "rds:Describe*",
        "cloudformation:Describe*",
        "cloudformation:Get*",
        "cloudformation:List*",
        "cloudformation:Validate*",
        "cloudformation:Estimate*"
      ],
      "Resource" : "*"
    }
  ]
}
```

Beispielrichtlinien auf Basis von Ressourcenberechtigungen

In diesem Abschnitt wird ein Anwendungsfall für die Steuerung der Benutzerberechtigungen für Elastic Beanstalk-Aktionen durchgegangen, die auf bestimmte Elastic Beanstalk-Ressourcen zugreifen. Dabei werden die Beispielrichtlinien für diesen Anwendungsfall vorgestellt. Weitere Informationen zu den Richtlinien für Elastic Beanstalk-Ressourcen finden Sie unter Erstellen einer benutzerdefinierten Benutzerrichtlinie. Weitere Informationen zum Verknüpfen von Richtlinien mit Benutzern und Gruppen finden Sie im Abschnitt Verwalten von IAM-Richtlinien unter Verwenden von AWS Identity and Access Management.

In diesem Anwendungsfall entwickelt das kleine Beratungsunternehmen Example Corp. Anwendungen für zwei verschiedene Kunden. Als Entwicklungsmanager ist John für die Entwicklung

der beiden Elastic Beanstalk-Anwendungen "app1" und "app2" zuständig. John entwickelt die beiden Anwendungen und führt einige Tests aus; nur er kann die Produktionsumgebung für die beiden Anwendungen aktualisieren. Für "app1" und "app2" benötigt er folgende Berechtigungen:

- Anzeigen der Anwendung, Anwendungsversionen, Umgebungen und Konfigurationsvorlagen
- Erstellen von Anwendungsversionen sowie deren Bereitstellung in der Staging-Umgebung
- Aktualisieren der Produktionsumgebung
- Erstellen und Beenden von Umgebungen

Jill ist Testerin und muss die folgenden Ressourcen anzeigen können, um die beiden Anwendungen zu überwachen und zu testen: Anwendungen, Anwendungsversionen, Umgebungen und Konfigurationsvorlagen. Sie sollte jedoch nicht in der Lage sein, Änderungen an Elastic Beanstalk-Ressourcen vorzunehmen.

Jack ist der Entwickler von "app1" und muss alle Ressourcen für "app1" anzeigen können. Zudem soll er Anwendungsversionen für "app1" erstellen und diese in der Staging-Umgebung bereitstellen.

Judy ist die Administratorin des AWS Kontos für Example Corp. Sie hat IAM-Benutzer für John, Jill und Jack erstellt und fügt diesen Benutzern die folgenden Richtlinien hinzu, um den Anwendungen app1 und app2 die entsprechenden Berechtigungen zu gewähren.

Beispiel 1: John – Entwicklungsmanager für app1, app2

Wir haben die Richtlinie von John in drei einzelne Richtlinien unterteilt, denn diese können einfacher gelesen und verwaltet werden. Gemeinsam geben sie John die Berechtigungen, die er benötigt, um Entwicklungs-, Test- und Bereitstellungsaktionen für die beiden Anwendungen durchzuführen.

Die erste Richtlinie spezifiziert Aktionen für Auto Scaling, Amazon S3, Amazon EC2, Amazon SNS CloudWatch, Elastic Load Balancing, Amazon RDS und. CloudFormation Elastic Beanstalk nutzt diese zusätzlichen Services zur Bereitstellung zugrunde liegender Ressourcen für die Umgebungserstellung.

Beachten Sie, dass diese Richtlinie als Beispiel dient. Es gewährt eine breite Palette von Berechtigungen für die AWS Produkte, die Elastic Beanstalk zur Verwaltung von Anwendungen und Umgebungen verwendet. `ec2:*`Ermöglicht es einem IAM-Benutzer beispielsweise, jede Aktion auf einer beliebigen Amazon EC2 EC2-Ressource im AWS Konto auszuführen. Diese Berechtigungen sind nicht auf Ressourcen beschränkt, die mit Elastic Beanstalk verwendet werden. Als bewährte Methode sollten Sie Personen nur die Berechtigungen erteilen, die sie für ihre Aufgaben benötigen.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
   "Action": [
   "ec2:*",
   "ecs:*",
   "ecr:*",
   "elasticloadbalancing:*",
   "autoscaling:*",
   "cloudwatch:*",
   "s3:*",
   "sns:*",
   "cloudformation:*",
   "dynamodb:*",
   "rds:*",
   "sqs:*",
   "logs:*",
   "iam:GetPolicyVersion",
   "iam:GetRole",
   "iam:ListRolePolicies",
   "iam:ListAttachedRolePolicies",
   "iam:ListInstanceProfiles",
   "iam:ListRoles",
   "iam:ListServerCertificates",
   "acm:DescribeCertificate",
   "acm:ListCertificates",
   "codebuild:CreateProject",
   "codebuild:DeleteProject",
   "codebuild:BatchGetBuilds",
   "codebuild:StartBuild"
   ],
   "Resource": "*"
   },
   {
   "Effect": "Allow",
   "Action": [
   "iam:PassRole"
   ],
   "Resource": "arn:aws:iam::111122223333:role/MyRole"
```

```
   }
   ]
   }
```

In der zweiten Richtlinie werden die Elastic Beanstalk-Aktionen definiert, die John für die Ressourcen von "app1" und "app2" ausführen kann. Mit der Anweisung `AllCallsInApplications` können alle Elastic Beanstalk-Aktionen (`"elasticbeanstalk:*"`) für alle Ressourcen in "app1" und "app2" ausgeführt werden (z. B. `elasticbeanstalk:CreateEnvironment`). Mit der Anweisung `AllCallsOnApplications` können alle Elastic Beanstalk-Aktionen (`"elasticbeanstalk:*"`) für die Anwendungsressourcen von "app1" und "app2" ausgeführt werden (z. B. `elasticbeanstalk:DescribeApplications`, `elasticbeanstalk:UpdateApplication` usw.). Mit der Anweisung `AllCallsOnSolutionStacks` können alle Elastic Beanstalk-Aktionen (`"elasticbeanstalk:*"`) für Lösungs-Stack-Ressourcen ausgeführt werden (z. B. `elasticbeanstalk:ListAvailableSolutionStacks`).

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllCallsInApplications",
         "Action":[
            "elasticbeanstalk:*"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ],
         "Condition":{
            "StringEquals":{
               "elasticbeanstalk:InApplication":[
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app1",
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app2"
               ]
            }
         }
      },
```

```
      {
         "Sid":"AllCallsOnApplications",
         "Action":[
            "elasticbeanstalk:*"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
         ]
      },
      {
         "Sid":"AllCallsOnSolutionStacks",
         "Action":[
            "elasticbeanstalk:*"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
         ]
      }
   ]
}
```

Die dritte Richtlinie legt die Elastic Beanstalk-Aktionen fest, für die die zweite Richtlinie Berechtigungen benötigt, um diese Elastic Beanstalk-Aktionen abzuschließen. Mit der Anweisung `AllNonResourceCalls` kann die `elasticbeanstalk:CheckDNSAvailability`-Aktion ausgeführt werden, die für den Aufruf von `elasticbeanstalk:CreateEnvironment` und anderen Aktionen erforderlich ist. Zudem kann damit die `elasticbeanstalk:CreateStorageLocation`-Aktion ausgeführt werden, die für `elasticbeanstalk:CreateApplication`, `elasticbeanstalk:CreateEnvironment` und weitere Aktionen benötigt wird.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllNonResourceCalls",
         "Action":[
```

```
            "elasticbeanstalk:CheckDNSAvailability",
            "elasticbeanstalk:CreateStorageLocation"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Beispiel 2: Jill – Tester für app1, app2

Wir haben die Richtlinie von Jill in drei einzelne Richtlinien unterteilt, denn diese können einfacher gelesen und verwaltet werden. Mit diesen drei Richtlinien erhält Jill die benötigten Berechtigungen, um die Aktionen für das Testen und Überwachen der beiden Anwendungen auszuführen.

Die erste Richtlinie spezifiziert `Describe*List*`, und `Get*` Aktionen für Auto Scaling, Amazon S3, Amazon EC2, Amazon SNS CloudWatch, Elastic Load Balancing, Amazon RDS und CloudFormation (für nicht ältere Containertypen), sodass die Elastic Beanstalk Beanstalk-Aktionen die relevanten Informationen über die zugrunde liegenden Ressourcen der App1- und App2-Anwendungen abrufen können.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ec2:Describe*",
            "elasticloadbalancing:Describe*",
            "autoscaling:Describe*",
            "cloudwatch:Describe*",
            "cloudwatch:List*",
            "cloudwatch:Get*",
            "s3:Get*",
            "s3:List*",
            "sns:Get*",
            "sns:List*",
```

```
            "rds:Describe*",
            "cloudformation:Describe*",
         "cloudformation:Get*",
         "cloudformation:List*",
         "cloudformation:Validate*",
         "cloudformation:Estimate*"
         ],
         "Resource":"*"
      }
   ]
}
```

In der zweiten Richtlinie werden die Elastic Beanstalk-Aktionen definiert, die Jill für die Ressourcen von "app1" und "app2" ausführen kann. Mit der Anweisung `AllReadCallsInApplications` können die `Describe*`-Aktionen sowie die Aktionen für Umgebungsinformationen ausgeführt werden. Mit der Anweisung `AllReadCallsOnApplications` können die Aktionen `DescribeApplications` und `DescribeEvents` für die Anwendungsressourcen von "app1" und "app2" ausgeführt werden. Mit der Anweisung `AllReadCallsOnSolutionStacks` können Anzeigeaktionen für Lösungs-Stack-Ressourcen ausgeführt werden (`ListAvailableSolutionStacks`, `DescribeConfigurationOptions` und `ValidateConfigurationSettings`).

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllReadCallsInApplications",
         "Action":[
            "elasticbeanstalk:Describe*",
            "elasticbeanstalk:RequestEnvironmentInfo",
            "elasticbeanstalk:RetrieveEnvironmentInfo"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ],
         "Condition":{
            "StringEquals":{
```

```
               "elasticbeanstalk:InApplication":[
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app1",
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app2"
               ]
            }
         }
      },
      {
         "Sid":"AllReadCallsOnApplications",
         "Action":[
            "elasticbeanstalk:DescribeApplications",
            "elasticbeanstalk:DescribeEvents"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
         ]
      },
      {
         "Sid":"AllReadCallsOnSolutionStacks",
         "Action":[
            "elasticbeanstalk:ListAvailableSolutionStacks",
            "elasticbeanstalk:DescribeConfigurationOptions",
            "elasticbeanstalk:ValidateConfigurationSettings"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
         ]
      }
   ]
}
```

Die dritte Richtlinie legt die Elastic Beanstalk-Aktionen fest, für die die zweite Richtlinie Berechtigungen benötigt, um diese Elastic Beanstalk-Aktionen abzuschließen. Mit der Anweisung `AllNonResourceCalls` kann die `elasticbeanstalk:CheckDNSAvailability`-Aktion ausgeführt werden, die für einige Anzeigeaktionen erforderlich ist.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllNonResourceCalls",
         "Action":[
            "elasticbeanstalk:CheckDNSAvailability"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Beispiel 3: Jack – Entwickler für app1

Wir haben die Richtlinie von Jack in drei einzelne Richtlinien unterteilt, denn diese können einfacher gelesen und verwaltet werden. Gemeinsam geben sie Jack die Berechtigungen, die er benötigt, um Test-, Überwachungs- und Bereitstellungsaktionen auf der app1-Ressource durchzuführen.

Die erste Richtlinie spezifiziert die Aktionen auf Auto Scaling, Amazon S3, Amazon EC2, Amazon SNS CloudWatch, Elastic Load Balancing, Amazon RDS und CloudFormation (für nicht ältere Containertypen), sodass die Elastic Beanstalk Beanstalk-Aktionen die zugrunde liegenden Ressourcen von app1 anzeigen und mit ihnen arbeiten können. Eine Liste mit unterstützten aktuellen Containertypen finden Sie unter the section called “Warum sind einige Plattformversionen als Legacy gekennzeichnet?”.

Beachten Sie, dass diese Richtlinie als Beispiel dient. Es gewährt eine breite Palette von Berechtigungen für die AWS Produkte, die Elastic Beanstalk zur Verwaltung von Anwendungen und Umgebungen verwendet. `ec2:*`Ermöglicht es einem IAM-Benutzer beispielsweise, jede Aktion auf einer beliebigen Amazon EC2 EC2-Ressource im AWS Konto auszuführen. Diese Berechtigungen sind nicht auf Ressourcen beschränkt, die mit Elastic Beanstalk verwendet werden. Als bewährte Methode sollten Sie Personen nur die Berechtigungen erteilen, die sie für ihre Aufgaben benötigen.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ec2:*",
            "elasticloadbalancing:*",
            "autoscaling:*",
            "cloudwatch:*",
            "s3:*",
            "sns:*",
            "rds:*",
            "cloudformation:*"
         ],
         "Resource":"*"
      }
   ]
}
```

In der zweiten Richtlinie werden die Elastic Beanstalk-Aktionen definiert, die Jack für die "app1"-Ressource ausführen kann.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllReadCallsAndAllVersionCallsInApplications",
         "Action":[
            "elasticbeanstalk:Describe*",
            "elasticbeanstalk:RequestEnvironmentInfo",
            "elasticbeanstalk:RetrieveEnvironmentInfo",
            "elasticbeanstalk:CreateApplicationVersion",
            "elasticbeanstalk:DeleteApplicationVersion",
            "elasticbeanstalk:UpdateApplicationVersion"
         ],
```

```
         "Effect":"Allow",
         "Resource":[
            "*"
         ],
         "Condition":{
            "StringEquals":{
               "elasticbeanstalk:InApplication":[
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app1"
               ]
            }
         }
      },
      {
         "Sid":"AllReadCallsOnApplications",
         "Action":[
            "elasticbeanstalk:DescribeApplications",
            "elasticbeanstalk:DescribeEvents"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1"
         ]
      },
      {
         "Sid":"UpdateEnvironmentInApplications",
         "Action":[
            "elasticbeanstalk:UpdateEnvironment"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/app1/
app1-staging*"
         ],
         "Condition":{
            "StringEquals":{
               "elasticbeanstalk:InApplication":[
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app1"
               ]
            },
            "ArnLike":{
               "elasticbeanstalk:FromApplicationVersion":[
```

```
                "arn:aws:elasticbeanstalk:us-
east-2:123456789012:applicationversion/app1/*"
             ]
          }
       }
    },
    {
       "Sid":"AllReadCallsOnSolutionStacks",
       "Action":[
          "elasticbeanstalk:ListAvailableSolutionStacks",
          "elasticbeanstalk:DescribeConfigurationOptions",
          "elasticbeanstalk:ValidateConfigurationSettings"
       ],
       "Effect":"Allow",
       "Resource":[
          "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
       ]
    }
 ]
}
```

Die dritte Richtlinie legt die Elastic Beanstalk-Aktionen fest, für die die zweite Richtlinie Berechtigungen benötigt, um diese Elastic Beanstalk-Aktionen abzuschließen. Mit der Anweisung `AllNonResourceCalls` kann die `elasticbeanstalk:CheckDNSAvailability`-Aktion ausgeführt werden, die für den Aufruf von `elasticbeanstalk:CreateEnvironment` und anderen Aktionen erforderlich ist. Zudem kann damit die `elasticbeanstalk:CreateStorageLocation`-Aktion ausgeführt werden, die für `elasticbeanstalk:CreateEnvironment` und weitere Aktionen benötigt wird.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllNonResourceCalls",
         "Action":[
            "elasticbeanstalk:CheckDNSAvailability",
            "elasticbeanstalk:CreateStorageLocation"
         ],
```

```
            "Effect":"Allow",
            "Resource":[
               "*"
            ]
         }
      ]
}
```

Verhinderung des umgebungsübergreifenden Amazon S3 S3-Bucket-Zugriffs

In diesem Thema wird erklärt, wie verwaltete Richtlinien den umgebungsübergreifenden Zugriff auf S3-Buckets ermöglichen können und wie Sie benutzerdefinierte Richtlinien erstellen können, um diese Art von Zugriff zu verwalten.

Elastic Beanstalk bietet verwaltete Richtlinien für die Verwaltung der AWS Ressourcen, die von den Elastic Beanstalk Beanstalk-Umgebungen in Ihrem Konto benötigt werden. AWS Die standardmäßig für eine Anwendung in Ihrem AWS Konto bereitgestellten Berechtigungen haben Zugriff auf S3-Ressourcen, die zu anderen Anwendungen in demselben Konto gehören. AWS

Wenn auf Ihrem AWS Konto mehrere Beanstalk-Anwendungen ausgeführt werden, können Sie die Sicherheit Ihrer Richtlinien einschränken, indem Sie Ihre eigene benutzerdefinierte Richtlinie erstellen, die Sie an Ihre eigene Servicerolle oder Ihr Instanzprofil für jede Umgebung anhängen. Anschließend können Sie die S3-Berechtigungen in Ihrer benutzerdefinierten Richtlinie auf eine bestimmte Umgebung beschränken.

> **Note**
>
> Beachten Sie, dass Sie für die Aufrechterhaltung Ihrer benutzerdefinierten Richtlinie verantwortlich sind. Wenn sich eine von Elastic Beanstalk verwaltete Richtlinie, auf der Ihre benutzerdefinierte Richtlinie basiert, ändert, müssen Sie Ihre benutzerdefinierte Richtlinie mit den entsprechenden Änderungen an der Basisrichtlinie ändern. Eine Änderungshistorie der von Elastic Beanstalk verwalteten Richtlinien finden Sie unter. Elastic Beanstalk Beanstalk-Updates für verwaltete Richtlinien AWS

Beispiel für Berechtigungen mit eingeschränktem Geltungsbereich

Das folgende Beispiel basiert auf der AWSElasticBeanstalkWebTierverwalteten Richtlinie.

Die Standardrichtlinie umfasst die folgenden Zeilen für Berechtigungen für S3-Buckets. Diese Standardrichtlinie beschränkt die S3-Bucket-Aktionen nicht auf bestimmte Umgebungen oder Anwendungen.

```
{
   "Sid" : "BucketAccess",
   "Action" : [
      "s3:Get*",
      "s3:List*",
      "s3:PutObject"
     ],
   "Effect" : "Allow",
   "Resource" : [
      "arn:aws:s3:::elasticbeanstalk-*",
      "arn:aws:s3:::elasticbeanstalk-*/*"
     ]
}
```

Sie können den Zugriff einschränken, indem Sie bestimmte Ressourcen für eine als angegebene Servicerolle qualifizieren`Principal`. Das folgende Beispiel stellt die benutzerdefinierten `aws-elasticbeanstalk-ec2-role-my-example-env` Servicerollenberechtigungen für S3-Buckets in der Umgebung mit der ID `my-example-env-ID` bereit.

Example Gewähren Sie Berechtigungen nur für die S3-Buckets einer bestimmten Umgebung

```
{
   "Sid": "BucketAccess",
   "Action": [
      "s3:Get*",
      "s3:List*",
      "s3:PutObject"
    ],
   "Effect": "Allow",
   "Principal": {
      "AWS": "arn:aws:iam::...:role/aws-elasticbeanstalk-ec2-role-my-example-env"
     },
   "Resource": [
      "arn:aws:s3:::elasticbeanstalk-my-region-account-id-12345",
```

```
      "arn:aws:s3:::elasticbeanstalk-my-region-account-id-12345/resources/environments/
my-example-env-ID/*"
    ]
}
```

> **Note**
>
> Der Ressourcen-ARN muss die Elastic Beanstalk Beanstalk-Umgebungs-ID enthalten (nicht den Umgebungsnamen). Sie können die Umgebungs-ID von der Elastic Beanstalk Beanstalk-Konsole auf der Seite mit der Umgebungsübersicht abrufen. Sie können diese Informationen auch mit dem Befehl AWS CLI describe-environments abrufen.

Weitere Informationen zur Aktualisierung der S3-Bucket-Berechtigungen für Ihre Elastic Beanstalk Beanstalk-Umgebungen finden Sie in den folgenden Ressourcen:

- Verwenden von Elastic Beanstalk mit Amazon S3 in dieser Anleitung
- Von Amazon S3 im Service Authorization Reference Guide definierte Ressourcentypen
- ARN-Format im IAM-Benutzerhandbuch

Verwenden von Elastic Beanstalk mit Amazon RDS

In diesem Abschnitt wird erklärt, wie Sie Elastic Beanstalk mit Amazon Relational Database Service (Amazon RDS) verwenden können, um eine relationale Datenbank einzurichten, zu betreiben und zu skalieren. Wir erklären einige Konzepte zur Konfiguration und geben Empfehlungen. Anschließend führen wir Sie durch den Prozess zum Erstellen und Herstellen einer Verbindung zu einem Amazon RDS.

Für den Einstieg gibt es zwei Möglichkeiten:

- Erstellen Sie eine neue Datenbank in Amazon RDS.
- Beginnen Sie mit einer Datenbank, die zuvor von Elastic Beanstalk erstellt und anschließend von einer Beanstalk-Umgebung entkoppelt wurde. Weitere Informationen finden Sie unter the section called “Datenbank”.

Wählen Sie den Ansatz

Sie können beide Ansätze verwenden, um eine Datenbank-Instance in Amazon RDS auszuführen und Ihre Anwendung so zu konfigurieren, dass sie beim Start eine Verbindung zu ihr herstellt. Sie können mehrere Umgebungen mit einer Datenbank verbinden.

> **Note**
>
> Wenn Sie noch keine Datenbank-Instance mit Ihrer Anwendung verwendet haben, empfehlen wir Ihnen, einer Testumgebung zuerst eine Datenbank mit der Elastic-Beanstalk-Konsole hinzuzufügen. Auf diese Weise können Sie überprüfen, ob Ihre Anwendung die Umgebungseigenschaften lesen, eine Verbindungszeichenfolge erstellen und eine Verbindung zu einer Datenbank-Instance herstellen kann, ohne dass zusätzliche Konfigurationsarbeiten für eine eigenständige Datenbank erforderlich sind. Weitere Informationen finden Sie unter Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung.

Konfigurieren einer Sicherheitsgruppe

Damit die EC2 Amazon-Instances in Ihrer Umgebung eine Verbindung zu einer externen Datenbank herstellen können, konfigurieren Sie eine zusätzliche Sicherheitsgruppe für die Auto Scaling Scaling-Gruppe, die Ihrer Umgebung zugeordnet ist. Sie können dieselbe Sicherheitsgruppe anhängen, die an Ihre Datenbankinstance angehängt ist. Sie können aber auch eine eigene Sicherheitsgruppe verwenden. Wenn Sie eine andere Sicherheitsgruppe anhängen, müssen Sie die Sicherheitsgruppe konfigurieren, die an Ihre Datenbank angefügt ist, um eingehenden Zugriff von dieser Sicherheitsgruppe aus zu ermöglichen.

> **Note**
>
> Sie können Ihre Umgebung mit einer Datenbank verbinden, indem Sie eine Regel zur Sicherheitsgruppe hinzufügen, die mit der Datenbank verknüpft ist. Diese Regel muss eingehenden Zugriff von der automatisch generierten Sicherheitsgruppe ermöglichen, die Elastic Beanstalk an die Auto-Scaling-Gruppe für Ihre Umgebung anhängt. Wenn Sie jedoch wissen, dass Sie durch das Erstellen dieser Regel auch eine Abhängigkeit zwischen den beiden Sicherheitsgruppen erstellen. Wenn Sie anschließend versuchen, die Umgebung zu beenden, kann Elastic Beanstalk die Sicherheitsgruppe der Umgebung nicht beenden, da die Sicherheitsgruppe der Datenbank davon abhängig ist.

Konfigurieren Sie die Datenbankverbindung

Nach dem Starten Ihrer Datenbank-Instance und der Konfiguration der Sicherheitsgruppen können Sie die Verbindungsinformationen (Endpunkt, Passwort usw.) an die Anwendung übergeben, indem Sie Umgebungseigenschaften verwenden. Hierbei handelt es sich um denselben Mechanismus, den Elastic Beanstalk im Hintergrund verwendet, wenn Sie eine Datenbank-Instance in Ihrer Umgebung ausführen.

Für eine zusätzliche Sicherheitebene können Sie Ihre Verbindungsinformationen in Amazon S3 speichern und Elastic Beanstalk so konfigurieren, dass sie während der Bereitstellung abgerufen werden. Mit Konfigurationsdateien (`.ebextensions`) können Sie die Instances in Ihrer Umgebung so konfigurieren, dass sie sicher Dateien aus Amazon S3 abrufen, wenn Sie Ihre Anwendung bereitstellen.

Themen

- Starten von und Verbinden mit einer externen Amazon RDS-Instance in einer Standard-VPC
- Speichern der Amazon RDS-Anmeldeinformationen in AWS Secrets Manager
- Bereinigen einer externen Amazon RDS-Instance

Starten von und Verbinden mit einer externen Amazon RDS-Instance in einer Standard-VPC

Die folgenden Verfahren beschreiben den Prozess für die Verbindung mit einer externen Amazon RDS-Instance zu einer Standard-VPC. Der Prozess ist identisch, wenn Sie eine benutzerdefinierte VPC verwenden. Die einzigen zusätzlichen Voraussetzungen sind, dass sich Ihre Umgebung und DB-Instance im selben Subnetz oder in Subnetzen befindet, die miteinander kommunizieren dürfen. Weitere Informationen zum Konfigurieren einer benutzerdefinierten VPC zur Verwendung mit Elastic Beanstalk finden Sie unter Verwenden von Elastic Beanstalk mit Amazon VPC.

> **Note**
>
> - Eine Alternative zum Starten einer neuen DB-Instance besteht darin, mit einer Datenbank zu beginnen, die zuvor von Elastic Beanstalk erstellt und anschließend von einer Beanstalk-Umgebung entkoppelt wurde. Weitere Informationen finden Sie unter the section called “Datenbank”. Mit dieser Option müssen Sie das Verfahren zum Start eine neue Datenbank nicht ausführen. Sie müssen jedoch die nachfolgenden Verfahren ausführen, die in diesem Thema beschrieben sind.

- Wenn Sie mit einer Datenbank beginnen, die von Elastic Beanstalk erstellt und anschließend von einer Beanstalk-Umgebung entkoppelt wurde, können Sie die erste Gruppe von Schritten überspringen und mit den Schritten fortfahren, die unter So ändern Sie die Eingangsregeln für die Sicherheitsgruppe Ihrer RDS-Instance gruppiert sind.
- Wenn Sie vorhaben, die Datenbank zu verwenden, die Sie für eine Produktionsumgebung entkoppeln, überprüfen Sie, ob der Speichertyp, den die Datenbank verwendet, für Ihren Workload geeignet ist. Weitere Informationen finden Sie unter DB-Instance-Speicher und Ändern einer DB-Instance im Amazon-RDS-Benutzerhandbuch.

So starten Sie eine RDS DB-Instance in einer Standard-VPC

1. Öffnen Sie die RDS-Konsole.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.
4. Wählen Sie Standard Create (Standarderstellung).

> **Important**
>
> Wählen Sie nicht Easy Create (Einfache Erstellung) aus. Wenn Sie diese Option auswählen, können Sie die erforderlichen Einstellungen zum Starten dieser RDS-Datenbank nicht konfigurieren.

5. Geben Sie unter Additional configuration (Zusätzliche Konfiguration) für Initial database name (Anfangsdatenbankname) „**ebdb**“ ein.
6. Überprüfen Sie die Standardeinstellungen und passen Sie diese Einstellungen an Ihre spezifischen Anforderungen an. Achten Sie auf die folgenden Optionen:
   - DB instance class (DB-Instance-Klasse) – Wählen Sie eine Instance-Größe, die eine geeignete Speichermenge und CPU-Leistung für den Workload bereitstellt.
   - Multi-AZ-Bereitstellung — Für hohe Verfügbarkeit stellen Sie diese Option auf Einen Replica/ Reader Aurora-Knoten in einer anderen AZ erstellen ein.
   - Master username (Master-Benutzername) und Master password (Master-Passwort) – Der Benutzername und das Passwort für die Datenbank. Notieren Sie sich diese Einstellungen, um sie später zu verwenden.

7. Überprüfen Sie die Standardeinstellungen für die übrigen Optionen und wählen Sie dann Create database (Datenbank starten) aus.

Als Nächstes bearbeiten Sie die Sicherheitsgruppe für die DB-Instance, damit eingehender Datenverkehr über den entsprechenden Port zulässig ist. Es handelt sich um die gleiche Sicherheitsgruppe, die später mit der Elastic-Beanstalk-Umgebung verknüpft wird. Infolgedessen gewährt die von Ihnen hinzugefügte Regel eingehenden Zugriff auf andere Ressourcen in derselben Sicherheitsgruppe.

So ändern Sie die Eingangsregeln für die Sicherheitsgruppe, die an Ihre RDS-Instance angehängt ist

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance, damit die Details der Instance angezeigt werden.
4. Notieren Sie im Abschnitt Konnektivität die auf dieser Seite angezeigten Subnetze, Sicherheitsgruppen und Endpunkte. Auf diese Weise können Sie diese Informationen später verwenden.
5. In Security (Sicherheit) wird die mit der DB-Instance verknüpfte Sicherheitsgruppe angezeigt. Öffnen Sie den Link, um die Sicherheitsgruppe in der EC2 Amazon-Konsole anzuzeigen.
6. Klicken Sie in den Details der Sicherheitsgruppe auf Inbound (Eingehend).
7. Wählen Sie Edit aus.
8. Klicken Sie auf Add Rule (Regel hinzufügen).
9. Wählen Sie unter Type (Typ) die von der Anwendung verwendete DB-Engine aus.
10. Geben Sie für Source (Quelle) **`sg-`** ein, um eine Liste der verfügbaren Sicherheitsgruppen anzuzeigen. Wählen Sie die Sicherheitsgruppe aus, die mit der Auto-Scaling-Gruppe verknüpft ist, die mit Ihrer Elastic-Beanstalk-Umgebung verwendet wird. Auf diese Weise können EC2 Amazon-Instances in der Umgebung auf die Datenbank zugreifen.



11. Wählen Sie Speichern.

Fügen Sie als Nächstes die Sicherheitsgruppe für die DB-Instance zu Ihrer laufenden Umgebung hinzu. In diesem Verfahren stellt Elastic Beanstalk alle Instances in Ihrer Umgebung mit der angehängten zusätzlichen Sicherheitsgruppe erneut bereit.

So fügen Sie eine Sicherheitsgruppe zur Umgebung hinzu

- Gehen Sie folgendermaßen vor:
  - So fügen Sie eine Sicherheitsgruppe mit der Elastic Beanstalk-Konsole hinzu
    a. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
    b. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
    c. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
    d. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).
    e. Wählen Sie unter EC2 Sicherheitsgruppen die Sicherheitsgruppe aus, die an die Instances angehängt werden soll, zusätzlich zu der Instance-Sicherheitsgruppe, die Elastic Beanstalk erstellt.
    f. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.
    g. Lesen Sie die Warnmeldung und klicken Sie dann auf Bestätigen.
  - Fügen Sie eine Sicherheitsgruppe mit einer Konfigurationsdatei hinzufügen. Verwenden Sie die `securitygroup-addexisting.config`-Beispieldatei.

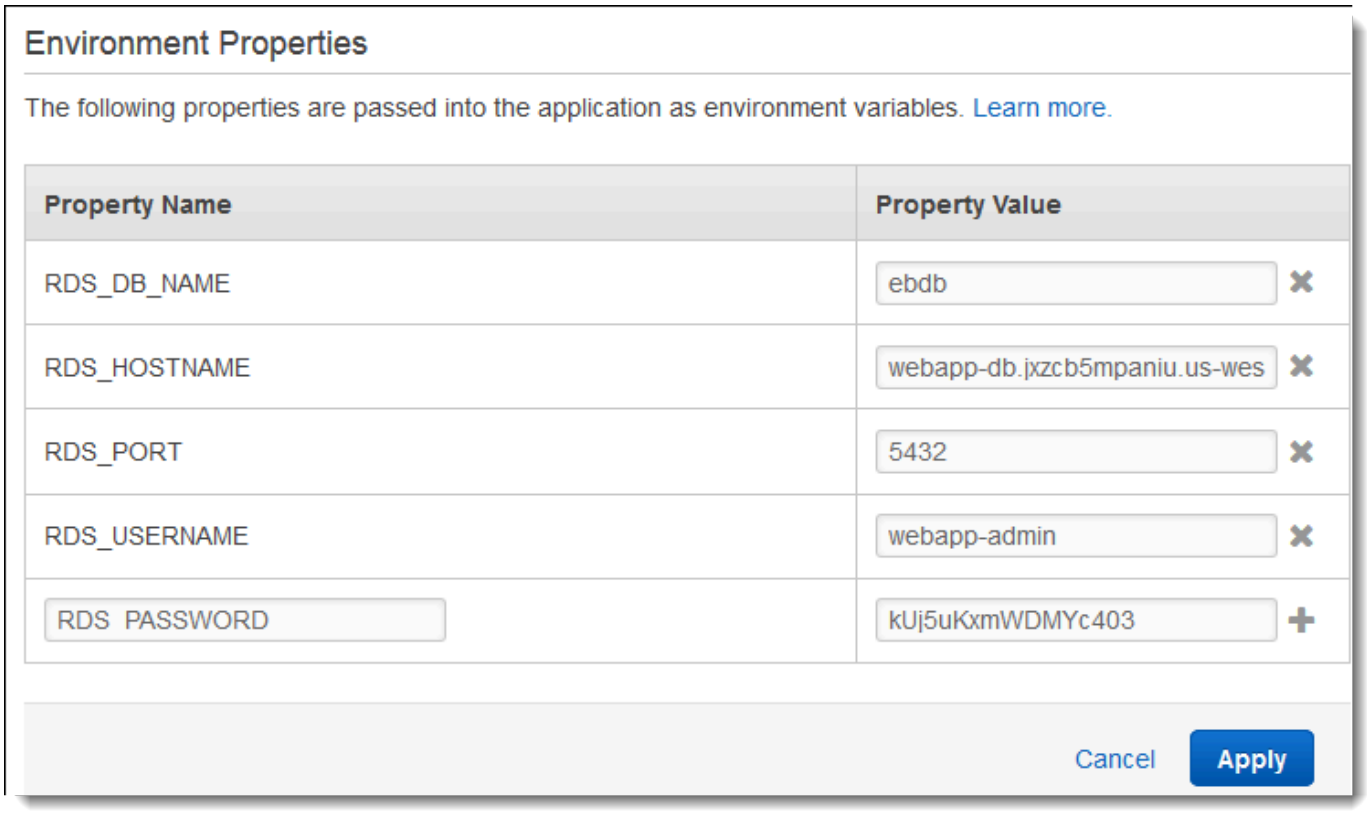
Anschließend werden die Verbindungsinformationen mithilfe von Umgebungseigenschaften an die Umgebung übergeben. Wenn Sie mit der Elastic-Beanstalk-Konsole der Umgebung eine DB-Instance hinzufügen, verwendet Elastic Beanstalk Umgebungseigenschaften, wie RDS_HOSTNAME, um Verbindungsinformationen an die Anwendung zu übergeben. Sie können die gleichen Eigenschaften verwenden. Auf diese Weise verwenden Sie denselben Anwendungscode mit integrierten DB-Instances und externen DB-Instances. Alternativ können Sie auch Ihre eigenen Eigenschaftsnamen wählen.

So konfigurieren Sie Umgebungseigenschaften für eine DB-Instance von Amazon RDS

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.
5. Definieren Sie im Abschnitt Environment properties (Umgebungseigenschaften) die Variablen, die Ihre Anwendung liest, um eine Verbindungszeichenfolge zu bilden. Zur Sicherstellung der Kompatibilität mit Umgebungen, die eine integrierte RDS-DB-Instance haben, verwenden Sie die folgenden Namen und Werte. Sie finden alle Werte außer dem Passwort in der RDS-Konsole.

Name der Eigenschaft	Beschreibung	Eigenschaftenwert
`RDS_HOSTNAME`	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
`RDS_PORT`	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.

Name der Eigenschaft	Beschreibung	Eigenschaftenwert
RDS_DB_NAME	Der Datenbanknam, **ebdb**.	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfigur ation) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	Es ist in der Amazon RDS-Konsole nicht als Referenz angegeben.



6. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Wenn Sie Ihre Anwendung noch nicht für das Programmieren von Umgebungseigenschaften und Erstellen einer Verbindungszeichenfolge programmiert haben, finden Sie Anweisungen in den folgenden sprachspezifischen Themen:

- Java SE – Verbinden mit einer Datenbank (Java SE-Plattformen)
- Java mit Tomcat – Verbinden mit einer Datenbank (Tomcat-Plattformen)
- Node.js – Herstellen einer Verbindung zu einer Datenbank
- .NET – Verbinden mit einer Datenbank
- PHP – Mit einem PDO oder My verbinden Sie sich mit einer Datenbank SQLi
- Python – Herstellen einer Verbindung zu einer Datenbank
- Ruby – Herstellen einer Verbindung zu einer Datenbank

Je nachdem, wann Ihre Anwendung Umgebungsvariablen liest, müssen Sie den Anwendungsserver auf den Instances in Ihrer Umgebung möglicherweise neu starten.

So starten Sie die App-Server der Umgebung neu

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Klicken Sie auf Actions (Aktionen) und wählen Sie dann Restart App Server(s) (App-Server neu starten) aus.

Speichern der Amazon RDS-Anmeldeinformationen in AWS Secrets Manager

In diesem Thema wird erklärt, wie AWS Secrets Manager Sie Ihre Sicherheitslage beim Abrufen von Anmeldeinformationen mit Elastic Beanstalk verbessern können. Es enthält auch Verweise auf bestimmte Ressourcen, die Ihnen bei der Konfiguration der Anmeldeinformationen für Ihre Elastic Beanstalk Beanstalk-Anwendung helfen können.

AWS Secrets Manager hilft Ihnen dabei, Ihre Sicherheitslage zu verbessern, indem es die Möglichkeit bietet, verschlüsselte Anmeldedaten zu speichern und abzurufen. Das Speichern der Anmeldeinformationen im Secrets Manager trägt dazu bei, eine mögliche Kompromittierung durch

jemanden zu verhindern, der Ihre Anwendung oder die Komponenten inspizieren kann. Ihr Code kann einen Runtime-Aufruf an den Secrets Manager-Dienst senden, um Anmeldeinformationen dynamisch abzurufen. Secrets Manager bietet auch Funktionen wie clientseitige geheime Caching-Komponenten für Laufzeitsprachen wie Python, Go und Java.

Weitere Informationen finden Sie in folgenden Themen im AWS Secrets Manager - Benutzerhandbuch:

- So verwendet Amazon RDS
- Erstellen Sie ein AWS Secrets Manager Datenbankgeheimnis
- Rufen Sie Geheimnisse ab von AWS Secrets Manager

Bereinigen einer externen Amazon RDS-Instance

Wenn Sie eine externe Amazon-RDS-Instance mit Ihrer Elastic-Beanstalk-Umgebung verbinden, ist die Datenbank-Instance nicht vom Lebenszyklus Ihrer Umgebung abhängig und wird daher nicht gelöscht, wenn Sie Ihre Umgebung beenden. Um sicherzustellen, dass personenbezogene Informationen, die Sie möglicherweise in der Datenbankinstance gespeichert haben, nicht unnötig beibehalten werden, löschen Sie nicht mehr benötigte Datensätze. Löschen Sie alternativ die Datenbank-Instance.

Verwenden von Elastic Beanstalk mit Amazon S3

In diesem Thema wird erklärt, wie Elastic Beanstalk Amazon Simple Storage Service (Amazon S3) nutzt und welche Objekttypen es in S3-Buckets speichert. Außerdem wird darin vermerkt, welche Objekte Sie manuell löschen müssen, nachdem Sie Ihre Elastic Beanstalk Beanstalk-Umgebung beendet haben, und es werden Anweisungen dazu bereitgestellt.

Der Elastic Beanstalk Amazon S3 S3-Kundenkonto-Bucket

Elastic Beanstalk erstellt einen verschlüsselten Amazon S3 S3-Bucket, der `elasticbeanstalk-`*`region`*`-`*`account-id`* nach jeder Region benannt ist, in der Sie Umgebungen erstellen. Ihr AWS Konto besitzt diesen Bucket. Elastic Beanstalk speichert temporäre Konfigurationsdateien und andere Objekte für den ordnungsgemäßen Betrieb Ihrer Anwendung in diesem Bucket. Elastic Beanstalk erfordert aktivierte Buckets ACLs für service-verwaltete Buckets und aktiviert daher die Access Control List (ACL) dieses Buckets.

Beachten Sie, dass Amazon S3 Bucket Access Control Lists (ACLs) standardmäßig deaktiviert. Darüber hinaus wird im Thema ACL-Übersicht im Amazon S3 S3-Benutzerhandbuch empfohlen, die ACLs Option deaktiviert zu lassen, mit Ausnahme bestimmter Anwendungsfälle. Die vom Service verwalteten Elastic Beanstalk Beanstalk-Buckets fallen in einen Anwendungsfall, der aktiviert werden muss. ACLs Um die Sicherheit zu gewährleisten, erzwingen Elastic Beanstalk Beanstalk-Bereitstellungen, dass dieser Bucket dem Konto gehört, auf dem die Anwendung ausgeführt wird.

Elastic Beanstalk behält die Standardverschlüsselung bei, die von Amazon S3 S3-Buckets bereitgestellt wird. Weitere Informationen zur Bucket-Verschlüsselung finden Sie unter Amazon S3 S3-Standardverschlüsselung im Amazon Simple Storage Service-Benutzerhandbuch.

Inhalt des Elastic Beanstalk Amazon S3 S3-Kundenkonto-Buckets

In der folgenden Tabelle sind einige Objekte aufgeführt, die Elastic Beanstalk in Ihrem Kundenkonto-Bucket speichert. Die Tabelle zeigt auch, welche Objekte manuell gelöscht werden müssen. Um unnötige Speicherkosten zu vermeiden und sicherzustellen, dass personenbezogene Daten nicht gespeichert werden, sollten Sie diese Objekte manuell löschen, wenn Sie sie nicht mehr benötigen.

Objekt	Wann wird gespeichert?	Wann gelöscht?
Anwendungsversionen	Wenn Sie eine Umgebung erstellen oder Ihren Anwendungscode in eine bestehende Umgebung implementieren, speichert Elastic Beanstalk eine Anwendungsversion in Amazon S3 und ordnet sie der Umgebung zu.	Beim Löschen der Anwendung und gemäß Lebenszyklus der Version.
Quell-Bundles	Wenn Sie eine neue Anwendungsversion über die Elastic Beanstalk-Konsole oder die EB CLI hochladen, speichert Elastic Beanstalk eine Kopie davon in Amazon S3 und legt sie als Quell-Bundle Ihrer Umgebung fest.	Manuell. Wenn Sie eine Anwendungsversion löschen, können Sie Delete versions from Amazon S3 (Versionen von Amazon S3 löschen) auswählen, um auch das zugehörige Quell-Bundle zu löschen. Details hierzu finden Sie unter Verwalten von Anwendungsversionen.

Objekt	Wann wird gespeichert?	Wann gelöscht?
Custom platforms	Wenn Sie eine benutzerdefinierte Plattform erstellen, speichert Elastic Beanstalk die zugehörigen Daten vorübergehend in Amazon S3.	Nach erfolgreichem Abschluss der Erstellung der benutzerdefinierten Plattform.
Protokolldateien	Sie können Elastic Beanstalk auffordern, Instance-Protokolldateien (Protokollfragmente oder Bundle-Protokolle) abzurufen und in Amazon S3 zu speichern. Sie können auch die Protokollrotation aktivieren und Ihre Umgebung so konfigurieren, dass Protokolle nach der Rotation automatisch in Amazon S3 veröffentlicht werden.	Protokollfragmente und Bundle-Protokolle: 15 Minuten nach ihrer Erstellung. Rotierte Protokolle: Manuell.
Gespeicherte Konfigurationen	Manuell.	Manuell.

Löschen von Objekten im Elastic Beanstalk Amazon S3-Bucket

Wenn Sie eine Umgebung beenden oder eine Anwendung löschen, löscht Elastic Beanstalk die meisten zugehörigen Objekte aus Amazon S3. Um die Speicherkosten einer laufenden Anwendung zu minimieren, löschen Sie regelmäßig Objekte, die Ihre Anwendung nicht benötigt. Achten Sie außerdem auf Objekte, die Sie manuell löschen müssen, wie unter Inhalt des Elastic Beanstalk Amazon S3 S3-Kundenkonto-Buckets aufgelistet. Um sicherzustellen, dass private Informationen nicht unnötig beibehalten werden, löschen Sie diese Objekte, wenn Sie sie nicht mehr benötigen.

- Löschen Sie Anwendungsversionen, von denen Sie nicht mehr erwarten, dass sie in Ihrer Anwendung verwendet werden. Wenn Sie eine Anwendungsversion löschen, können Sie Delete versions from Amazon S3 (Versionen aus Amazon S3 löschen) auswählen, um auch das zugehörige Quellcode-Bundle zu löschen – eine Kopie des Quellcodes und der Konfigurationsdateien Ihrer Anwendung, die Elastic Beanstalk nach Amazon S3 hochgeladen hat, wenn Sie eine Anwendung bereitgestellt oder eine Anwendungsversion hochgeladen haben.

Weitere Informationen zum Löschen einer Anwendungsversion finden Sie unter Verwalten von Anwendungsversionen.

- Löschen Sie rotierte Protokolle, die Sie nicht benötigen. Alternativ können Sie sie herunterladen oder zur weiteren Analyse nach Amazon Glacier verschieben.
- Löschen Sie gespeicherte Konfigurationen, die Sie in keiner Umgebung mehr verwenden werden.

Löschen des Elastic Beanstalk Amazon S3-Buckets

Wenn Elastic Beanstalk einen Bucket erstellt, wird auch eine Richtlinie für den Bucket erstellt, die auf den neuen Bucket angewendet wird. Diese Richtlinie dient zwei Zwecken:

- Um Umgebungen das Schreiben in den Bucket zu ermöglichen.
- Um ein versehentliches Löschen des Buckets zu verhindern.

Aufgrund der Richtlinie, die Elastic Beanstalk auf die Buckets anwendet, welche es für Ihre Umgebungen erstellt, ist es Ihnen nicht erlaubt, diese Buckets zu löschen, es sei denn, Sie löschen die Bucket-Richtlinie vorher bewusst. Sie können die Richtlinie für den Bucket im Abschnitt Berechtigungen der Bucket-Eigenschaften in der Amazon-S3-Konsole löschen.

> **Warning**
>
> Wir empfehlen Ihnen, bestimmte nicht benötigte Objekte aus Ihrem Elastic Beanstalk Amazon S3 S3-Bucket zu löschen, anstatt den gesamten Bucket zu löschen.
> Wenn Sie einen Bucket löschen, den Elastic Beanstalk in Ihrem Konto erstellt hat, und Sie noch vorhandene Anwendungen und laufende Umgebungen in der entsprechenden Region haben, funktionieren Ihre Anwendungen möglicherweise nicht mehr richtig. Beispiel:
>
> - Wenn eine Umgebung skaliert wird, sollte Elastic Beanstalk in der Lage sein, die Anwendungsversion der Umgebung im Amazon S3 S3-Bucket zu finden und sie zum Starten neuer EC2 Amazon-Instances zu verwenden.
> - Wenn Sie eine benutzerdefinierte Plattform erstellen, verwendet Elastic Beanstalk während des Erstellungsprozesses eine temporäre Amazon S3-Speicherung.

Weitere Informationen zu den Auswirkungen des Löschens eines S3-Buckets finden Sie in den Überlegungen, die unter Löschen eines Buckets im Amazon S3 S3-Benutzerhandbuch aufgeführt sind.

So löschen Sie einen Elastic Beanstalk-Speicher-Bucket (Konsole)

Das allgemeine Verfahren zum Löschen eines S3-Buckets wird auch unter Löschen eines Buckets im Amazon S3 S3-Benutzerhandbuch beschrieben. Da wir im folgenden Verfahren einen von Elastic Beanstalk erstellten Bucket löschen, fügen wir zusätzliche Schritte hinzu, um zuerst die Bucket-Richtlinie zu löschen.

1. Öffnen Sie die Amazon S3-Konsole.
2. Öffnen Sie die Seite des Elastic Beanstalk-Speicher-Buckets, indem Sie den Bucket-Namen auswählen.
3. Wählen Sie die Registerkarte Berechtigungen.
4. Wählen Sie Bucket Policy aus.
5. Wählen Sie Löschen aus.
6. Gehen Sie zurück zur Hauptseite der Amazon-S3-Konsole und wählen Sie dann den Speicher-Bucket von Elastic Beanstalk aus.
7. Wählen Sie Delete Bucket (Bucket löschen).
8. Bestätigen Sie, dass Sie den Bucket löschen möchten, indem Sie den Bucket-Namen in das Textfeld eingeben, und wählen Sie dann Bucket löschen.

Elastic Beanstalk mit AWS Secrets Manager und AWS Systems Manager Parameter Store verwenden

In diesem Thema wird erklärt, wie Sie AWS Systems Manager Parameter Store in Ihrer Elastic Beanstalk Beanstalk-Umgebung verwenden AWS Secrets Manager können, um vertrauliche Informationen wie Anmeldeinformationen und API-Schlüssel sicher zu speichern und abzurufen. Ihre Anwendung kann gespeicherte Geheimnisse und Parameter direkt aus diesen Speichern abrufen, indem Sie die APIs oder die Befehlszeilentools dieser Dienste verwenden.

Elastic Beanstalk bietet auch die Möglichkeit, Secrets Manager- und Systems Manager Parameter Store-Daten in Umgebungsvariablen zu referenzieren. Dies ist eine sichere Option für Ihre

Anwendung, um nativ auf Secrets und Parameter zuzugreifen, die von diesen Services gespeichert werden, ohne API-Aufrufe an sie verwalten zu müssen.

Themen

- Abrufen von Geheimnissen und Parametern für Elastic Beanstalk Beanstalk-Umgebungsvariablen
- Erforderliche IAM-Berechtigungen für Elastic Beanstalk für den Zugriff auf Geheimnisse und Parameter
- Verwendung AWS Secrets Manager und Speicherung von AWS Systems Manager Parametern
- Fehlerbehebung bei der Integration von Geheimnissen mit Elastic Beanstalk Beanstalk-Umgebungsvariablen

Abrufen von Geheimnissen und Parametern für Elastic Beanstalk Beanstalk-Umgebungsvariablen

Elastic Beanstalk kann beim Instance-Bootstrapping Werte aus dem AWS Secrets Manager AWS Systems Manager Parameterspeicher abrufen und sie Umgebungsvariablen zuweisen, die Ihre Anwendung verwenden kann.

In den folgenden Punkten werden Konfiguration, Synchronisation und Zugriff für die Verwendung von Umgebungsvariablen als Geheimnisse zusammengefasst:

- Konfigurieren Sie Ihre Umgebungsvariablen zum Speichern von Geheimnissen, indem Sie die Amazon-Ressourcennamen (ARNs) für die Geheimnisse und Parameter angeben, die sie speichern werden.
- Wenn geheime Werte in Secrets Manager oder Systems Manager Parameter Store aktualisiert oder rotiert werden, müssen Sie Ihre Umgebungsvariablen manuell aktualisieren.
- Die geheimen Umgebungsvariablen sind für Bextension-Container-Befehle und Plattform-Hooks verfügbar.

Unterstützte Plattformversionen

Plattformversionen, die am oder nach dem 26. März 2025 veröffentlicht wurden, unterstützen AWS Secrets Manager Geheimnisse und AWS Systems Manager Parameter Store-Parameter, die als Umgebungsvariablen konfiguriert sind.

> **Note**
>
> Mit Ausnahme der Docker- und ECS-basierten Docker-Plattformen unterstützen die Amazon Linux 2-Plattformversionen keine mehrzeiligen Variablenwerte. Weitere Informationen zur Unterstützung mehrzeiliger Variablen finden Sie unter. Mehrzeilige Werte

Themen

- Preisgestaltung
- Secrets als Elastic Beanstalk Beanstalk-Umgebungsvariablen konfigurieren
- Extrahieren von JSON-Schlüsseln aus Secrets Manager Manager-Geheimnissen
- Bewährte Methoden für die Synchronisation von Geheimnissen mit Elastic Beanstalk Beanstalk-Umgebungsvariablen
- Mehrzeilige Werte in Amazon Linux 2-Umgebungsvariablen

Preisgestaltung

Für die Nutzung von Secrets Manager und Systems Manager Parameter Store fallen Standardgebühren an. Weitere Informationen zur Preisgestaltung finden Sie auf den folgenden Websites:

- AWS Secrets Manager Preisgestaltung
- AWS Systems Manager Preisgestaltung (wählen Sie Parameter Store aus der Inhaltsliste aus)

Elastic Beanstalk erhebt keine Gebühren dafür, dass Ihre Anwendung über Umgebungsvariablen auf Umgebungsgeheimnisse verweist. Für Anfragen, die Elastic Beanstalk in Ihrem Namen an diese Services stellt, fallen jedoch Standardgebühren an.

Secrets als Elastic Beanstalk Beanstalk-Umgebungsvariablen konfigurieren

Sie können die Elastic Beanstalk Beanstalk-Konsole, die Konfigurationsdateien in `.ebextensions`, die AWS CLI und das AWS SDK verwenden, um Geheimnisse und Parameter als Umgebungsvariablen zu konfigurieren.

Themen

- Voraussetzungen

- Verwenden der Konsole
- Konfiguration mithilfe von Dateien in den Dateierweiterungen.ebextensions
- Konfiguration mit dem AWS CLI
- Konfiguration mit dem SDK AWS

Voraussetzungen

Bevor Sie Ihre Umgebungsvariablen so einrichten können, dass sie auf Secrets verweisen, müssen Sie zunächst die folgenden Schritte ausführen.

Allgemeines Verfahren vor der Konfiguration der Umgebungsvariablen

1. Erstellen Sie die Secrets Manager Manager-Geheimnisse oder die Parameter Store-Parameter, um Ihre sensiblen Daten zu speichern. Weitere Informationen finden Sie in einem oder beiden der folgenden Themen:
   - Geheimnisse erstellen in the section called “Verwendung von Secrets Manager”
   - Parameter erstellen in the section called “Verwendung des Systems Manager Parameter Store”
2. Richten Sie die erforderlichen IAM-Berechtigungen für die EC2-Instances Ihrer Umgebung ein, um die Geheimnisse und Parameter abzurufen. Weitere Informationen finden Sie unter Erforderliche IAM-Berechtigungen.

Verwenden der Konsole

Sie können die Elastic Beanstalk Beanstalk-Konsole verwenden, um Secrets als Umgebungsvariablen zu konfigurieren.

So konfigurieren Sie Secrets als Umgebungsvariablen in der Elastic Beanstalk Beanstalk-Konsole

1. Öffnen Sie die Elastic Beanstalk Beanstalk-Konsole und wählen Sie in der Liste Regionen Ihre aus. AWS-Region
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus und wählen Sie dann in der Liste den Namen Ihrer Umgebung aus.
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Updates, monitoring and logging (Updates, Überwachung und Protokolle) die Option Edit (Bearbeiten) aus.

5. Scrollen Sie nach unten zu Runtime-Umgebungsvariablen.
6. Wählen Sie Umgebungsvariable hinzufügen aus.
7. Wählen Sie als Quelle entweder Secrets Manager oder SSM Parameter Store aus.

   > **Note**
   >
   > Weitere Informationen zur Option Klartext in der Dropdownliste finden Sie unterKonfiguration von Umgebungseigenschaften (Umgebungsvariablen).

8. Geben Sie unter Name der Umgebungsvariablen den Namen der Umgebungsvariablen ein, die den geheimen Wert oder den Parameterwert enthalten soll.
9. Geben Sie als Wert für die Umgebungsvariable den ARN des Systems Manager-Parameterspeicher-Parameters oder das Secrets Manager Manager-Geheimnis ein. Beim Instance-Bootstrapping initiiert Elastic Beanstalk den Wert der Variablen, die Sie in Schritt 8 eingegeben haben, auf den Wert, der in dieser ARN-Ressource gespeichert ist.

   Die Konsole überprüft, ob der von Ihnen eingegebene Wert ein gültiges ARN-Format für den Store ist, den Sie in Schritt 7 ausgewählt haben. Es überprüft jedoch nicht, ob die im ARN angegebene Ressource vorhanden ist oder ob Sie über die erforderlichen IAM-Berechtigungen für den Zugriff darauf verfügen.
10. Wenn Sie weitere Variablen hinzufügen müssen, wiederholen Sie Schritt 6 bis Schritt 9.
11. Wählen Sie unten auf der Seite die Option Apply (Anwenden) aus, um die Änderungen zu speichern.

Konfiguration mithilfe von Dateien in den Dateierweiterungen.ebextensions

Sie können Elastic Beanstalk Beanstalk-Konfigurationsdateien verwenden, um Geheimnisse als Umgebungsvariablen zu konfigurieren. Verwenden Sie den aws:elasticbeanstalk:application:environmentsecrets Namespace, um Umgebungseigenschaften zu definieren.

> **Note**
>
> Secrets Manager hängt automatisch 6 zufällige Zeichen an geheime Namen im ARN-Format an, um die Eindeutigkeit zu gewährleisten.

Example.ebextensions/options.config für Umgebungsgeheimnisse (Kurzsyntax)

```
option_settings:
  aws:elasticbeanstalk:application:environmentsecrets:
    MY_SECRET: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-AbCd12
    MY_PARAMETER: arn:aws:ssm:us-east-1:111122223333:parameter/myparam
```

Example.ebextensions/options.config für Umgebungsgeheimnisse (Standardsyntax)

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environmentsecrets
    option_name: MY_SECRET
    value: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-AbCd12
  - namespace: aws:elasticbeanstalk:application:environmentsecrets
    option_name: MY_PARAMETER
    value: arn:aws:ssm:us-east-1:111122223333:parameter/myparam
```

Konfiguration mit dem AWS CLI

Sie können die AWS Befehlszeilenschnittstelle (AWS CLI) verwenden, um Secrets als Elastic Beanstalk Beanstalk-Umgebungsvariablen zu konfigurieren. Dieser Abschnitt enthält Beispiele für die Befehle create-environment und update-environment mit dem Namespace. aws:elasticbeanstalk:application:environmentsecrets Wenn Elastic Beanstalk die EC2-Instances für die Umgebungen bootet, auf die diese Befehle verweisen, initialisiert es die Umgebungsvariablen mit dem abgerufenen Geheimnis und den Parameterwerten. Es ruft diese Werte aus dem jeweiligen ARNs Secrets Manager und Systems Manager Parameter Store ab.

In den beiden folgenden Beispielen wird der Befehl create-environment verwendet, um ein Geheimnis und einen Parameter hinzuzufügen, die als Umgebungsvariablen mit dem Namen und konfiguriert sind. `MY_SECRET` `MY_PARAMETER`

Example von create-environment mit Geheimnissen, die als Umgebungsvariablen konfiguriert sind (Inline-Namespace-Optionen)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
```

```
--platform-arn "arn:aws:elasticbeanstalk:::platform/Node.js 24 running on 64bit Amazon
 Linux 2023" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-
elasticbeanstalk-ec2-role \
Namespace=aws:elasticbeanstalk:application:environmentsecrets,OptionName=MY_SECRET,Value=arn:aw
east-1:111122223333:secret:mysecret-AbCd12 \
Namespace=aws:elasticbeanstalk:application:environmentsecrets,OptionName=MY_PARAMETER,Value=arn
east-1:111122223333:parameter/myparam
```

Verwenden Sie alternativ eine`options.json`-Datei, um die Namespace-Optionen anzugeben, anstatt sie einzubinden.

Example von create-environment mit Geheimnissen, die als Umgebungsvariablen konfiguriert sind (Namespace-Optionen in der Datei) **`options.json`**

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--platform-arn "arn:aws:elasticbeanstalk:::platform/Node.js 24 running on 64bit Amazon
 Linux 2023" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:elasticbeanstalk:application:environmentsecrets",
    "OptionName": "MY_SECRET",
    "Value": "arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-AbCd12"
  },
  {
    "Namespace": "aws:elasticbeanstalk:application:environmentsecrets",
    "OptionName": "MY_PARAMETER",
```

```
    "Value": "arn:aws:ssm:us-east-1:111122223333:parameter/myparam"
  }
]
```

Im nächsten Beispiel werden Umgebungsvariablen mit dem Namen `MY_SECRET` und `MY_PARAMETER` so konfiguriert, dass sie ein Geheimnis und einen Parameter für eine bestehende Umgebung speichern. Der Befehl update-environment übergibt Optionen mit derselben Syntax wie der `create-environment` Befehl, entweder direkt oder mit einer Datei. `options.json` Das folgende Beispiel demonstriert, dass der Befehl dieselbe `options.json` Datei verwendet, die auch im vorherigen Beispiel verwendet wurde.

Example von update-environment mit Geheimnissen, die als Umgebungsvariablen konfiguriert sind (Namespace-Optionen in der Datei) **`options.json`**

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--platform-arn "arn:aws:elasticbeanstalk:::platform/Node.js 24 running on 64bit Amazon
 Linux 2023" \
--option-settings file://options.json
```

Konfiguration mit dem SDK AWS

Sie können Geheimnisse und Parameter als Umgebungsvariablen mit dem konfigurieren AWS SDKs. Ähnlich wie bei den im vorherigen Abschnitt erwähnten `create-environment` AWS CLI Befehlen `update-environment` und können Sie die UpdateEnvironmentAPI-Aktionen CreateEnvironmentund verwenden. Verwenden Sie den `OptionSettings` Anforderungsparameter, um die Optionen des aws:elasticbeanstalk:application:environmentsecrets Namespaces anzugeben.

Extrahieren von JSON-Schlüsseln aus Secrets Manager Manager-Geheimnissen

Plattformversionen, die am oder nach dem 13. Januar 2026 veröffentlicht wurden, unterstützen das Extrahieren bestimmter Felder aus Secrets Manager im JSON-Format, indem ein Doppelpunkt und der JSON-Schlüsselname an den geheimen ARN angehängt werden. Auf diese Weise können Sie innerhalb eines Geheimnisses auf einzelne Schlüssel-Wert-Paare verweisen und nicht auf das gesamte Geheimnis.

Syntax

Um einen bestimmten JSON-Schlüssel aus einem Secret zu extrahieren, hängen Sie ihn `:json-key-name` an den geheimen ARN an:

```
arn:aws:secretsmanager:region:account:secret:secret-name-XXXXXX:json-key-name
```

Die JSON-Schlüsselextraktion kann mit den gleichen Methoden wie normale Umgebungsgeheimnisse konfiguriert werden: Konsole, Konfigurationsdateien in `.ebextensions`, AWS CLI oder AWS SDKs.

Einschränkungen

- Die Extraktion von JSON-Schlüsseln wird nur für Secrets Manager Manager-Geheimnisse unterstützt, nicht für Systems Manager Parameter Store-Parameter.
- Nur JSON-Schlüssel der obersten Ebene werden unterstützt. Der Zugriff auf verschachtelte Schlüssel (z. B.**`config.database.host`**) und die Array-Indizierung (z. B.**`servers[0]`**) werden nicht unterstützt. Wenn Sie auf verschachtelte Werte zugreifen müssen, extrahieren Sie das übergeordnete Objekt und analysieren Sie es in Ihrem Anwendungscode. Verschachtelte Objekte und Arrays, auf die mit dem Schlüssel der obersten Ebene zugegriffen wird, werden zurück in das JSON-Format serialisiert.
- JSON-Schlüsselnamen dürfen keine Doppelpunkte () enthalten. `:` Doppelpunkte sind in der ARN-Syntax als Trennzeichen reserviert. Wenn ein JSON-Schlüsselname einen Doppelpunkt enthält, wird nur der Teil vor dem ersten Doppelpunkt als Schlüsselname verwendet, und alle Zeichen nach dem Doppelpunkt werden ignoriert.
- Die von ECS verwaltete Docker-Plattform verwendet die native ECS-Syntax für die Referenzierung von Geheimnissen. Weitere Informationen finden Sie unter Weitergabe von Secrets Manager Manager-Geheimnissen durch Amazon ECS-Umgebungsvariablen im Amazon Elastic Container Service Developer Guide.

Beispiel

Das folgende Beispiel zeigt, wie verschiedene Werttypen aus einem JSON-Secret extrahiert werden. Angesichts dieses Geheimnisses:

```
{
  "stringKey": "value1",
```

```
  "numberKey": 42,
  "objectKey": {
    "nested": "data"
  },
  "arrayKey": ["item1", "item2"]
}
```

Sie können die Extraktion konfigurieren in `.ebextensions`:

Example.ebextensions — Beispiel mit JSON-Schlüsselextraktion

```
option_settings:
  aws:elasticbeanstalk:application:environmentsecrets:
    STRING_VAR: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-
AbCd12:stringKey
    NUMBER_VAR: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-
AbCd12:numberKey
    OBJECT_VAR: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-
AbCd12:objectKey
    ARRAY_VAR: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-
AbCd12:arrayKey
```

Dies führt zu den folgenden Umgebungsvariablen:

- `STRING_VAR=value1`
- `NUMBER_VAR=42`
- `OBJECT_VAR={"nested":"data"}`
- `ARRAY_VAR=["item1","item2"]`

Bewährte Methoden für die Synchronisation von Geheimnissen mit Elastic Beanstalk Beanstalk-Umgebungsvariablen

In diesem Thema werden bewährte Methoden für Ihre Anwendung zur Verwendung von Umgebungsgeheimnissen mit Secrets Manager oder dem Systems Manager Parameter Store empfohlen. Ihre Elastic Beanstalk Beanstalk-Anwendung erhält nicht automatisch aktualisierte Werte, wenn die Secret Store-Daten aktualisiert oder rotiert werden. Elastic Beanstalk ruft Geheimnisse nur zum Zeitpunkt des Instance-Bootstrappings in Umgebungsvariablen ab.

Aktualisieren Sie Ihre Umgebungsvariablen

Um Ihre Elastic Beanstalk Beanstalk-Umgebung dazu zu veranlassen, die neuesten Werte der Secrets aus ihren Secret Stores erneut abzurufen, empfehlen wir, entweder die Operation oder auszuführen. `UpdateEnvironment` `RestartAppServer` Sie können diese Operationen mit der Elastic Beanstalk Beanstalk-Konsole AWS CLI, der oder der Elastic Beanstalk Beanstalk-API ausführen. Weitere Informationen finden Sie in den AWS CLI Beispielen für Elastic Beanstalk oder in der AWS Elastic Beanstalk API-Referenz.

Verwaltung von Auto-Scaling-Effekten bei geheimer Synchronisation

Wenn nach der Aktualisierung des Secret Store ein Scale-Out-Ereignis eintritt oder eine Instanz ersetzt wird, hat die neue Instance, die geöffnet wird, die neuesten geheimen Werte von Secrets Manager oder Systems Manager Parameter Store. Ein solches Ereignis kann auch dann eintreten, wenn nicht alle anderen Instanzen in der Umgebung aktualisiert wurden, um die neuen Secrets abzurufen.

> **Important**
>
> Sie müssen sicherstellen, dass Ihre Anwendung zwei verschiedene geheime Werte für dieselbe Umgebungsvariable verwenden kann. Dies eignet sich für Ereignisse, bei denen ein geheimes Update in Secrets Manager oder Systems Manager Parameter Store erfolgt, gefolgt von einer Skalierung oder einem Instanzersatz in Ihrer Umgebung, während die anderen Instanzen noch auf die Aktualisierung der Umgebungsvariablen warten. Während der Wartezeit für die Aktualisierung haben nicht alle Umgebungsinstanzen dieselben Werte für die Umgebungsvariablen des geheimen Speichers.

Ein Beispiel für einen solchen Anwendungsfall ist die Rotation von Datenbankanmeldedaten. Wenn auf die Rotation der Anmeldeinformationen ein Scale-Out-Ereignis folgt, enthalten die Umgebungsgeheimnisse, auf die die neu gestarteten Instanzen verweisen, die aktualisierten Datenbankanmeldedaten. Die Umgebungsgeheimnisse, auf die von den vorhandenen Instanzen verwiesen wird, behalten jedoch den alten Wert, bis sie durch die OR-Operationen aktualisiert werden. `UpdateEnvironment` `RestartAppServer`

Mehrzeilige Werte in Amazon Linux 2-Umgebungsvariablen

Mehrzeilige Werte bestehen aus mehr als einer Zeile und enthalten ein Zeilenumbruchzeichen. Mit Ausnahme von Docker und ECS-basierten Docker-Plattformen unterstützen Plattformen, die auf Amazon Linux 2 laufen, keine mehrzeiligen Werte für Umgebungsvariablen

> **Note**
>
> Elastic Beanstalk schlägt bei der Bereitstellung der betroffenen Umgebungen fehl, wenn ein mehrzeiliger Wert erkannt wird.

Die folgenden Optionen können als Behelfslösung oder als Lösung für das Problem mit mehreren Zeilen dienen:

- Führen Sie ein Upgrade Ihrer Amazon Linux 2-Umgebung auf Amazon Linux 2023 durch. Weitere Informationen finden Sie unter Migration von Amazon Linux 2 zu Amazon Linux 2023.
- Entfernen Sie Zeilenumbruchzeichen aus Ihren geheimen Werten. Ein Beispiel für einen Ansatz besteht darin, Ihre Werte mit Base64 zu codieren, bevor Sie sie im geheimen Speicher speichern. Ihre Anwendung müsste dann den Wert wieder in das ursprüngliche Format dekodieren, wenn sie aus der geheimen Umgebungsvariablen auf ihn verweist.
- Entwerfen Sie Ihren Anwendungscode so, dass die Daten direkt aus Secrets Manager oder Systems Manager Parameter Store abgerufen werden. Weitere Informationen finden Sie unter Abrufen von Geheimnissen in Verwendung von Secrets Manager oder Abrufen von Parametern. Verwendung des Systems Manager Parameter Store

Erforderliche IAM-Berechtigungen für Elastic Beanstalk für den Zugriff auf Geheimnisse und Parameter

Sie müssen den EC2-Instances Ihrer Umgebung die erforderlichen Berechtigungen erteilen, um die Geheimnisse und Parameter für und den Parameterspeicher abzurufen. AWS Secrets Manager AWS Systems Manager Berechtigungen werden den EC2-Instances über eine EC2-Instance-Profilrolle zur Verfügung gestellt.

In den folgenden Abschnitten sind die spezifischen Berechtigungen aufgeführt, die Sie einem EC2-Instance-Profil hinzufügen müssen, je nachdem, welchen Dienst Sie verwenden. Folgen Sie den

Schritten unter Aktualisieren der Berechtigungsrichtlinie für eine Rolle im IAM-Benutzerhandbuch, um diese Berechtigungen hinzuzufügen.

> **IAM-Berechtigungen für die von ECS verwaltete Docker-Plattform**
>
> Für die von ECS verwaltete Docker-Plattform sind zusätzlich zu den in diesem Thema angegebenen IAM-Berechtigungen weitere IAM-Berechtigungen erforderlich. Weitere Informationen zu allen erforderlichen Berechtigungen für Ihre von ECS verwaltete Docker-Plattformumgebung zur Unterstützung der Integration von Elastic Beanstalk Beanstalk-Umgebungsvariablen mit Geheimnissen finden Sie unter. ARN-Format der Ausführungsrolle

Themen

- Erforderliche IAM-Berechtigungen für Secrets Manager
- Erforderliche IAM-Berechtigungen Systems Manager Parameter Store

Erforderliche IAM-Berechtigungen für Secrets Manager

Die folgenden Berechtigungen gewähren Zugriff auf das Abrufen verschlüsselter Geheimnisse aus dem AWS Secrets Manager Store:

- Geheimnismanager: GetSecretValue
- kms:Decrypt

Die Erlaubnis zum Entschlüsseln eines AWS KMS key ist nur erforderlich, wenn Ihr Geheimnis einen vom Kunden verwalteten Schlüssel anstelle des Standardschlüssels verwendet. Durch das Hinzufügen Ihres benutzerdefinierten Schlüssels ARN erhalten Sie die Erlaubnis, den vom Kunden verwalteten Schlüssel zu entschlüsseln.

Example Richtlinie mit Secrets Manager- und KMS-Schlüsselberechtigungen

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:my-secret",
                "arn:aws:kms:us-east-1:111122223333:key/my-key"
            ]
        }
    ]
}
```

Erforderliche IAM-Berechtigungen Systems Manager Parameter Store

Die folgenden Berechtigungen gewähren Zugriff auf das Abrufen verschlüsselter Parameter aus dem AWS Systems Manager Parameterspeicher:

- ssm: GetParameter
- kms:Decrypt

Die Berechtigung zum Entschlüsseln eines AWS KMS key ist nur für `SecureString` Parametertypen erforderlich, die einen vom Kunden verwalteten Schlüssel anstelle eines Standardschlüssels verwenden. Durch das Hinzufügen Ihres benutzerdefinierten Schlüssels ARN erhalten Sie die Erlaubnis, den vom Kunden verwalteten Schlüssel zu entschlüsseln. Die regulären Parametertypen, die nicht verschlüsselt sind `String` und `StringList` keine benötigen. AWS KMS key

Example Richtlinie mit Systems Manager und AWS KMS Schlüsselberechtigungen

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
```

```
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:parameter/my-parameter",
                "arn:aws:kms:us-east-1:111122223333:key/my-key"
            ]
        }
    ]
}
```

Verwendung AWS Secrets Manager und Speicherung von AWS Systems Manager Parametern

Dieses Thema bietet eine kurze Einführung in den AWS Systems Manager Parameterspeicher, Preisinformationen und Verweise, um mehr über das Erstellen und Abrufen von Geheimnissen sowohl mit der Konsole als auch mit programmgesteuerten Optionen zu erfahren. AWS Secrets Manager

Über Secrets Manager

AWS Secrets Manager hilft Ihnen dabei, Secrets während ihres gesamten Lebenszyklus zu verwalten, abzurufen und zu rotieren. Zu den geheimen Daten, die Sie verwalten können, gehören beispielsweise Datenbankanmeldedaten, Anwendungsanmeldedaten, OAuth Token und API-Schlüssel. Secrets Manager ermöglicht es Ihnen, einen automatischen Rotationsplan für Ihre Secrets zu konfigurieren.

Über Systems Manager Parameter Store

Parameter Store ist ein Tool in AWS Systems Manager. Es bietet sicheren, hierarchischen Speicher für die Verwaltung von Konfigurationsdaten und Geheimnissen. Sie können wichtige Konfigurationsdaten als Parameterwerte verwalten. Zu den Daten, die Sie mit Parameter Store verwalten können, gehören Amazon Machine Image (AMI) IDs, Lizenzcodes, Passwörter und Datenbankzeichenfolgen.

> **Preisgestaltung**
>
> Für die Nutzung von Secrets Manager und Systems Manager Parameter Store fallen Standardgebühren an. Weitere Informationen zur Preisgestaltung finden Sie auf den folgenden Websites:

- AWS Secrets Manager Preisgestaltung
- AWS Systems Manager Preisgestaltung (wählen Sie Parameter Store aus der Inhaltsliste aus)

Themen

- Secrets Manager zum Erstellen und Abrufen von Geheimnissen verwenden
- Verwenden des Systems Manager Manager-Parameterspeichers zum Erstellen und Abrufen von Parametern

Secrets Manager zum Erstellen und Abrufen von Geheimnissen verwenden

Sie können Secrets Manager mit der AWS Secrets Manager Konsole, dem oder dem AWS CLI AWS SDK erstellen und abrufen. In den folgenden Ressourcen erfahren Sie mehr über verschiedene Methoden zum Erstellen und Abrufen von Secrets Manager Manager-Geheimnissen.

Geheimnisse erstellen

- Konsole — Erstellen Sie im AWS Secrets Manager Benutzerhandbuch ein AWS Secrets Manager Geheimnis (Konsole)
- AWS CLI — AWS CLI im AWS Secrets Manager Benutzerhandbuch
- AWS SDK — AWS SDK im AWS Secrets Manager Benutzerhandbuch

Geheimnisse abrufen

- Konsole — Rufen Sie im AWS Secrets Manager Benutzerhandbuch einen geheimen Wert (Konsole) ab
- AWS CLI — Holen Sie sich einen geheimen Wert (AWS CLI) im AWS Secrets Manager Benutzerhandbuch
- AWS SDK — Codebeispiele für Secrets Manager, die AWS SDKs in der AWS SDK Code Examples Code Library verwendet werden
- Andere Methoden — Holen Sie sich Geheimnisse aus AWS Secrets Manager dem AWS Secrets Manager Benutzerhandbuch

Weitere Informationen zu AWS Secrets Manager finden Sie unter Was ist AWS Secrets Manager? im AWS Secrets Manager Benutzerhandbuch.

Verwenden des Systems Manager Manager-Parameterspeichers zum Erstellen und Abrufen von Parametern

Sie können Parameter Store-Parameter mit der AWS Systems Manager Konsole, dem oder dem AWS CLI AWS SDK erstellen und abrufen. In den folgenden Ressourcen erfahren Sie mehr über verschiedene Methoden zum Erstellen und Abrufen von Parameter Store-Parametern.

Erstellen von Parametern

- Konsole — Erstellen Sie einen Systems Manager Manager-Parameter (Konsole) im AWS Systems Manager Benutzerhandbuch
- AWS CLI — Erstellen Sie einen Systems Manager Manager-Parameter (AWS CLI) im AWS Systems Manager Benutzerhandbuch
- AWS SDK — Verwendung PutParameter mit einem AWS SDK oder AWS CLI in der Codebibliothek mit AWS SDK-Codebeispielen

Parameter werden abgerufen

- Konsole — Suche nach einem Parameter (Konsole) im AWS Systems Manager Benutzerhandbuch
- AWS CLI — Verwendung GetParameter mit einem AWS SDK oder AWS CLI in der AWS SDK-Codebeispiel-Codebibliothek
- AWS SDK — Verwendung GetParameter mit einem AWS SDK oder AWS CLI in der AWS SDK-Codebeispiel-Codebibliothek

Weitere Informationen finden Sie unter AWS Systems Manager -Parameterspeicher im Benutzerhandbuch für AWS Systems Manager .

Fehlerbehebung bei der Integration von Geheimnissen mit Elastic Beanstalk Beanstalk-Umgebungsvariablen

> **Testen Sie Amazon Q Developer CLI zur KI-gestützten Fehlerbehebung**
>
> Amazon Q Developer CLI kann Ihnen helfen, Umgebungsprobleme schnell zu beheben. Die Q CLI bietet Lösungen, indem sie den Umgebungsstatus überprüft, Ereignisse überprüft,

Protokolle analysiert und klärende Fragen stellt. Weitere Informationen und detaillierte Anleitungen finden Sie in den Blogs unter Troubleshooting Elastic Beanstalk Environments with Amazon Q Developer CLI. AWS

Dieser Abschnitt enthält Anleitungen zur Behebung von Problemen mit Geheimnissen in Ihrer Elastic Beanstalk Beanstalk-Umgebung.

Ereignis: Beim Instance-Deployment konnten ein oder mehrere Secrets nicht abgerufen werden

Diese Meldung weist darauf hin, dass Elastic Beanstalk eines oder mehrere der während der Anwendungsbereitstellung angegebenen Geheimnisse nicht abrufen konnte.

- Überprüfen Sie, ob die durch die ARN-Werte in Ihrer Umgebungsvariablenkonfiguration angegebenen Ressourcen vorhanden sind.
- Vergewissern Sie sich, dass Ihre Elastic Beanstalk EC2-Instance-Profilrolle über die erforderlichen IAM-Berechtigungen für den Zugriff auf die Ressourcen verfügt.
- Wenn dieses Ereignis durch den `RestartAppServer` Vorgang ausgelöst wurde, versuchen Sie nach der Behebung des Problems erneut, den `RestartAppServer` Anruf zu starten, um das Problem zu lösen.
- Wenn das Ereignis durch einen `UpdateEnvironment` Anruf ausgelöst wurde, wiederholen Sie den `UpdateEnvironment` Vorgang.

Beispiele für diese Befehle finden Sie unter AWS CLI Beispiele für Elastic Beanstalk. Weitere Informationen zu den API-Aktionen für diese Operationen finden Sie in der AWS Elastic Beanstalk API-Referenz.

Ereignis: Bei der Instanzbereitstellung wurden ein oder mehrere mehrzeilige Umgebungswerte erkannt, die für diese Plattform nicht unterstützt werden

Mehrzeilige Variablen werden für Amazon Linux 2-Plattformen nicht unterstützt, ausgenommen Docker- und ECS-verwaltete Docker-Plattformen. Die verfügbaren Optionen, um fortzufahren, finden Sie unter. Mehrzeilige Werte

Ereignis: CreateEnvironment schlägt fehl, wenn ein Geheimnis angegeben wird

Wenn ein `CreateEnvironment` Fehler auftritt und Sie Geheimnisse als Umgebungsvariablen haben, müssen Sie das zugrundeliegende Problem beheben und es dann

verwendenUpdateEnvironment, um das Umgebungs-Setup abzuschließen. Verwenden Sie es nichtRestartAppServer, da es in dieser Situation nicht ausreicht, um die Umgebung zum Laufen zu bringen. Beispiele für diese Befehle finden Sie unter AWS CLI Beispiele für Elastic Beanstalk. Weitere Informationen zu den API-Aktionen für diese Operationen finden Sie in der AWS Elastic Beanstalk API-Referenz.

Verwenden von Elastic Beanstalk mit Amazon VPC

In diesem Thema werden die Vorteile der Verwendung von VPC-Endpoints mit Elastic Beanstalk und die verschiedenen Arten von Konfigurationen, die Sie implementieren können, erklärt.

Sie können eine Amazon Virtual Private Cloud (Amazon VPC) verwenden, um ein sicheres Netzwerk für Ihre Elastic Beanstalk Beanstalk-Anwendung und zugehörige Ressourcen einzurichten. AWS Beim Erstellen Ihrer Umgebung wählen Sie VPC, Subnetze und Sicherheitsgruppen für Ihre Anwendungs-Instances und Load Balancer aus. Sie können die VPC-Konfiguration beliebig festlegen, solange sie die folgenden Voraussetzungen erfüllt.

VPC-Anforderungen

- Internetzugriff – Instances müssen über eine der folgenden Methoden auf das Internet zugreifen können:
  - Öffentliches Subnetz – Instances besitzen eine öffentliche IP-Adresse und verwenden für den Zugriff auf das Internet ein Internet-Gateway.
  - Privates Subnetz – Instances verwenden für den Zugriff auf das Internet ein NAT-Gerät.

> **Note**
>
> Wenn Sie in Ihrer VPC VPC-Endpunkte für die Herstellung von Verbindungen mit `elasticbeanstalk` und `elasticbeanstalk-health` konfigurieren, ist der Internetzugriff optional und nur erforderlich, wenn die Anwendung sie spezifisch benötigt. Wenn keine VPC-Endpunkte vorhanden sind, muss Ihre VPC Zugriff auf das Internet haben.
> Die Standard-VPC, die Elastic Beanstalk für Sie einrichtet, stellt einen Internetzugang bereit.

Elastic Beanstalk unterstützt keine Proxy-Einstellungen wie `HTTPS_PROXY` für die Konfiguration eines Web-Proxys.

- NTP – Instances in Ihrer Elastic Beanstalk-Umgebung verwenden das Network Time Protocol (NTP), um die Systemuhr zu synchronisieren. Wenn Instances nicht auf UDP-Port 123 kommunizieren können, ist die Uhr möglicherweise nicht synchron, wodurch Probleme mit den Elastic Beanstalk-Integritätsberichten verursacht werden. Stellen Sie sicher, dass Ihre VPC-Sicherheitsgruppen und Ihr Netzwerk eingehenden und ausgehenden UDP-Verkehr auf Port 123 ACLs zulassen, um diese Probleme zu vermeiden.

Das elastic-beanstalk-samplesRepository bietet CloudFormation Vorlagen, mit denen Sie eine VPC für Ihre Elastic Beanstalk Beanstalk-Umgebungen erstellen können.

Um Ressourcen mit einer Vorlage zu erstellen CloudFormation

1. Klonen Sie das Beispiel-Repository oder laden Sie über die Links in der README-Datei eine Vorlage herunter.
2. Öffnen Sie die CloudFormation -Konsole.
3. Wählen Sie Stack erstellen aus.
4. Wählen Sie Upload a template to Amazon S3 (Vorlage auf Amazon S3 hochladen) aus.
5. Wählen Sie Upload file (Datei hochladen) aus und laden Sie die Vorlagendatei von Ihrem lokalen Computer hoch.
6. Klicken Sie auf Next (Weiter) und befolgen Sie die Anweisungen zum Erstellen eines Stapels mit den Ressourcen in der Vorlage.

Wenn die Stack-Erstellung abgeschlossen ist, suchen Sie auf der Registerkarte Ausgaben nach der VPC-ID und dem IDs Subnetz. Verwenden Sie diese zum Konfigurieren der VPC in der neuen Netzwerkkonfigurationskategorie des Umgebungsassistenten.

Themen

- Öffentliche VPC
- Öffentliche/private VPC
- Private VPC
- Beispiel: Starten einer Elastic Beanstalk-Anwendung in einer VPC mit Bastion-Hosts
- Beispiel: Starten einer Elastic Beanstalk-Anwendung in einer VPC mit Amazon RDS
- Verwenden von Elastic Beanstalk mit VPC-Endpunkten
- Verwenden von Endpunktrichtlinien zur Steuerung des Zugriffs mit VPC-Endpunkten

Öffentliche VPC

CloudFormation Vorlage — vpc-public.yaml

Einstellungen (mit Load Balancing)

- Load balancer visibility (Load Balancer-Sichtbarkeit) – Öffentlich
- Load balancer subnets (Load Balancer-Subnetze) – Beide öffentliche Subnetze
- Instance public IP (Öffentliche IP der Instance) – Aktiviert
- Instance subnets (Instance-Subnetze) – Beide öffentliche Subnetze
- Instance security groups (Instance-Sicherheitsgruppen) – Standard-Sicherheitsgruppe hinzufügen

Einstellungen (einzelne Instance)

- Instance subnets (Instance-Subnetze) – Eines der öffentlichen Subnetze
- Instance security groups (Instance-Sicherheitsgruppen) – Standard-Sicherheitsgruppe hinzufügen

Ein einfaches rein öffentliches VPC-Layout umfasst mindestens ein öffentliches Subnetz, ein Internet-Gateway und eine Standardsicherheitsgruppe, die den Datenverkehr zwischen den Ressourcen in der VPC zulässt. Wenn Sie in der VPC eine Umgebung erstellen, generiert Elastic Beanstalk zusätzliche Ressourcen, die sich je nach Typ der Umgebung unterscheiden.

VPC-Ressourcen

- Einzelne Instance – Elastic Beanstalk erstellt eine Sicherheitsgruppe für die Anwendungs-Instance, die Datenverkehr aus dem Internet auf Port 80 zulässt, und weist der Instance eine Elastic IP-Adresse zu, damit sie eine öffentliche IP-Adresse hat. Der Domänenname der Umgebung wird zur öffentlichen IP-Adresse der Instance aufgelöst.
- Lastausgleich – Elastic Beanstalk erstellt eine Sicherheitsgruppe für den Load Balancer, der Datenverkehr aus dem Internet auf Port 80 zulässt, und eine Sicherheitsgruppe für die Anwendungs-Instances, die Datenverkehr aus der Sicherheitsgruppe für den Load Balancer zulassen. Der Domänenname der Umgebung wird zum öffentlichen Domänennamen für den Load Balancer aufgelöst.

Dies ähnelt der Art und Weise, wie Elastic Beanstalk die Netzwerke verwaltet, wenn Sie die Standard-VPC verwenden. Die Sicherheit in einem öffentlichen Subnetz hängt vom Load Balancer

und von den Instance-Sicherheitsgruppen ab, die von Elastic Beanstalk erstellt wurden. Dies ist die kostengünstigste Konfiguration, da kein NAT-Gateway erforderlich ist.

Öffentliche/private VPC

CloudFormation Vorlage — vpc-privatepublic.yaml

Einstellungen (mit Load Balancing)

- Load balancer visibility (Load Balancer-Sichtbarkeit) – Öffentlich
- Load balancer subnets (Load Balancer-Subnetze) – Beide öffentliche Subnetze
- Instance public IP (Öffentliche IP der Instance) – Deaktiviert
- Instance subnets (Instance-Subnetze) – Beide private Subnetze
- Instance security groups (Instance-Sicherheitsgruppen) – Standard-Sicherheitsgruppe hinzufügen

Fügen Sie Ihrer VPC zur Erhöhung der Sicherheit private Subnetze hinzu, um ein öffentlich/privat-Layout zu erstellen. Dieses Layout erfordert einen Load Balancer und ein NAT-Gateway in den öffentlichen Subnetzen. Es bietet Ihnen die Möglichkeit, Anwendungs-Instances, Datenbanken und andere Ressourcen in privaten Subnetzen auszuführen. Instances in privaten Subnetzen können nur über den Load Balancer und das NAT-Gateway mit dem Internet kommunizieren.

Private VPC

CloudFormation Vorlage — vpc-private.yaml

Einstellungen (mit Load Balancing)

- Load balancer visibility (Load Balancer-Sichtbarkeit) – Privat
- Load balancer subnets (Load Balancer-Subnetze) – Beide private Subnetze
- Instance public IP (Öffentliche IP der Instance) – Deaktiviert
- Instance subnets (Instance-Subnetze) – Beide private Subnetze
- Instance security groups (Instance-Sicherheitsgruppen) – Standard-Sicherheitsgruppe hinzufügen

Im Fall interner Anwendungen, auf die kein Zugriff aus dem Internet möglich sein darf, können Sie alle Vorgänge in privaten Subnetzen ausführen und den Load Balancer so konfigurieren, dass er nur intern verfügbar ist. Hierzu ändern Sie den Wert für Load balancer visibility (Load Balancer-Sichtbarkeit) in Internal (Intern). Diese Vorlage erstellt eine VPC ohne öffentliche Subnetze und

ohne Internet-Gateway. Verwenden Sie dieses Layout für Anwendungen, auf die der Zugriff nur über dieselbe VPC oder ein zugehöriges VPN möglich sein soll.

Ausführung einer Elastic Beanstalk-Umgebung in einer privaten VPC

Wenn Sie Ihre Elastic Beanstalk-Umgebung in einer privaten VPC erstellen, hat die Umgebung keinen Zugriff auf das Internet. Ihre Anwendung benötigt möglicherweise Zugriff auf den Elastic Beanstalk-Service oder andere Services. Ihre Umgebung könnte z. B. erweiterte Integritätsberichte verwenden. In diesem Fall senden die Umgebungs-Instances Integritätsdaten an den erweiterten Integritätsservice. Und Elastic Beanstalk Beanstalk-Code auf Umgebungsinstanzen sendet Traffic an andere AWS Services und anderen Traffic an AWS Nicht-Endpoints (zum Beispiel, um Abhängigkeitspakete für Ihre Anwendung herunterzuladen). Im Folgenden finden Sie einige Schritte, die Sie in diesem Fall möglicherweise ausführen müssen, um sicherzustellen, dass Ihre Umgebung ordnungsgemäß funktioniert.

- Konfigurieren von VPC-Endpunkten für Elastic Beanstalk – Elastic Beanstalk und der erweiterte Integritätsservice unterstützen VPC-Endpunkte. Diese stellen sicher, dass der Datenverkehr zu diesen Services ausschließlich innerhalb des Amazon-Netzwerks ausgeführt wird und keinen Internetzugriff erfordert. Weitere Informationen finden Sie unter the section called “VPC-Endpunkte”.
- VPC-Endpunkte für zusätzliche Services konfigurieren — Elastic Beanstalk Beanstalk-Instances senden in Ihrem Namen Traffic an mehrere andere AWS Services: Amazon Simple Storage Service (Amazon S3), Amazon Simple Queue Service (Amazon SQS) und Amazon Logs. AWS CloudFormation CloudWatch Sie müssen auch für diese Services VPC-Endpunkte konfigurieren. Detaillierte Informationen zu VPC-Endpunkten einschließlich Links pro Service finden Sie unter VPC-Endpunkte im Amazon VPC-Benutzerhandbuch.

  > **Note**
  >
  > Einige AWS Dienste, darunter Elastic Beanstalk, unterstützen VPC-Endpunkte in einer begrenzten Anzahl von Regionen. AWS Wenn Sie Ihre private VPC-Lösung entwerfen, stellen Sie sicher, dass Elastic Beanstalk und die anderen hier genannten abhängigen Dienste VPC-Endpoints in der AWS von Ihnen ausgewählten Region unterstützen.

- Bereitstellen eines privaten Docker-Image – In einer Docker-Umgebung könnte Code auf den Umgebungs-Instances versuchen, während der Erstellung der Umgebung das konfigurierte Docker-Image aus dem Internet abzurufen, und fehlschlagen. Um diesen Fehler zu vermeiden, erstellen Sie in Ihrer Umgebung ein benutzerdefiniertes Docker-Image oder verwenden ein in

Amazon Elastic Container Registry (Amazon ECR) gespeichertes Docker-Image und konfigurieren einen VPC-Endpunkt für den Amazon ECR-Service.

- DNS-Namen aktivieren — Elastic Beanstalk Beanstalk-Code auf Umgebungsinstanzen sendet Traffic an alle AWS Services, die ihre öffentlichen Endpunkte verwenden. Damit dieser Datenverkehr das Ziel erreicht, wählen Sie während der Konfigurierung der VPC-Schnittstellenendpunkte die Option Enable DNS name (DNS-Name aktivieren) aus. Hierdurch wird ein DNS-Eintrag in Ihrer VPC hinzugefügt, der den Endpunkt für den öffentlichen Service dem VPC-Schnittstellenendpunkt zuordnet.

  > **Important**
  >
  > Wenn Ihre VPC nicht privat ist, Zugang zum öffentlichen Internet hat und Enable DNS name (DNS-Name aktivieren) für einen VPC-Endpunkt deaktiviert ist, wird der Datenverkehr zum jeweiligen Dienst über das öffentliche Internet transportiert. Das wird wahrscheinlich nicht von Ihnen gewünscht. Sie können dieses Problem bei einer privaten VPC leicht erkennen, da dies verhindert, dass der Datenverkehr das Ziel erreicht, und Ihnen Fehler angezeigt werden. Bei einer öffentlichen VPC erhalten Sie jedoch keine Hinweise.

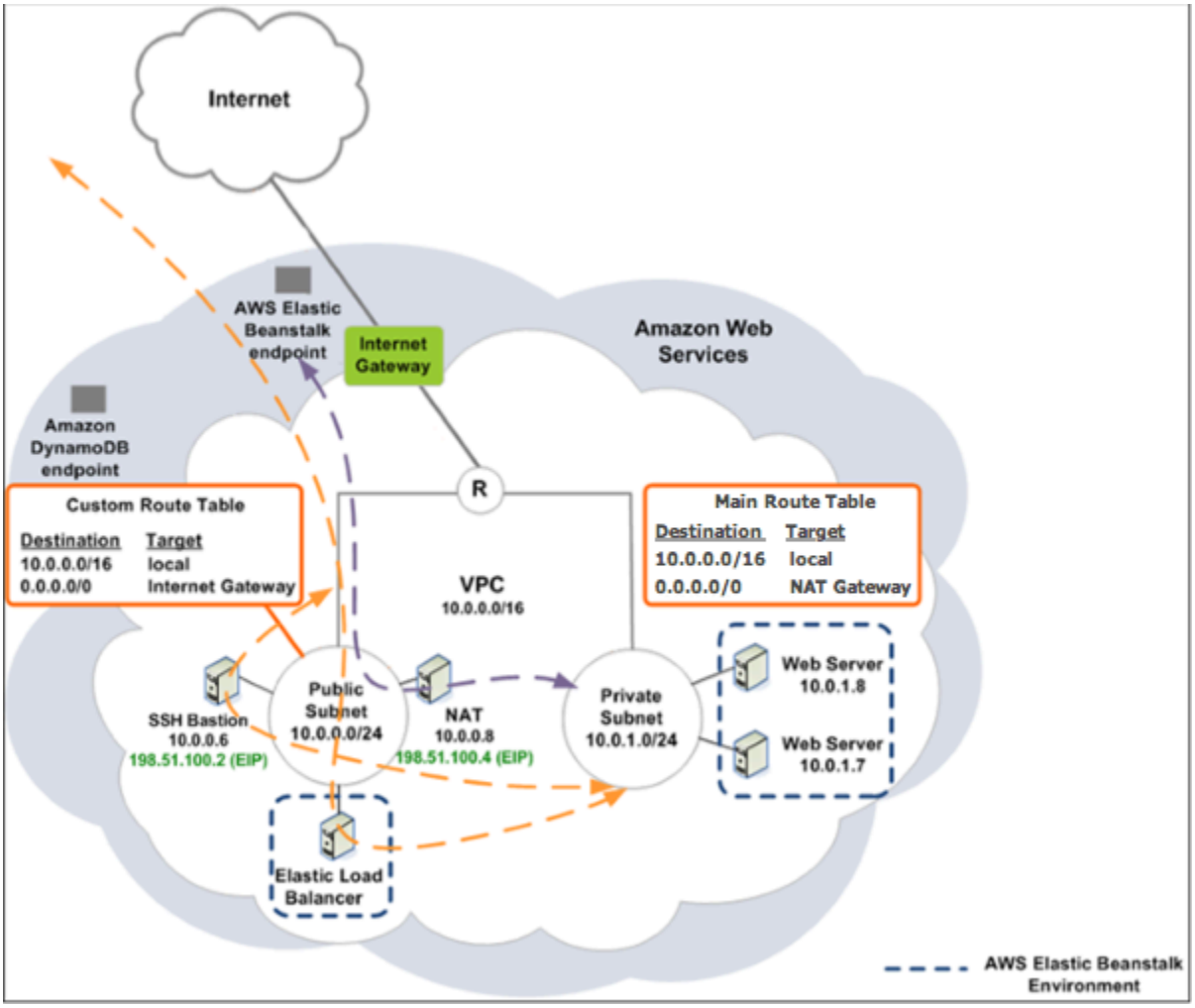
- Einschluss von Anwendungsabhängigkeiten – Wenn Ihre Anwendung Abhängigkeiten wie Sprachlaufzeitpakete besitzt, versucht sie möglicherweise, diese während der Erstellung der Umgebung aus dem Internet herunterzuladen und zu installieren, und schlägt fehl. Um diesen Fehler zu vermeiden, sollten Sie alle Abhängigkeitspakete in das Quellpaket Ihrer Anwendung einschließen.
- Verwenden einer aktuellen Plattformversion – Stellen Sie sicher, dass Ihre Umgebung eine Plattformversion verwendet, die am 24. Februar 2020 oder später veröffentlicht wurde. Sie sollten insbesondere eine Plattformversion verwenden, die mit oder nach einem der folgenden beiden Updates veröffentlicht wurde: Linux Update 2020-02-28, Windows Update 2020-02-24.

  > **Note**
  >
  > Eine aktuelle Plattformversion ist deswegen notwendig, weil in älteren Versionen ein Problem auftrat, das verhinderte, dass mit der Option Enable DNS name (DNS-Name aktivieren) erstellte DNS-Einträge für Amazon SQS ordnungsgemäß funktionieren.

Beispiel: Starten einer Elastic Beanstalk-Anwendung in einer VPC mit Bastion-Hosts

In diesem Abschnitt wird erklärt, wie Sie mithilfe eines Bastion-Hosts eine Elastic Beanstalk Beanstalk-Anwendung in einer VPC bereitstellen und warum Sie diese Topologie implementieren sollten.

Wenn sich Ihre EC2 Amazon-Instances in einem privaten Subnetz befinden, können Sie keine Remoteverbindung zu ihnen herstellen. Zum Herstellen einer Verbindung können Sie Bastion-Server im öffentlichen Subnetz herstellen, die als Proxys fungieren. Beispiel: Sie können SSH-Port-Weiterleitungen oder RDP-Gateways im öffentlichen Subnetz einrichten, um für den Datenverkehr an Ihre Datenbankserver aus Ihrem eigenen Netzwerk einen Proxy bereitzustellen. In diesem Abschnitt finden Sie ein Beispiel dafür, wie Sie eine VPC mit einem privaten und öffentlichen Subnetz erstellen. Die Instances befinden sich im privaten Subnetz und Bastion-Host, NAT-Gateway und Load Balancer befinden sich im öffentlichen Subnetz. Ihre Infrastruktur sollte dann wie das folgende Diagramm aussehen.



Um eine Elastic Beanstalk-Anwendung in einer VPC mithilfe eines Bastion-Host bereitzustellen, führen Sie die in den folgenden Unterabschnitten beschriebenen Schritte aus.

Schritte

- Erstellen einer VPC mit einem öffentlichen und privaten Subnetz
- Erstellen und Konfigurieren der Bastion-Host-Sicherheitsgruppe
- Aktualisieren der Instance-Sicherheitsgruppe
- Erstellen eines Bastion-Host

Erstellen einer VPC mit einem öffentlichen und privaten Subnetz

Führen Sie alle Verfahren in Öffentliche/private VPC durch. Bei der Bereitstellung der Anwendung müssen Sie ein EC2 Amazon-Schlüsselpaar für die Instances angeben, damit Sie sich remote mit ihnen verbinden können. Weitere Informationen darüber, wie Sie das Instance-Schlüsselpaar

angeben können, finden Sie unter Die EC2 Amazon-Instances für Ihre Elastic Beanstalk Beanstalk-Umgebung.

Erstellen und Konfigurieren der Bastion-Host-Sicherheitsgruppe

Erstellen Sie eine Sicherheitsgruppe für den Bastion-Host und fügen Sie Regeln hinzu, die eingehenden SSH-Verkehr aus dem Internet und ausgehenden SSH-Verkehr in das private Subnetz, das die Amazon-Instances enthält, zulassen. EC2

So erstellen Sie die Bastion-Host-Sicherheitsgruppe

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
3. Wählen Sie Sicherheitsgruppen erstellen aus.
4. Geben Sie im Dialogfeld Create Security Group (Sicherheitsgruppe erstellen) Folgendes ein und wählen Sie Yes, Create (Ja, erstellen) aus.

   Name tag (Namens-Tag) (optional)

   Geben Sie einen Namens-Tag für die Sicherheitsgruppe ein.

   Group name (Gruppenname)

   Geben Sie den Namens der Sicherheitsgruppe ein.

   Beschreibung

   Geben Sie eine Beschreibung für die Sicherheitsgruppe ein.

   VPC

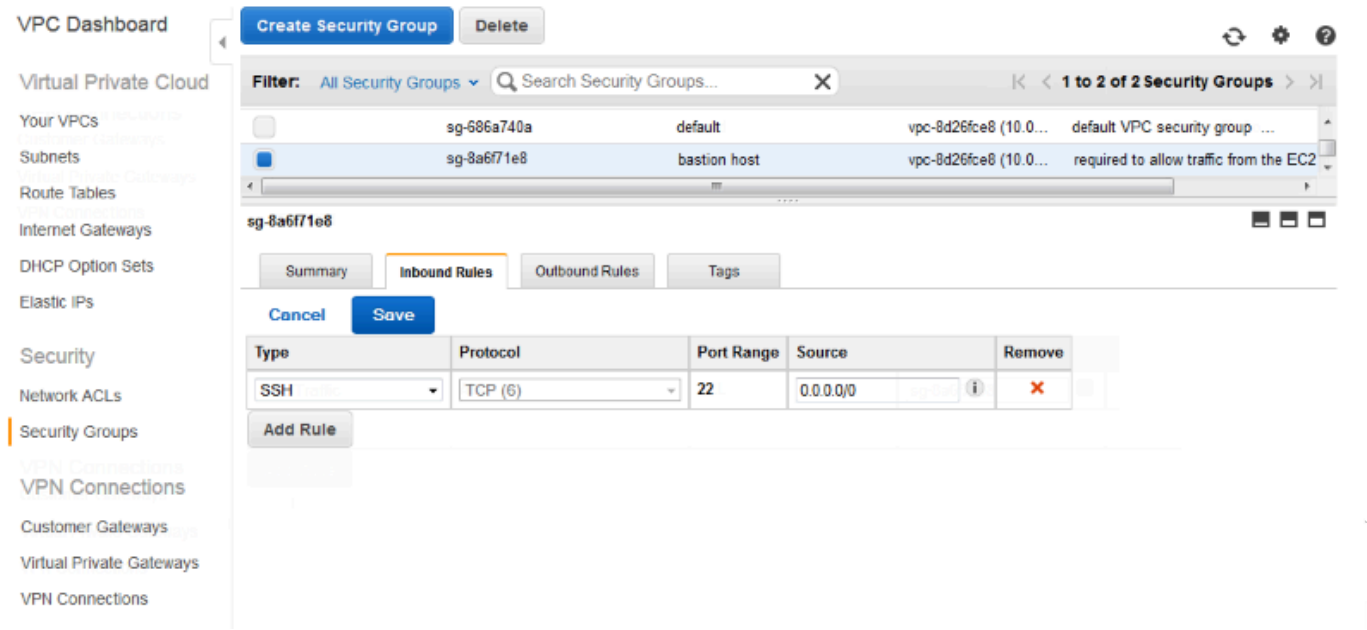
   Wählen Sie Ihre VPC aus.

   Die Sicherheitsgruppe wird erstellt und auf der Seite Security Groups (Sicherheitsgruppen) angezeigt. Die Gruppe ist mit einer ID versehen (z. B. `sg-xxxxxxxx`). Möglicherweise müssen Sie die Spalte Group ID (Gruppen-ID) aktivieren, indem Sie oben rechts auf der Seite auf Show/Hide (Einblenden/Ausblenden) klicken.

Konfigurieren der Bastion-Host-Sicherheitsgruppe

1. Aktivieren Sie in der Liste der Sicherheitsgruppen das Kontrollkästchen für die Sicherheitsgruppe, die Sie gerade für Ihren Bastion-Host erstellt haben.
2. Wählen Sie auf der Registerkarte Inbound Rules die Option Edit aus.
3. Wählen Sie bei Bedarf Add another rule (Weitere Regel hinzufügen) aus.
4. Wenn der Bastion-Host eine Linux-Instance ist, wählen Sie unter Type (Typ) die Option SSH aus.

   Wenn der Bastion-Host eine Windows-Instance ist, wählen Sie unter Type (Typ) die Option RDP aus.
5. Geben Sie den gewünschten Quell-CIDR-Bereich im Feld Source (Quelle) ein und wählen Sie Save (Speichern) aus.



6. Wählen Sie auf der Registerkarte Outbound Rules (Ausgehende Regeln) die Option Edit (Bearbeiten) aus.
7. Wählen Sie bei Bedarf Add another rule (Weitere Regel hinzufügen) aus.
8. Wählen Sie unter Type (Typ) den für die eingehende Regel angegebenen Typ aus.
9. Geben Sie in das Feld Source (Quelle) den CIDR-Bereich des Subnetzes der Hosts im privaten Subnetz der VPC ein.

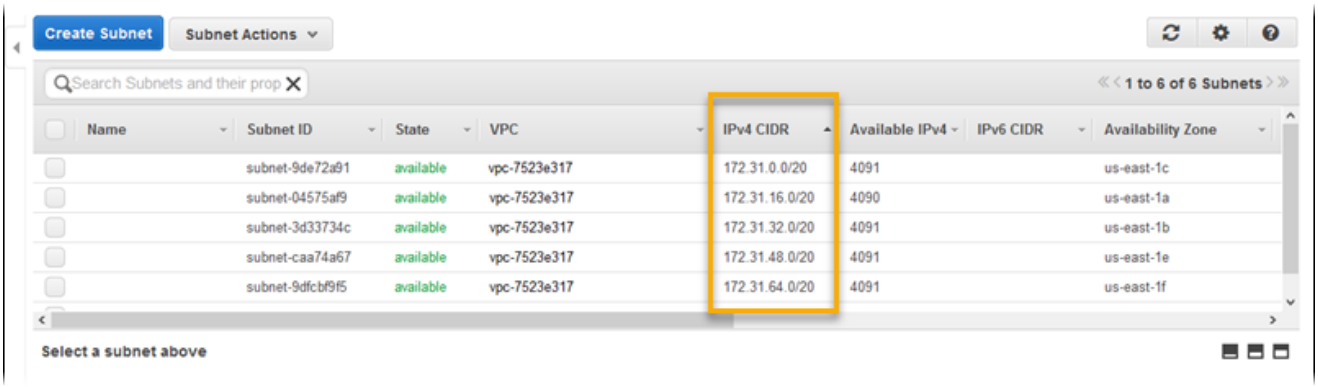
   So suchen Sie:

   a. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.

b. Wählen Sie im Navigationsbereich Subnetze aus.

c. Notieren Sie sich den Wert unter IPv4 CIDR für jede Availability Zone, in der Sie Hosts haben, zu denen der Bastion-Host eine Verbindung herstellen soll.

> **Note**
>
> Wenn Sie Hosts in mehreren Availability Zones haben, erstellen Sie eine Regel für ausgehenden Datenverkehr für jede dieser Availability Zones.



10. Wählen Sie Speichern.

Aktualisieren der Instance-Sicherheitsgruppe

Standardmäßig erlaubt die Sicherheitsgruppe, die Sie für Ihre Instances erstellt haben, eingehenden Datenverkehr nicht. Während Elastic Beanstalk die Standardgruppe so ändert, dass die Instances SSH-Datenverkehr zulassen, müssen Sie Ihre benutzerdefinierte Instance-Sicherheitsgruppe für das Zulassen von RDP-Datenverkehr ändern, wenn Ihre Instances Windows-Instances sind.

So aktualisieren Sie die Instance-Sicherheitsgruppe für RDP

1. Aktivieren Sie in der Liste der Sicherheitsgruppen das Kontrollkästchen für die Instance-Sicherheitsgruppe.
2. Klicken Sie auf die Registerkarte Inbound und wählen Sie Edit aus.
3. Wählen Sie bei Bedarf Add another rule (Weitere Regel hinzufügen) aus.
4. Geben Sie die folgenden Werte ein und wählen Sie Save (Speichern) aus.

Typ

RDP

Protocol (Protokoll)

TCP

Port Range (Port-Bereich)

3389

Quelle

Geben Sie die ID der Bastion-Host-Sicherheitsgruppe ein (z. B. `sg-8a6f71e8`) und wählen Sie Save (Speichern) aus.

Erstellen eines Bastion-Host

Um einen Bastion-Host zu erstellen, starten Sie eine EC2 Amazon-Instance in Ihrem öffentlichen Subnetz, die als Bastion-Host fungiert.

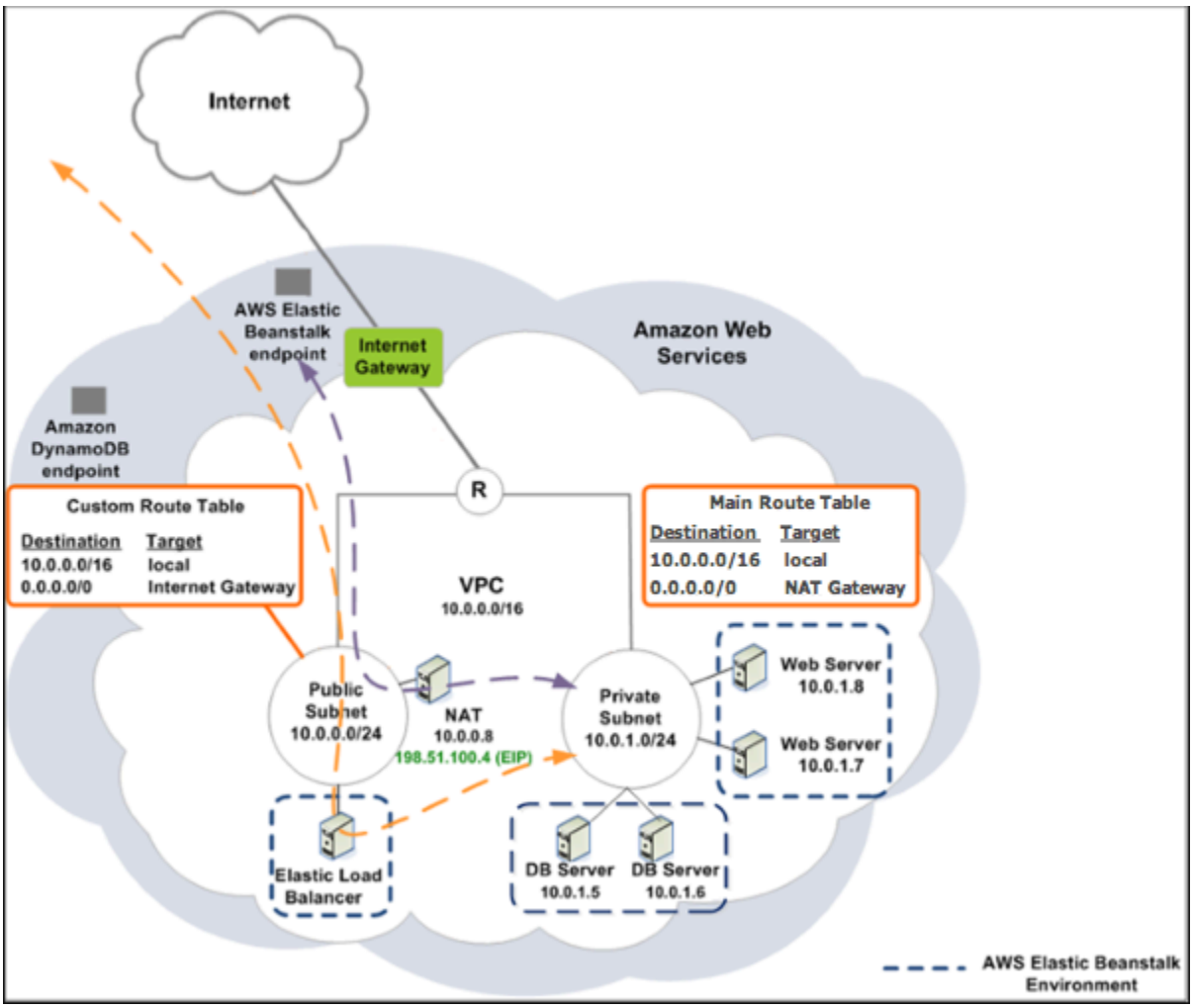
Weitere Informationen zur Einrichtung eines Bastion-Hosts für Windows-Instances im privaten Subnetz finden Sie unter Steuern des Netzwerkzugriffs auf EC2 Instances mithilfe eines Bastion-Servers.

Weitere Informationen zum Einrichten eines Bastion-Host für Linux-Instances im privaten Subnetz finden Sie unter Securely Connect to Linux Instances Running in a Private Amazon VPC.

Beispiel: Starten einer Elastic Beanstalk-Anwendung in einer VPC mit Amazon RDS

In diesem Abschnitt werden Sie schrittweise durch die Aufgaben geführt, die für die Bereitstellung einer Elastic Beanstalk-Anwendung mit Amazon RDS in einer VPC über ein NAT-Gateway notwendig sind.

Ihre Infrastruktur sollte dann wie das folgende Diagramm aussehen.



> **Note**
>
> Wenn Sie noch keine DB-Instance mit Ihrer Anwendung verwendet haben, versuchen Sie, eine zu einer Testumgebung hinzuzufügen und eine Verbindung mit einer externen DB-Instance herzustellen, bevor Sie eine VPC-Konfiguration hinzufügen.

Erstellen einer VPC mit einem öffentlichen und privaten Subnetz

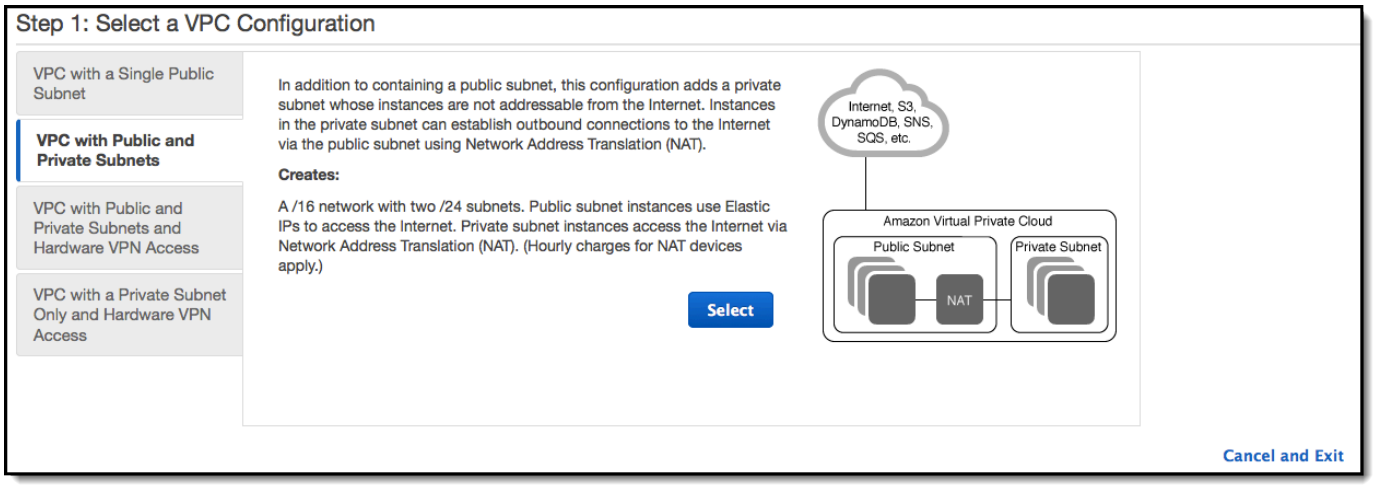
Sie können die Amazon VPC-Konsole zum Erstellen einer VPC verwenden.

So erstellen Sie eine VPC

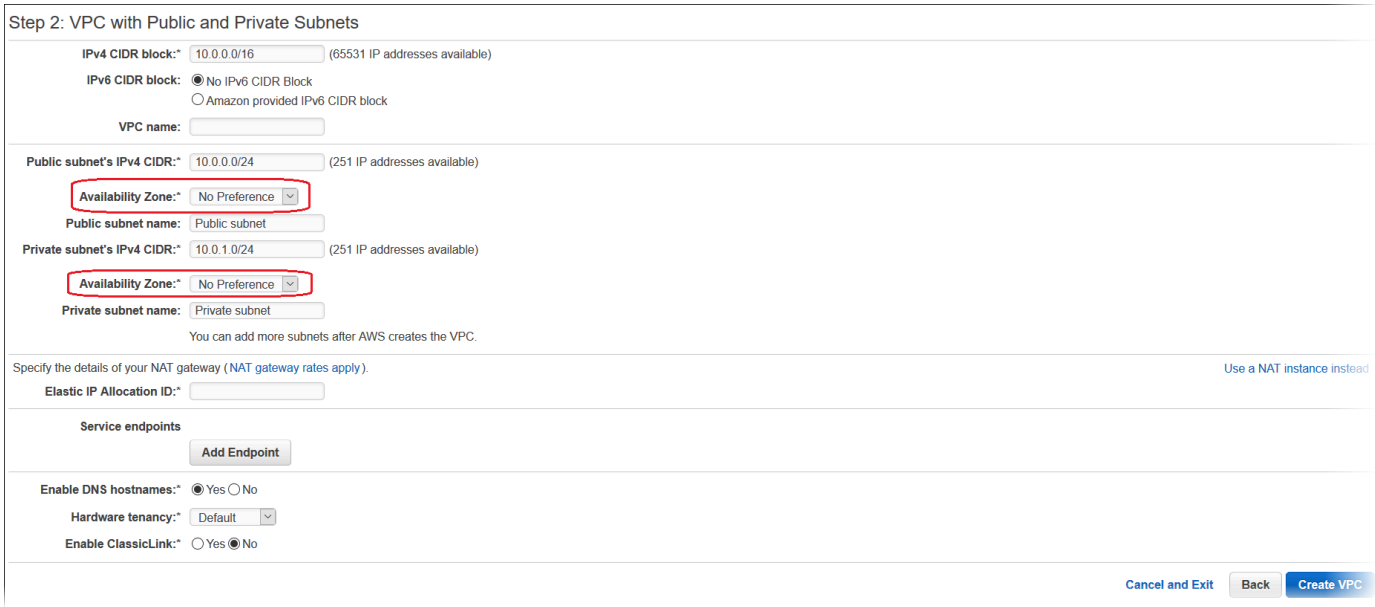
1. Melden Sie sich bei der Amazon VPC-Konsole an.

2. Wählen Sie im Navigationsbereich VPC Dashboard (VPC-Dashboard) aus. Wählen Sie dann die Option Create VPC (VPC erstellen).

3. Klicken Sie auf VPC with Public and Private Subnets (VPC mit öffentlichen und privaten Subnetzen) und anschließend auf Select (Auswählen).



4. Ihr Elastic Load Balancing Load Balancer und Ihre EC2 Amazon-Instances müssen sich in derselben Availability Zone befinden, damit sie miteinander kommunizieren können. Wählen Sie in jeder Liste Availability Zone dieselbe Availability Zone aus.



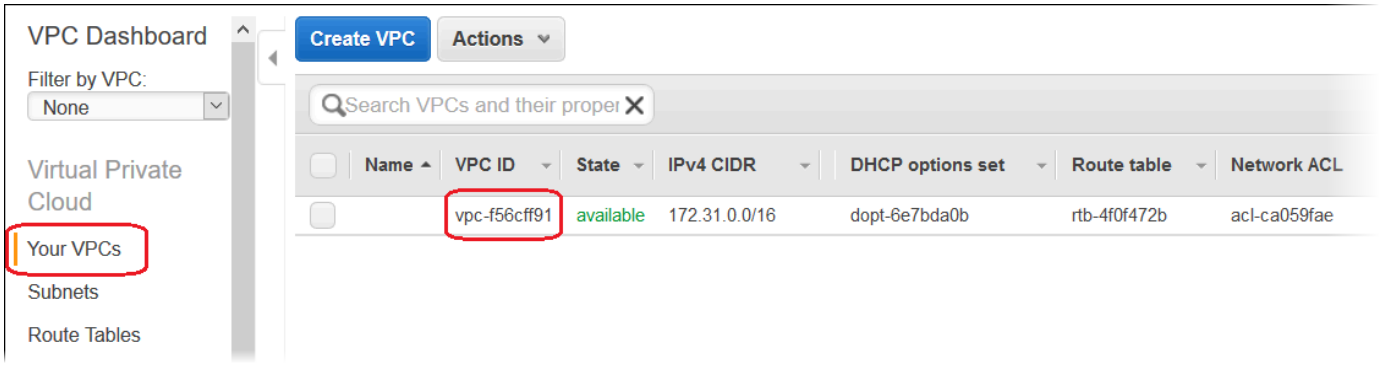
5. Wählen Sie eine Elastic IP-Adresse für Ihr NAT-Gateway aus.

6. Wählen Sie VPC erstellen aus.

Der Assistent erstellt Ihre VPC, Subnetze und das Internet-Gateway. Außerdem wird die Haupt-Routing-Tabelle aktualisiert und eine benutzerdefinierte Routing-Tabelle erstellt. Zum Schluss erstellt der Assistent ein NAT-Gateway im öffentlichen Subnetz.

> **Note**
>
> Sie können auswählen, ob eine NAT-Instance statt im NAT-Gateway im öffentlichen Subnetz starten soll. Weitere Informationen finden Sie unter Szenario 2: VPC mit öffentlichen und privaten Subnetzen (NAT) im Amazon VPC-Benutzerhandbuch.

7. Nachdem die VPC erfolgreich erstellt wurde, erhalten Sie eine VPC-ID. Sie benötigen diesen Wert im nächsten Schritt. Um Ihre VPC-ID anzuzeigen, wählen Sie VPCs im linken Bereich der Amazon VPC-Konsole Ihre aus.



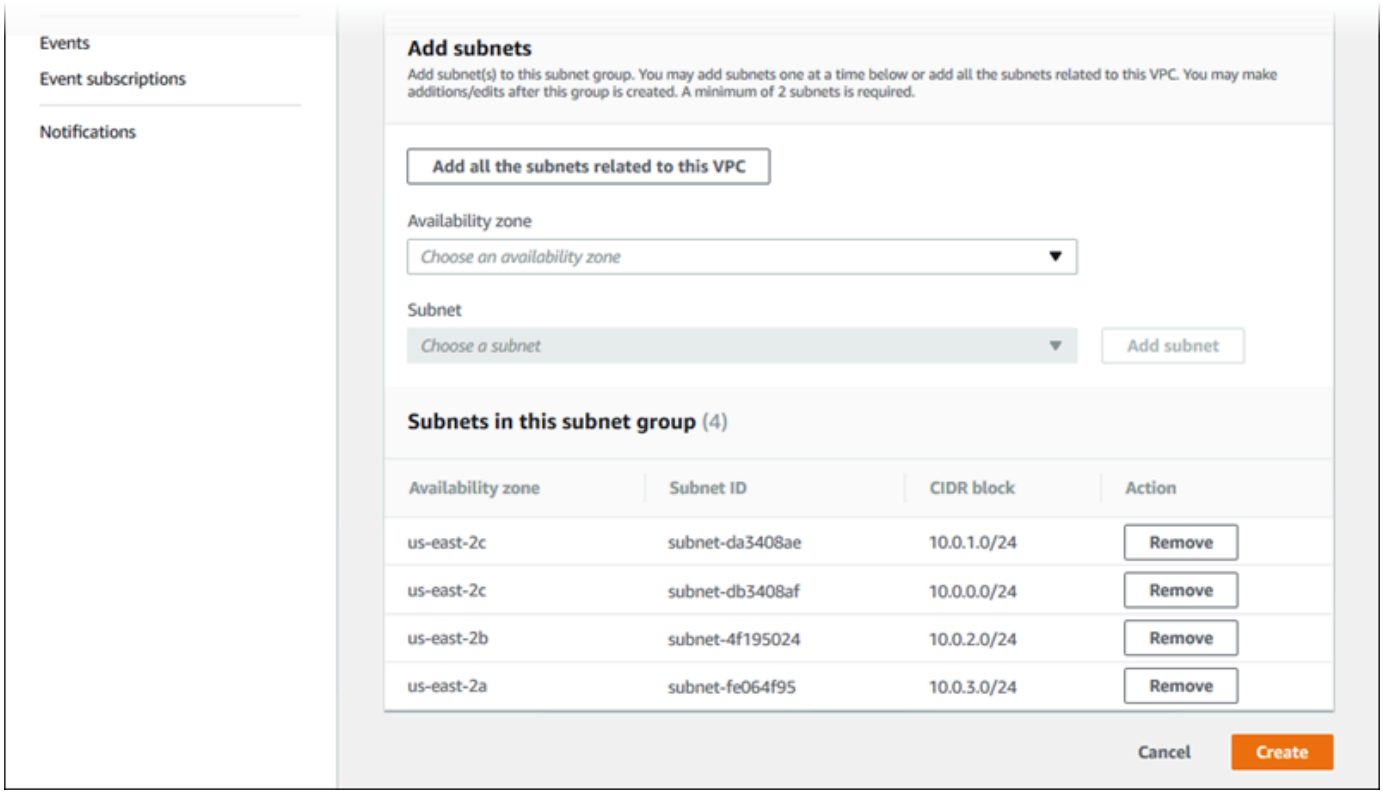
Erstellen einer DB-Subnetzgruppe

Eine DB-Subnetzgruppe für eine VPC ist eine Sammlung von Subnetzen (in der Regel private Subnetze), die Sie für Ihre Backend-RDS-DB-Instances bestimmen sollten. Jede DB-Subnetzgruppe sollte mindestens ein Subnetz für jede Availability Zone in einer bestimmten Region haben. AWS Weitere Informationen erhalten Sie unter Erstellen eines Subnetzes in Ihrer VPC.

Erstellen einer DB-Subnetzgruppe

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.
3. Wählen Sie DB-Subnetzgruppe erstellen aus.
4. Wählen Sie Name und geben Sie dann den Namen Ihrer DB-Subnetzgruppe ein.
5. Wählen Sie Description (Beschreibung) und beschreiben Sie dann Ihre DB-Subnetzgruppe.

6. Wählen Sie für VPC die ID der zuvor erstellten VPC aus.
7. Wählen Sie im Abschnitt Add subnets (Subnetze hinzufügen) die Option Add all the subnets related to this VPC (Alle zu dieser VPC gehörenden Subnetze hinzufügen).



8. Wenn Sie fertig sind, wählen Sie Erstellen aus.

   Ihre neue DB-Subnetzgruppe wird in der Liste der Subnetzgruppen in der Amazon RDS-Konsole angezeigt. Sie können sie wählen, um die Details im Details-Bereich unten auf der Seite anzeigen zu lassen – einschließlich aller Subnetze, die dieser Gruppe angehören.

Bereitstellen in Elastic Beanstalk

Nach dem Einrichten Ihrer VPC können Sie Ihre Umgebung darin erstellen und Ihre Anwendung für Elastic Beanstalk bereitstellen. Sie können dazu die Elastic Beanstalk Beanstalk-Konsole oder die AWS Toolkits, die EB CLI oder die Elastic Beanstalk Beanstalk-API verwenden. AWS CLI Wenn Sie die Elastic Beanstalk-Konsole verwenden, laden Sie einfach Ihre `.war`- oder `.zip`-Datei hoch und wählen Sie die VPC-Einstellung im Assistenten aus. Elastic Beanstalk erstellt Ihre Umgebung in Ihrer VPC und stellt Ihre Anwendung bereit. Alternativ können Sie die AWS Toolkits AWS CLI, die EB CLI oder die Elastic Beanstalk API verwenden, um Ihre Anwendung bereitzustellen. Dazu müssen

Sie Ihre VPC-Optionseinstellungen in einer Konfigurationsdatei definieren und diese Datei mit Ihrem Quell-Bundle bereitstellen. Dieses Thema enthält Anleitungen für beide Methoden.

Bereitstellen mit der Elastic Beanstalk-Konsole

Die Elastic Beanstalk-Konsole führt Sie durch die Schritte zum Erstellen Ihrer neuen Umgebung in der VPC. Sie müssen eine `.war`-Datei (für Java-Anwendungen) bzw. eine `.zip`-Datei (für alle anderen Anwendungen) angeben. Nehmen Sie auf der Seite VPC Configuration (VPC-Konfiguration) des Elastic Beanstalk-Umgebungsassistenten folgende Einstellungen vor:

VPC

Wählen Sie Ihre VPC aus.

VPC Security Group (VPC-Sicherheitsgruppe)

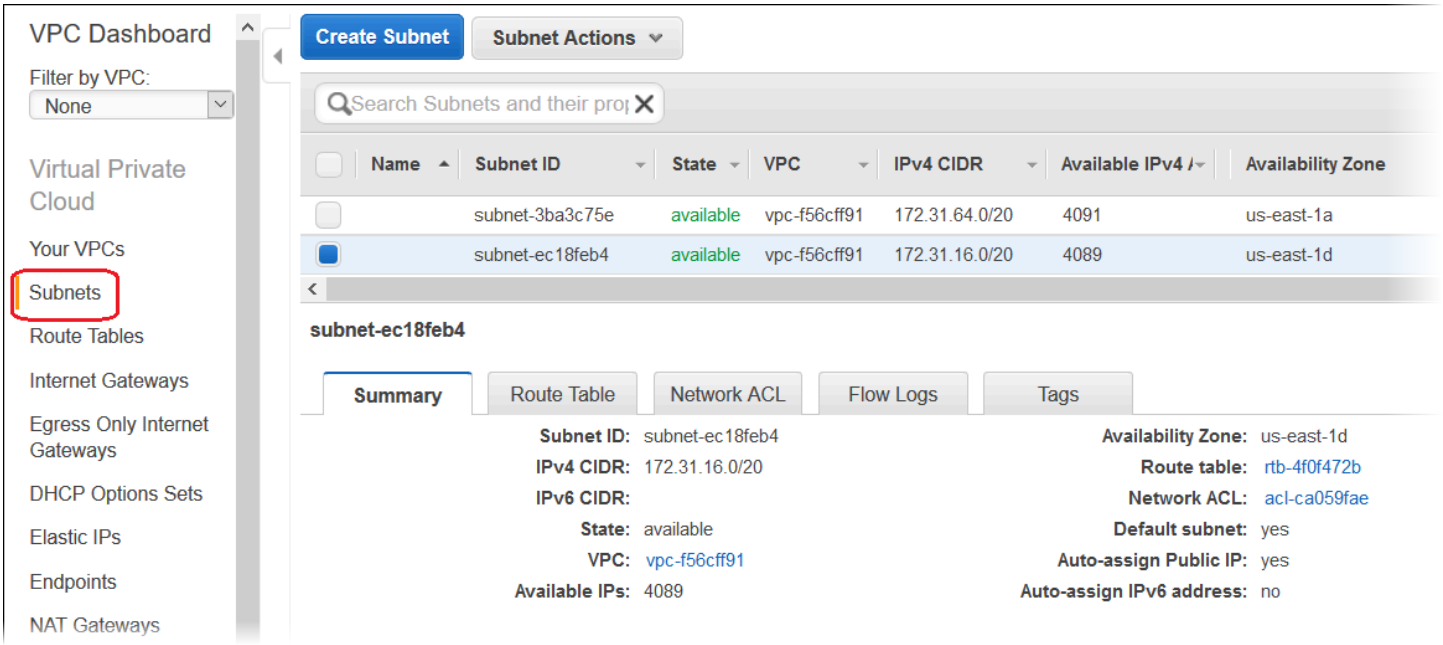
Wählen Sie die weiter oben erstellte Instance-Sicherheitsgruppe aus.

ELB visibility (ELB-Sichtbarkeit)

Wählen Sie `External` aus, wenn Ihr Load Balancer öffentlich verfügbar sein soll, oder `Internal`, wenn der Load Balancer nur in Ihrer VPC zur Verfügung stehen soll.

Wählen Sie die Subnetze für Ihren Load Balancer und Ihre Instances aus. EC2 Stellen Sie sicher, dass Sie das öffentliche Subnetz für den Load Balancer und das private Subnetz für Ihre Amazon-Instances auswählen. EC2 Standardmäßig erstellt der VPC-Assistent das öffentliche Subnetz in `10.0.0.0/24` und das private Subnetz in `10.0.1.0/24`.

Sie können Ihr Subnetz anzeigen, IDs indem Sie in der Amazon VPC-Konsole Subnetze auswählen.



Bereitstellung mit den AWS Toolkits, EB CLI oder API AWS CLI

Wenn Sie Ihre Anwendung mithilfe der AWS Toolkits, der EB-CLI oder der API auf Elastic Beanstalk bereitstellen AWS CLI, können Sie Ihre VPC-Optionseinstellungen in einer Datei angeben und sie zusammen mit Ihrem Quellpaket bereitstellen. Weitere Informationen finden Sie unter Erweiterte Umgebungsanpassung mit Konfigurationsdateien (`.ebextensions`).

Wenn Sie die Optionseinstellungen aktualisieren, müssen Sie mindestens Folgendes angeben:

- VPCId— Enthält die ID der VPC.
- Subnets (Subnetze) – Enthält die ID des Auto Scaling-Gruppensubnetzes. In diesem Beispiel ist dies die ID des privaten Subnetzes.
- ELBSubnets— Enthält die ID des Subnetzes für den Load Balancer. In diesem Beispiel ist dies die ID des öffentlichen Subnetzes.
- SecurityGroups— Enthält die ID der Sicherheitsgruppen.
- DBSubnets— Enthält die ID der DB-Subnetze.

> **Note**
>
> Wenn Sie DB-Subnetze verwenden, müssen Sie zusätzliche Subnetze in Ihrer VPC erstellen, um alle Availability Zones in der Region abzudecken. AWS

Optional können Sie auch die folgenden Informationen angeben:

- ELBScheme— Geben Sie `internal` an, dass ein interner Load Balancer in Ihrer VPC eingerichtet werden soll, sodass auf Ihre Elastic Beanstalk Beanstalk-Anwendung nicht von außerhalb Ihrer VPC zugegriffen werden kann.

Nachfolgend finden Sie ein Beispiel für die Optionseinstellungen, die Sie bei der Elastic Beanstalk-Anwendungsbereitstellung in einer VPC verwenden können. Weitere Informationen zu VPC-Optionseinstellungen (einschließlich Beispiele, wie diese angegeben werden, Standardwerte und gültige Werte) finden Sie in der aws:ec2:vpc-Namespace-Tabelle in Konfigurationsoptionen.

```
option_settings:
  - namespace: aws:autoscaling:launchconfiguration
    option_name: EC2KeyName
    value: ec2keypair

  - namespace: aws:ec2:vpc
    option_name: VPCId
    value: vpc-170647c

  - namespace: aws:ec2:vpc
    option_name: Subnets
    value: subnet-4f195024

  - namespace: aws:ec2:vpc
    option_name: ELBSubnets
    value: subnet-fe064f95

  - namespace: aws:ec2:vpc
    option_name: DBSubnets
    value: subnet-fg148g78

  - namespace: aws:autoscaling:launchconfiguration
    option_name: InstanceType
    value: m1.small

  - namespace: aws:autoscaling:launchconfiguration
    option_name: SecurityGroups
    value: sg-7f1ef110
```

> **Note**
>
> Wenn Sie DB-Subnetze verwenden, stellen Sie sicher, dass Ihre VPC über Subnetze verfügt, um alle Availability Zones in der Region abzudecken. AWS

Verwenden von Elastic Beanstalk mit VPC-Endpunkten

In diesem Thema werden die Vorteile erklärt, die ein VPC-Endpoint Ihrer Elastic Beanstalk Beanstalk-Anwendung bieten kann. Es enthält auch Anweisungen zum Erstellen eines VPC-Schnittstellen-Endpunkts für einen Elastic Beanstalk-Service.

Mit einem VPC-Endpunkt können Sie Ihre VPC privat mit unterstützten AWS Diensten und VPC-Endpunktdiensten verbinden, die von bereitgestellt werden AWS PrivateLink, ohne dass ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine Verbindung erforderlich ist. Direct Connect

Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit Ressourcen im Service zu kommunizieren. Der Datenverkehr zwischen Ihrer VPC und dem anderen Service verlässt das Amazon-Netzwerk nicht. Vollständige Informationen zu VPC-Endpunkten finden Sie unter VPC-Endpunkte im Amazon VPC-Benutzerhandbuch.

AWS Elastic Beanstalk unterstützt AWS PrivateLink, was private Konnektivität zum Elastic Beanstalk-Service bietet und verhindert, dass Datenverkehr dem öffentlichen Internet ausgesetzt wird. Damit Ihre Anwendung Anfragen an Elastic Beanstalk senden kann, konfigurieren Sie einen VPC-Endpunkttyp AWS PrivateLink, der als Schnittstellen-VPC-Endpunkt (Schnittstellenendpunkt) bezeichnet wird. Weitere Informationen finden Sie unter Schnittstellen-VPC-Endpunkte (AWS PrivateLink) im Amazon-VPC-Benutzerhandbuch.

> **Note**
>
> Elastic Beanstalk unterstützt AWS PrivateLink und VPC VPC-Endpoints in einer begrenzten Anzahl von. AWS-Regionen Wir arbeiten daran, den Support AWS-Regionen in naher future auf weitere auszudehnen.

IPv6 Unterstützung mit Dual-Stack-Konfiguration

Elastic Beanstalk unterstützt eingehenden Traffic über IPv4 und. IPv6 In diesem Abschnitt werden die öffentlichen Endpoints beschrieben, die beide unterstützen, IPv4 IPv6 und es wird auch erklärt, wie Sie Ihre Elastic Beanstalk VPC-Endpoints so konfigurieren, dass sie Dual-Stack-Traffic unterstützen.

Öffentliche Endpunkte

Der Elastic Beanstalk-Service besteht aus zwei Gruppen von Endpunkten, die aus den älteren Endpunkten und den neueren IPv4 Endpunkten mit Dual-Stack-Fähigkeit bestehen. Beide Gruppen von Endpunkten folgen den Benennungsstandards: AWS

- IPv4 Endpunkte verwenden das Domänenformat `amazonaws.com` für allgemeine Dienstendpunkte: `elasticbeanstalk.region.amazonaws.com`
- Dual-Stack-Endpunkte verwenden das Domain-Format für den allgemeinen Service-Endpunkt`api.aws`: `elasticbeanstalk.region.api.aws`

Die Endpunkte für Service Health und FIPS haben unterschiedliche Hostnamen, folgen aber demselben Domainnamenmuster. Eine Liste der Endpunkte finden Sie unter Elastic Beanstalk Service Endpoints in der. Allgemeine Amazon Web Services-Referenz

Anfragen an Elastic Beanstalk

Wenn Sie Anfragen mit dem oder dem AWS SDK an den Elastic Beanstalk-Service senden, können Sie einen IPv4 Endpunkt AWS CLIoder einen Dual-Stack-Endpunkt angeben. Das AWS SDK AWS CLI und das SDK verwenden standardmäßig die Endpunkte „ IPv4-only“, wenn keine Endpunkt-URL angegeben ist.

Das folgende Beispiel zeigt das AWS CLI Senden einer Anfrage an einen Dual-Stack-Endpunkt:

Example

```
aws elasticbeanstalk list-available-solution-stacks \
    --endpoint-url "https://elasticbeanstalk.us-east-1.api.aws"
```

Das folgende Beispiel zeigt, wie das AWS Python-SDK eine Anfrage an einen Dual-Stack-Endpunkt sendet:

Example

```
import boto3
```

```
dual_stack_eb_client = boto3.client(
    service_name='elasticbeanstalk',
    region_name='us-east-1',
    endpoint_url='https://elasticbeanstalk.us-east-1.api.aws';
)

print(dual_stack_eb_client.list_available_solution_stacks())
```

VPC-Endpunkte für Dual-Stack IPs

Um Ihre Elastic Beanstalk VPC-Endpoints für die Unterstützung von Dual-Stack-Verkehr zu konfigurieren, geben Sie Dualstack für den IP-Adresstypparameter des VPC-Endpoints an. Sie können dieses Feld über das, das SDK oder die AWS CLIKonsole angeben.AWS AWS PrivateLink Anweisungen dazu in der AWS PrivateLink Konsole finden Sie im AWS PrivateLink Handbuch unter Erstellen eines VPC-Endpunkts.

> **Note**
>
> Sie müssen den IP-Adresstyp des VPC-Endpunkts entweder als Dual-Stack IPv4oder als Dual-Stack angeben. Derzeit unterstützen Elastic Beanstalk VPC-Endpoints keinen IP-Adresstyp von IPv6, was auf eine reine Unterstützung hindeuten würde. IPv6 Die Dual-Stack-Option ermöglicht sowohl die als auch die Internetprotokolle. IPv4 IPv6

Das folgende Beispiel zeigt, wie Sie einen Dual-Stack-VPC-Endpunkt erstellen mit: AWS CLI

Example

```
aws ec2 create-vpc-endpoint \
    --vpc-id "vpc-example"
    --service-name "com.amazonaws.us-east-1.elasticbeanstalk"
    --ip-address-type "dualstack"
```

Einrichten eines VPC-Endpunkts für Elastic Beanstalk

Zur Erstellung des Schnittstellen-VPC-Endpunkts für den Elastic Beanstalk-Service in Ihrer VPC folgen Sie der Prozedur Erstellung eines Schnittstellenendpunkts.

- Wählen Sie für Service Name die Option com.amazonaws aus. ***region***.elasticbeanstalk.

- Wählen Sie als IP-Adresstyp entweder oder Dualstack. IPv4 Derzeit unterstützen Elastic Beanstalk VPC-Endpoints keinen IP-Adresstyp von IPv6, was auf eine reine Unterstützung hindeuten würde. IPv6 Die Dualstack-Option ermöglicht sowohl die als auch die Internetprotokolle. IPv4 IPv6

Wenn Ihre VPC mit öffentlichem Internetzugang konfiguriert ist, kann Ihre Anwendung weiterhin entweder über den `elasticbeanstalk.region.amazonaws.com` oder den öffentlichen Endpunkt über das Internet auf Elastic Beanstalk zugreifen. `elasticbeanstalk.region.api.aws` Sie können dies durch die Aktivierung von Enable DNS name (DNS-Namen aktivieren) während der Endpunkterstellung verhindern (standardmäßig „true“). Hierdurch wird ein DNS-Eintrag in Ihrer VPC hinzugefügt, der den Endpunkt für den öffentlichen Service dem VPC-Schnittstellenendpunkt zuordnet.

Einrichtung eines VPC-Endpunkts zur Verbesserung der Integrität

Wenn Sie die erweiterte Zustandsberichterstattung für Ihre Umgebung aktiviert haben, können Sie die erweiterten Gesundheitsinformationen so konfigurieren, dass sie ebenfalls gesendet werden. AWS PrivateLink Erweiterte Integritätsinformationen werden vom Daemon `healthd`, einer Elastic Beanstalk-Komponente auf den Instances in Ihrer Umgebung, an einen separaten, erweiterten Elastic Beanstalk-Integritätsservice gesendet. Zur Erstellung des Schnittstellen-VPC-Endpunkts für diesen Service in Ihrer VPC folgen Sie der Prozedur Erstellung eines Schnittstellenendpunkts.

- Wählen Sie für Service Name die Option com.amazonaws aus. ***region***.elasticbeanstalk-health.
- Wählen Sie als IP-Adresstyp entweder oder Dualstack. IPv4 Derzeit unterstützen Elastic Beanstalk VPC-Endpoints keinen IP-Adresstyp von IPv6, was auf eine reine Unterstützung hindeuten würde. IPv6 Die Dualstack-Option ermöglicht sowohl die als auch die Internetprotokolle. IPv4 IPv6

> **Important**
>
> Der `healthd` Daemon sendet erweiterte Gesundheitsinformationen an den öffentlichen Endpunkt oder. `elasticbeanstalk-health.region.amazonaws.com` `elasticbeanstalk-health.region.api.aws` Wenn Ihre VPC mit einem Zugang zum öffentlichem Internet konfiguriert und Enable DNS name (DNS-Name aktivieren) für den VPC-Endpunkt deaktiviert ist, werden erweiterte Integritätsdaten über das öffentliche Internet transportiert. Dies wird von Ihnen bei der Einrichtung eines VPC-Endpunkts für erweiterte Integrität wahrscheinlich nicht gewünscht. Stellen Sie daher sicher, dass Enable DNS name (DNS-Name aktivieren) aktiviert ist (standardmäßig „true“).

Verwenden von VPC-Endpunkten in einer privaten VPC

Eine private VPC (oder ein privates Subnetz in einer VPC) verfügt nicht über einen Zugang zum öffentlichen Internet. Möglicherweise möchten Sie Ihre Elastic Beanstalk-Umgebung in einer privaten VPC ausführen und Schnittstellen-VPC-Endpunkte konfigurieren, um die Sicherheit zu verbessern. In diesem Fall sollten Sie beachten, dass Ihre Umgebung möglicherweise versucht, aus anderen Gründen eine Verbindung zum Internet herzustellen, zusätzlich zur Kontaktaufnahme mit dem Elastic Beanstalk-Service. Weitere Informationen zur Ausführung einer Umgebung in einer privaten VPC finden Sie unter the section called “Ausführung einer Elastic Beanstalk-Umgebung in einer privaten VPC”.

Verwenden von Endpunktrichtlinien zur Steuerung des Zugriffs mit VPC-Endpunkten

In diesem Thema wird erklärt, wie Sie VPC-Endpoints eine Richtlinie hinzufügen können, um den Zugriff auf Ihre Anwendung (Ihren Service) und Ihre Elastic Beanstalk Beanstalk-Umgebung zu kontrollieren.

Eine Endpoint-Policy ist eine AWS Identity and Access Management (IAM) -Ressourcenrichtlinie, die den Zugriff vom Endpunkt auf den angegebenen Service steuert. Die Endpunktrichtlinie ist für den jeweiligen Endpunkt spezifisch. Sie ist von allen Benutzer- oder Instance-IAM-Richtlinien, die in Ihrer Umgebung möglicherweise vorhanden sind, getrennt; weder überschreibt sie diese noch ersetzt sie diese.

Standardmäßig ermöglicht ein VPC-Endpunkt den vollständigen Zugriff auf den Service, dem er zugeordnet ist. Wenn Sie einen Endpunkt erstellen oder ändern, können Sie ihm eine Endpunktrichtlinie hinzufügen, um den Zugriff auf bestimmte Ressourcen zu steuern, die dem Dienst zugeordnet sind. Einzelheiten zur Erstellung und Verwendung von VPC-Endpunktrichtlinien finden Sie im Handbuch unter Steuern des Zugriffs auf VPC-Endpoints mithilfe von Endpunktrichtlinien.AWS PrivateLink

> **Note**
>
> Wenn Sie restriktive Endpunktrichtlinien erstellen, müssen Sie den erforderlichen Ressourcen möglicherweise bestimmte Berechtigungen hinzufügen, damit der Zugriff auf diese Ressourcen nicht durch die Endpunktrichtlinie blockiert wird. Dadurch wird sichergestellt, dass Ihre Umgebung weiterhin ordnungsgemäß bereitgestellt wird und ordnungsgemäß funktioniert.

Das folgende Beispiel lehnt die Berechtigung zum Beenden einer Umgebung über den VPC-Endpunkt für alle Benutzer ab und ermöglicht den vollständigen Zugriff auf alle übrigen Aktionen.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "elasticbeanstalk:TerminateEnvironment",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

Erforderliche Amazon S3 S3-Bucket-Berechtigungen für restriktive VPC-Endpunktrichtlinien

Wenn Sie Ihren VPC-Endpunktrichtlinien Einschränkungen hinzufügen, müssen Sie spezifische Amazon S3 S3-Bucket-Berechtigungen angeben, um sicherzustellen, dass Ihre Umgebung weiterhin bereitgestellt wird und ordnungsgemäß funktioniert. Dieser Abschnitt erklärt die erforderlichen S3-Buckets und enthält Beispielrichtlinien.

Themen

- S3-Buckets, in denen Ressourcen zur Verwaltung von Umgebungsplattformen gespeichert werden
- S3-Buckets gehören CloudFormation
- S3-Buckets, die Kundenkonten gehören, zum Speichern von Quellcode und anderen Elementen
- S3-Buckets, die Kundenkonten gehören, zur Unterstützung der Docker-Registry-Authentifizierung
- Aktualisierung Ihrer VPC-Endpunktrichtlinie

S3-Buckets, in denen Ressourcen zur Verwaltung von Umgebungsplattformen gespeichert werden

Der Elastic Beanstalk-Service besitzt S3-Buckets, in denen die mit einem Lösungsstapel (Plattformversion) verknüpften Assets gespeichert werden. Zu diesen Ressourcen gehören

Konfigurationsdateien, die Beispielanwendung und verfügbare Instance-Typen. Wenn Elastic Beanstalk Ihre Umgebung erstellt und verwaltet, ruft es die erforderlichen Informationen für die jeweilige Plattformversion aus dem Asset-Bucket für jede entsprechende Umgebung ab. AWS-Region

S3-Bucket ARN

`arn:aws:s3:::elasticbeanstalk-samples-region`

Amazon-Linux-2 und höher

- `arn:aws:s3:::elasticbeanstalk-platform-assets-region`

> **Note**
>
> Der Bucket-Name folgt einer anderen Konvention für die BJS-Region. Die Zeichenfolge public-beta-cn-north-1 wird anstelle von *region* verwendet. Beispiel, `arn:aws:s3:::elasticbeanstalk-platform-assets-public-beta-cn-north-1`.

Windows Server, Amazon Linux (AMI), Amazon Linux 2 und höher

- `arn:aws:s3:::elasticbeanstalk-env-resources-region`
- `arn:aws:s3:::elasticbeanstalk-region`

> **Note**
>
> Die Bucket-Namen für Platform-Assets- und Env-Resources-Buckets folgen in einigen Regionen unterschiedlichen Konventionen. Einzelheiten finden Sie weiter unten im Abschnitt zu den regionsspezifischen Bucket-Namensmustern.

Regionsspezifische Bucket-ARN-Muster

Region	Bucket-Typ	Bucket-ARN-Muster
me-central-1	Plattform-Assets	`arn:aws:s3:::elasticbeanstalk-platform-assets-me-central-1-f08b818c`

Region	Bucket-Typ	Bucket-ARN-Muster
	Umgebungsressourcen	`arn:aws:s3:::elasticbeanstalk-env-resources-me-central-1-f08b818c`

Operationen

GetObject

Beispiel für eine VPC-Endpunktrichtlinie

Das folgende Beispiel zeigt, wie Sie Zugriff auf die S3-Buckets gewähren, die für Elastic Beanstalk Beanstalk-Operationen in der Region USA Ost (Ohio) (us-east-2) erforderlich sind. Das Beispiel listet alle Buckets sowohl für Amazon Linux- als auch für Windows Server-Plattformen auf. Aktualisieren Sie Ihre Richtlinie so, dass sie nur die Buckets enthält, die für das Betriebssystem Ihrer Umgebung gelten.

Beispielrichtline

> **Important**
>
> Wir empfehlen, dass Sie keine Platzhalterzeichen (*) für bestimmte Regionen in dieser Richtlinie verwenden. Verwenden Sie beispielsweise `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` und nicht `arn:aws:s3:::cloudformation-waitcondition-*/*`. Bei der Verwendung von Platzhaltern könnte Zugriff auf S3-Buckets erteilt werden, für die Sie keinen Zugriff gewähren möchten. Wenn Sie die Richtlinie für mehr als eine Region verwenden möchten, empfehlen wir, den ersten `Statement` Block für jede Region zu wiederholen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToAWSResources",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
```

```
            "Action": ["s3:GetObject"],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-platform-assets-us-east-2/*",
                "arn:aws:s3:::elasticbeanstalk-env-resources-us-east-2/*",
                "arn:aws:s3:::elasticbeanstalk-env-resources-us-east-2/*",
                "arn:aws:s3:::elasticbeanstalk-samples-us-east-2/*"
            ]
        }
    ]
}
```

S3-Buckets gehören CloudFormation

Elastic Beanstalk verwendet CloudFormation , um Ressourcen für Ihre Umgebung zu erstellen. CloudFormation besitzt jeweils S3-Buckets, um die Reaktionen AWS-Region auf Wartebedingungen zu überwachen.

Dienste wie Elastic Beanstalk kommunizieren mit, CloudFormation indem sie Anfragen an eine vorsignierte Amazon S3 S3-URL für den S3-Bucket senden, der Eigentümer ist. CloudFormation CloudFormation erstellt die vorsignierte Amazon S3 S3-URL mithilfe des `cloudformation.amazonaws.com` Service Principal.

Ausführlichere Informationen finden Sie unter Überlegungen zu CloudFormation VPC-Endpunkten im AWS CloudFormation Benutzerhandbuch. Weitere Informationen zu presigned URLs finden Sie unter Working with presigned URLs im Amazon S3 S3-Benutzerhandbuch.

S3-Bucket ARN

- `arn:aws:s3:::cloudformation-waitcondition-region`

  Wenn Sie Wartebedingungen verwenden, enthalten Regionennamen Bindestriche. Zum Beispiel us-west-2.
- `arn:aws:s3:::cloudformation-custom-resource-response-region`

  Wenn Sie benutzerdefinierte Ressourcen verwenden, enthalten Regionennamen keine Bindestriche. Zum Beispiel uswest2.

Operationen

GetObject

Beispiel für eine VPC-Endpunktrichtlinie

Das folgende Beispiel zeigt, wie Sie Zugriff auf die S3-Buckets gewähren, die für Elastic Beanstalk Beanstalk-Operationen in der Region USA Ost (Ohio) (us-east-2) erforderlich sind.

Beispielrichtline

> **Important**
>
> Wir empfehlen, dass Sie keine Platzhalterzeichen (*) für bestimmte Regionen in dieser Richtlinie verwenden. Verwenden Sie beispielsweise `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` und nicht `arn:aws:s3:::cloudformation-waitcondition-*/*`. Bei der Verwendung von Platzhaltern könnte Zugriff auf S3-Buckets erteilt werden, für die Sie keinen Zugriff gewähren möchten. Wenn Sie die Richtlinie für mehr als eine Region verwenden möchten, empfehlen wir, den ersten `Statement` Block für jede Region zu wiederholen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToCloudFormation",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": ["s3:GetObject"],
            "Resource": [
                "arn:aws:s3:::cloudformation-waitcondition-us-east-2/*",
                "arn:aws:s3:::cloudformation-custom-resource-response-us-east-2/
*"
            ]
         }
    ]
}
```

S3-Buckets, die Kundenkonten gehören, zum Speichern von Quellcode und anderen Elementen

Dieser Bucket gehört dem AWS Kundenkonto, dem die Umgebung gehört. Es speichert Ressourcen, die für Ihre Umgebung spezifisch sind, z. B. Quellcode und angeforderte Protokolle.

S3-Bucket ARN

`arn:aws:s3:::elasticbeanstalk-region-account-id`

Operationen

- GetObject
- GetObjectAcl
- PutObject
- PutObjectAcl
- ListBucket

Beispiel für eine VPC-Endpunktrichtlinie

Das folgende Beispiel zeigt, wie Sie Zugriff auf die S3-Buckets gewähren, die für Elastic Beanstalk Beanstalk-Operationen in der Region USA Ost (Ohio) (us-east-2) und für das Beispiel die Konto-ID 123456789012 erforderlich sind. AWS

Beispielrichtline

> **Important**
>
> Wir empfehlen, dass Sie keine Platzhalterzeichen (*) für bestimmte Regionen in dieser Richtlinie verwenden. Verwenden Sie beispielsweise `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` und nicht `arn:aws:s3:::cloudformation-waitcondition-*/*`. Bei der Verwendung von Platzhaltern könnte Zugriff auf S3-Buckets erteilt werden, für die Sie keinen Zugriff gewähren möchten. Wenn Sie die Richtlinie für mehr als eine Region verwenden möchten, empfehlen wir, den ersten Block für jede Region zu wiederholen. `Statement`

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToCustomerItems",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": [
               "s3:GetObject",
               "s3:GetObjectAcl",
               "s3:PutObject",
               "s3:PutObjectAcl",
               "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/*"
            ]
         }
    ]
}
```

S3-Buckets, die Kundenkonten gehören, zur Unterstützung der Docker-Registry-Authentifizierung

Dieser Bucket gilt nur für Umgebungen, die auf der Docker-Plattform basieren. Der Bucket speichert eine Datei, die zur Authentifizierung bei einer privaten Docker-Registry verwendet wird, die sich in einem vom Kunden bereitgestellten S3-Bucket befindet. Weitere Informationen finden Sie unter Verwenden der `Dockerrun.aws.json` v3-Datei im Kapitel Docker-Plattform dieses Handbuchs.

S3-Bucket ARN

Die ARN variiert je nach Kundenkonto.

Der S3-Bucket-ARN hat das folgende Format: `arn:aws:s3:::`*`bucket-name`*

Operationen

GetObject

Beispiel für eine VPC-Endpunktrichtlinie

Das folgende Beispiel zeigt, wie Sie Zugriff auf einen S3-Bucket mit dem Namen amzn-s3-demo-bucket1 gewähren.

Beispielrichtline

JSON

```
{
"Version":"2012-10-17",
"Statement": [
   {
      "Sid": "AllowRequestsToDockerRegistryAuth",
       "Effect": "Allow",
       "Action": [
          "s3:GetObject"
        ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket1"
    ]
  }
]
}
```

Aktualisierung Ihrer VPC-Endpunktrichtlinie

Da an einen VPC-Endpunkt nur eine Richtlinie angehängt ist, müssen Sie alle Berechtigungen in einer Richtlinie zusammenfassen. Im folgenden Beispiel werden alle vorherigen Beispiele zu einem zusammengefasst.

Einzelheiten zur Erstellung und Verwendung von VPC-Endpunktrichtlinien finden Sie im Handbuch unter Steuern des Zugriffs auf VPC-Endpoints mithilfe von Endpunktrichtlinien.AWS PrivateLink

Wie die vorherigen Beispiele zeigt auch das folgende Beispiel, wie Sie Zugriff auf die S3-Buckets bereitstellen, die für Elastic Beanstalk Beanstalk-Operationen in der Region USA Ost (Ohio) (us-east-2) erforderlich sind. Es enthält auch Buckets mit der AWS Beispielkonto-ID 123456789012 und dem Beispiel-Bucket-Namen amzn-s3-demo-bucket1.

> **Important**
>
> Wir empfehlen, dass Sie keine Platzhalterzeichen (*) für bestimmte Regionen in dieser Richtlinie verwenden. Verwenden Sie beispielsweise `arn:aws:s3:::cloudformation-`

`waitcondition-us-east-2/*` und nicht `arn:aws:s3:::cloudformation-waitcondition-*/*`. Bei der Verwendung von Platzhaltern könnte Zugriff auf S3-Buckets erteilt werden, für die Sie keinen Zugriff gewähren möchten. Wenn Sie die Richtlinie für mehr als eine Region verwenden möchten, empfehlen wir, den ersten Block für jede Region zu wiederholen. `Statement`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToAWSResources",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-platform-assets-us-east-2/*",
                "arn:aws:s3:::elasticbeanstalk-env-resources-us-east-2/*",
                "arn:aws:s3:::elasticbeanstalk-samples-us-east-2/*"
            ]
        },
        {
            "Sid": "AllowRequestsToCloudFormation",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::cloudformation-waitcondition-us-east-2/*",
                "arn:aws:s3:::cloudformation-custom-resource-response-us-east-2/
*"
            ]
        },
        {
            "Sid": "AllowRequestsToCustomerItems",
            "Effect": "Allow",
            "Action": [
               "s3:GetObject",
```

```
                "s3:GetObjectAcl",
               "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/*"
            ]
        },
        {
            "Sid": "AllowRequestsToDockerRegistryAuth",
            "Effect": "Allow",
            "Action": [
            "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
        }
    ]
}
```

AWS Elastic Beanstalk Sicherheit

In diesem Kapitel erfahren Sie mehr über die Sicherheitsaufgaben, für die Elastic Beanstalk verantwortlich ist, sowie über die Sicherheitskonfigurationen, die Sie bei der Verwendung von Elastic Beanstalk berücksichtigen sollten, um Ihre Sicherheits- und Compliance-Ziele zu erreichen.

Cloud-Sicherheit hat höchste AWS Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Im Modell der übergreifenden Verantwortlichkeit wird dies als Sicherheit der Cloud und Sicherheit in der Cloud bezeichnet.

Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der alle in der AWS Cloud angebotenen Dienste ausgeführt werden, und für die Bereitstellung von Diensten, die Sie sicher nutzen können. Unsere Sicherheitsverantwortung hat bei uns höchste Priorität AWS, und die Wirksamkeit unserer Sicherheit wird im Rahmen der AWS Compliance-Programme regelmäßig von externen Prüfern getestet und verifiziert. Informationen zu Elastic Beanstalk finden Sie in den Programmen „AWS Services in Scope of AWS Assurance“.

Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen, und anderen Faktoren wie der Sensibilität Ihrer Daten, den Anforderungen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften. Diese Dokumentation beschreibt, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Elastic Beanstalk anwenden können.

Topics

- Datenschutz in Elastic Beanstalk
- Identity and Access Management für Elastic Beanstalk
- Protokollierung und Überwachung in Elastic Beanstalk
- Compliance-Validierung für Elastic Beanstalk
- Resilienz in Elastic Beanstalk
- Infrastruktursicherheit in Elastic Beanstalk
- Konfigurations- und Schwachstellenanalyse in Elastic Beanstalk
- Bewährte Sicherheitsmethoden für Elastic Beanstalk

Datenschutz in Elastic Beanstalk

Das Modell der AWS gemeinsamen Verantwortung und geteilter Verantwortung gilt für den Datenschutz in AWS Elastic Beanstalk. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dazu gehört auch, wenn Sie mit Elastic Beanstalk oder anderen AWS-Services über die Konsole AWS CLI, API oder arbeiten. AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen

externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

Themen

- Datenschutz durch Verschlüsselung
- Richtlinie für den Datenverkehr zwischen Netzwerken

Datenschutz durch Verschlüsselung

Sie können verschiedene Formen der Datenverschlüsselung verwenden, um Ihre Elastic Beanstalk Beanstalk-Daten zu schützen. Datenschutz bezieht sich auf den Schutz von Daten während der Übertragung (auf dem Weg zu und von Elastic Beanstalk) und im Ruhezustand (während sie in AWS Rechenzentren gespeichert werden).

Verschlüsselung während der Übertragung

Sie können den Datenschutz während der Übertragung auf zwei Arten erreichen: Verschlüsseln Sie die Verbindung mit Secure Sockets Layer (SSL) oder verwenden Sie die clientseitige Verschlüsselung (wobei das Objekt vor dem Senden verschlüsselt wird). Beide Methoden sind für den Schutz Ihrer Anwendungsdaten gültig. Um die Verbindung zu sichern, verschlüsseln Sie sie mit SSL, wenn Ihre Anwendung, ihre Entwickler und Administratoren und ihre Endbenutzer Objekte senden oder empfangen. Weitere Informationen zum Verschlüsseln von Web-Datenverkehr zu und von Ihrer Anwendung finden Sie unter the section called “HTTPS”.

Die clientseitige Verschlüsselung ist keine gültige Methode zum Schutz Ihres Quellcodes in Anwendungsversionen und Quell-Bundles, die Sie hochladen. Elastic Beanstalk benötigt Zugriff auf diese Objekte, sodass sie nicht verschlüsselt werden können. Stellen Sie daher sicher, dass Sie die Verbindung zwischen Ihrer Entwicklungs- oder Bereitstellungsumgebung und Elastic Beanstalk sichern.

Verschlüsselung im Ruhezustand

Um die Daten Ihrer Anwendung im Ruhezustand zu schützen, erfahren Sie mehr über den Datenschutz im Speicherservice, den Ihre Anwendung verwendet. Beispiele finden Sie unter Datenschutz in Amazon RDS im Amazon-RDS-Benutzerhandbuch, Datenschutz in Amazon S3 im Entwicklerhandbuch für Amazon Simple Storage Service oder Verschlüsseln von Daten und Metadaten in EFS im Benutzerhandbuch für Amazon Elastic File System.

Elastic Beanstalk speichert verschiedene Objekte in einem verschlüsselten Amazon Simple Storage Service (Amazon S3) -Bucket, den es für jede AWS Region erstellt, in der Sie Umgebungen erstellen. Da Elastic Beanstalk die von Amazon S3 bereitgestellte Standardverschlüsselung beibehält, erstellt es verschlüsselte Amazon S3 S3-Buckets. Details hierzu finden Sie unter the section called “Amazon S3”. Sie stellen einige der gespeicherten Objekte bereit und senden sie an Elastic Beanstalk, z. B. Anwendungsversionen und Quell-Bundles. Elastic Beanstalk generiert andere Objekte, z. B. Protokolldateien. Zusätzlich zu den Daten, die Elastic Beanstalk speichert, kann Ihre Anwendung and/or im Rahmen ihres Betriebs auch Speicherdaten übertragen.

Um Daten zu schützen, die auf Amazon Elastic Block Store (Amazon EBS) -Volumes gespeichert sind, die mit den Instances Ihrer Umgebung verknüpft sind, aktivieren Sie die Amazon EBS-Verschlüsselung standardmäßig in Ihrem AWS Konto und Ihrer Region. Wenn diese Option aktiviert ist, werden alle neuen Amazon EBS-Volumes und ihre Snapshots automatisch mit Schlüsseln AWS Key Management Service verschlüsselt. Weitere Informationen finden Sie unter Standardverschlüsselung im Amazon EBS-Benutzerhandbuch.

Weitere Informationen zum Datenschutz enthält der Blog-Beitrag AWS Shared Responsibility Model and GDPR im AWS -Sicherheitsblog.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

Richtlinie für den Datenverkehr zwischen Netzwerken

Sie können Amazon Virtual Private Cloud (Amazon VPC) verwenden, um Grenzen zwischen Ressourcen in Ihrer Elastic Beanstalk-Anwendung zu erstellen und den Datenverkehr zwischen ihnen, Ihrem On-Premise-Netzwerk und dem Internet zu steuern. Details hierzu finden Sie unter the section called “Amazon VPC”.

Weitere Informationen zur Amazon VPC-Sicherheit finden Sie unter Sicherheit im Amazon VPC-Benutzerhandbuch.

Weitere Informationen zum Datenschutz enthält der Blog-Beitrag AWS Shared Responsibility Model and GDPR im AWS -Sicherheitsblog.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

Identity and Access Management für Elastic Beanstalk

AWS Identity and Access Management (IAM) ist ein AWS Dienst, der einem Administrator hilft, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren steuern, wer authentifiziert

(angemeldet) und autorisiert (Berechtigungen besitzt) ist, um AWS Elastic Beanstalk Ressourcen zu nutzen. IAM ist ein AWS Dienst, den Sie ohne zusätzliche Kosten nutzen können.

Weitere Informationen zum Arbeiten mit IAM finden Sie unter Verwenden von Elastic Beanstalk mit AWS Identity and Access Management.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

AWS verwaltete Richtlinien für AWS Elastic Beanstalk

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird. AWS AWS Verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

Elastic Beanstalk Beanstalk-Updates für verwaltete Richtlinien AWS

Hier finden Sie Details zu Aktualisierungen der AWS verwalteten Richtlinien für Elastic Beanstalk seit dem 1. März 2021.

Die JSON-Quelle für eine bestimmte verwaltete Richtlinie finden Sie im Referenzhandbuch für AWS verwaltete Richtlinien.

Änderungen	Beschreibung	Date
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, damit Elastic Beanstalk verwaltete Updates durchführen kann, wenn sie aktiviert Tag-Propagierung zu Startvorlagen ist. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	17. März 2026
AWSElasticBeanstalkWebTier— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, damit Elastic Beanstalk Amazon Bedrock für KI-gestützte Umgebungs analysen verwenden kann. Weitere Informationen finden Sie unter Elastic Beanstalk Instance-Profile verwalten.	11. März 2026
AWSElasticBeanstalkWorkerTier— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, damit Elastic Beanstalk Amazon Bedrock für KI-gestützte Umgebungs analysen verwenden kann. Weitere Informationen finden Sie unter Elastic Beanstalk Instance-Profile verwalten.	11. März 2026
AWSElasticBeanstalkMulticontainerDocker— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, damit Elastic Beanstalk Amazon Bedrock für KI-gestützte Umgebungs analysen verwenden kann.	11. März 2026

Änderungen	Beschreibung	Date
	Weitere Informationen finden Sie unter Elastic Beanstalk Instance-Profile verwalten.	
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, sodass Elastic Beanstalk verwaltete Updates durchführen kann, wenn sie für eine einzelne Instanz aktiviert Tag-Propagierung zu Startvorlagen ist. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	27. Januar 2026
AdministratorAccess-AWSElastic Beanstalk — Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk die Konfiguration von Blockeinstellungen für öffentlichen Zugriff und Bucket-Besitzkontrollen für S3-Buckets zu ermöglichen. Weitere Informationen finden Sie unter Elastic Beanstalk-Benutzerrichtlinien verwalten.	12. November 2025

Änderungen	Beschreibung	Date
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, damit Elastic Beanstalk verwaltete Updates durchführen kann, wenn sie aktiviert Tag-Propagierung zu Startvorlagen ist. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	27. Februar 2025
AdministratorAccess-AWSElastic Beanstalk — Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um den Operator durch den StringLikeOperator zu ersetzen, der ArnLikedie Schlüssel vom Typ ARN im Bedingungsblock auswertet . `iam:PolicyArn` Dies ermöglicht eine sicherere Durchsetzung. Weitere Informationen finden Sie unter Elastic Beanstalk-Benutzerrichtlinien verwalten.	11. Dezember 2024

Änderungen	Beschreibung	Date
Die folgenden Richtlinien wurden aktualisiert: • AWSElasticBeanstalkInternalMaintenanceRolePolicy • AWSElasticBeanstalkMaintenance • AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy • AWSElasticBeanstalkManagedUpdatesServiceRolePolicy • AWSElasticBeanstalkRoleCore	Diese Richtlinien wurden aktualisiert, sodass Elastic Beanstalk Tags hinzufügen oder entfernen kann, wenn ein AWS CloudFormation Stack oder ein Änderungssatz erstellt oder aktualisiert wird. Mehr über `AWSElasticBeanstalkManagedUpdatesServiceRolePolicy` erfahren Sie unter Berechtigungen für servicegebundene Rollen für Elastic Beanstalk. Mehr über `AWSElasticBeanstalkRoleCore` erfahren Sie unter Richtlinien für die Integration mit anderen Services.	30. April 2024

Änderungen	Beschreibung	Date
AWSElasticBeanstalkService— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk zu erlauben, beim Erstellen von Elastic Load Balancing, Auto Scaling-Gruppen (ASG) und Amazon ECS Ressourcen zu taggen. **Note** Diese Richtlinie wurde zuvor durch `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` ersetzt. Diese Richtlinie steht zwar nicht mehr für die Anbindung an neue IAM-Benutzer, -Gruppen oder -Rollen zur Verfügung, dennoch kann sie an bereits vorhandene Benutzer, Gruppen oder Rollen angefügt werden. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	10. Mai 2023

Änderungen	Beschreibung	Date
AWSElasticBeanstalkMulticontainerDocker— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk zu erlauben, beim Erstellen Tags zu Amazon ECS hinzuzufügen. Weitere Informationen finden Sie unter Elastic Beanstalk Instance-Profile verwalten.	23. März 2023
AWSElasticBeanstalkRoleECS — Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk zu erlauben, beim Erstellen Tags zu Amazon ECS hinzuzufügen. Weitere Informationen finden Sie unter Richtlinien für die Integration mit anderen Services.	23. März 2023
AdministratorAccess-AWSElastic Beanstalk — Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk zu erlauben, beim Erstellen Tags zu Amazon ECS hinzuzufügen. Weitere Informationen finden Sie unter Elastic Beanstalk-Benutzerrichtlinien verwalten.	23. März 2023

Änderungen	Beschreibung	Date
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy — Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk zu ermöglichen, beim Erstellen Tags zu Amazon-ECS-Ressourcen hinzuzufügen. Weitere Informationen finden Sie unter Berechtigungen für servicegebundene Rollen für Elastic Beanstalk.	23. März 2023
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk zu ermöglichen, beim Erstellen Tags zu Amazon-ECS-Ressourcen hinzuzufügen. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	23. März 2023
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk das Hinzufügen von Tags zu Auto-Scaling-Gruppen zu ermöglichen, wenn es diese erstellt. Weitere Informationen finden Sie unter Serviceverknüpfte Rolle für verwaltete Updates.	27. Januar 2023

Änderungen	Beschreibung	Date
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk zu ermöglichen, beim Erstellen einer Auto Scaling-Gruppe (ASG) Tags hinzuzufügen. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	23. Januar 2023
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, um Elastic Beanstalk zu ermöglichen, beim Erstellen eines Elastic Load Balancers (ELB) Tags hinzuzufügen. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	21. Dezember 2022

Änderungen	Beschreibung	Date
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy— Bestehende Richtlinie wurde aktualisiert	Dieser Richtlinie wurden Berechtigungen hinzugefügt, um Elastic Beanstalk bei verwalteten Updates Folgendes zu ermöglichen: • Erstellen und Löschen von Startvorlagen und Vorlagenversionen. • Starten von Amazon-EC2-Instances mit Startvorlagen. • Wenn ein Amazon RDS vorhanden ist, rufen Sie eine Liste der verfügbaren DB-Engines und Informationen zu bereitgestellten RDS-Instances ab. Weitere Informationen finden Sie unter Serviceverknüpfte Rolle für verwaltete Updates.	23. August 2022

Änderungen	Beschreibung	Date
AWSElasticBeanstalkReadOnlyAccess – als veraltet gekennzeichnet GovCloud (USA) AWS-Region	Diese Richtlinie wurde ersetzt durch `AWSElasticBeanstalkReadOnly` . Diese Richtlinie wird in den GovCloud (USA) AWS-Region schrittweise eingestellt. Wenn diese Richtlinie schrittweise eingestellt wird, steht sie nach dem 17. Juni 2021 nicht mehr für die Anbindung an neue IAM-Benutzer, -Gruppen oder Rollen zur Verfügung. Weitere Informationen finden Sie unter Benutzerrichtlinien.	17. Juni 2021
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Bestehende Richtlinie wurde aktualisiert	Diese Richtlinie wurde aktualisiert, damit Elastic Beanstalk Attribute für EC2 Availability Zones lesen kann. Es ermöglicht Elastic Beanstalk, eine effektivere Validierung der Auswahl des Instance-Typs über Availability Zones hinweg bereitzustellen. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	16. Juni 2021

Änderungen	Beschreibung	Date
AWSElasticBeanstalkFullAccess – als veraltet gekennzeichnet GovCloud (USA) AWS-Region	Diese Richtlinie wurde ersetzt durch `AdministratorAccess-AWSElasticBeanstalk` . Diese Richtlinie wird in den GovCloud (USA) AWS-Region schrittweise eingestellt. Wenn diese Richtlinie schrittweise eingestellt wird, steht sie nach dem 10. Juni 2021 nicht mehr für die Anbindung an neue IAM-Benutzer, -Gruppen oder Rollen zur Verfügung. Weitere Informationen finden Sie unter Benutzerrichtlinien.	10. Juni 2021

Änderungen	Beschreibung	Date
Die folgenden verwalteten Richtlinien waren in allen Ländern China AWS-Region veraltet: • AWSElasticBeanstalkFullAccess • AWSElasticBeanstalkReadOnlyAccess	Die `AWSElasticBeanstalkFullAccess` -Richtlinie wurde durch `AdministratorAccess-AWSElasticBeanstalk` ersetzt. Die `AWSElasticBeanstalkReadOnlyAccess` -Richtlinie wurde durch `AWSElasticBeanstalkReadOnly` ersetzt. Diese Richtlinien wurden in allen Ländern China schrittweise eingestellt AWS-Region. Diese Richtlinien stehen nach dem 3. Juni 2021 nicht mehr für die Anbindung an neue IAM-Benutzer, -Gruppen oder Rollen zur Verfügung. Weitere Informationen finden Sie unter Benutzerrichtlinien.	3. Juni 2021

Änderungen	Beschreibung	Date
AWSElasticBeanstalkService – als veraltet gekennzeichnet	Diese Richtlinie wurde durch `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` ersetzt. Diese Richtlinie läuft aus und kann nicht mehr an neue IAM-Benutzer, -Gruppen oder -Rollen angefügt werden. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	Juni 2021 – Januar 2022

Änderungen	Beschreibung	Date
Die folgenden verwalteten Richtlinien waren in allen Ländern AWS-Region außer China und GovCloud (USA) veraltet: • AWSElasticBeanstalkFullAccess • AWSElasticBeanstalkReadOnlyAccess	Die `AWSElasticBeanstalkFullAccess` -Richtlinie wurde durch `AdministratorAccess-AWSElasticBeanstalk` ersetzt. Die `AWSElasticBeanstalkReadOnlyAccess` -Richtlinie wurde durch `AWSElasticBeanstalkReadOnly` ersetzt. Diese Richtlinien wurden in allen Ländern außer in China und GovCloud (den AWS-Region USA) schrittweise eingestellt. Diese Richtlinien stehen nach dem 16. April 2021 nicht mehr für die Anbindung an neue IAM-Benutzer, -Gruppen oder Rollen zur Verfügung. Weitere Informationen finden Sie unter Benutzerrichtlinien.	16. April 2021

Änderungen	Beschreibung	Date
Die folgenden verwalteten Richtlinien wurden aktualisiert: • AdministratorAccess-Bohnenstange AWSElastic • AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	Beide Richtlinien unterstützen jetzt PassRole Genehmigungen in China AWS-Regionen. Weitere Informationen zu `AdministratorAccess-AWSElasticBeanstalk` finden Sie unter Benutzerrichtlinien. Weitere Informationen zu `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` finden Sie unter Rollenrichtlinien für verwaltete Dienste.	9. März 2021
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy – Neue Richtlinie	Elastic Beanstalk hat eine neue Richtlinie hinzugefügt, um die `AWSElasticBeanstalkService` verwaltete Richtlinie zu ersetzen. Diese neue verwaltete Richtlinie verbessert die Sicherheit Ihrer Ressourcen, indem sie einen restriktiveren Berechtigungssatz anwendet. Weitere Informationen finden Sie unter Rollenrichtlinien für verwaltete Dienste.	3. März 2021

Änderungen	Beschreibung	Date
Elastic Beanstalk hat mit Verfolgung von Änderungen begonnen	Elastic Beanstalk hat damit begonnen, Änderungen für AWS verwaltete Richtlinien nachzuverfolgen.	1. März 2021

Protokollierung und Überwachung in Elastic Beanstalk

AWS bietet verschiedene Tools zur Überwachung Ihrer Elastic Beanstalk Beanstalk-Ressourcen und zur Reaktion auf potenzielle Vorfälle. Die Überwachung ist wichtig, um die Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer AWS Lösungen AWS Elastic Beanstalk aufrechtzuerhalten. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etwaigen Ausfall an mehreren Punkten leichter debuggen können.

Weitere Informationen zur Überwachung finden Sie unter Überwachungsumgebungen in Elastic Beanstalk.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

Erweiterte Integritätsberichte

Erweiterte Zustandsberichte ist eine Funktion, die Sie auf Ihrer Umgebung aktivieren können, damit Elastic Beanstalk weitere Informationen über Ressourcen in Ihrer Umgebung erfassen kann. Elastic Beanstalk analysiert die gesammelten Informationen, um ein besseres Bild des Gesamtumgebungszustands zu erhalten und um bei der Identifizierung von Problemen zu helfen, die dazu führen können, dass Ihre Anwendung nicht mehr verfügbar ist. Weitere Informationen finden Sie unter Verbesserte Gesundheitsberichterstattung und Überwachung in Elastic Beanstalk.

EC2 Amazon-Instanzprotokolle

Die EC2 Amazon-Instances in Ihrer Elastic Beanstalk Beanstalk-Umgebung generieren Protokolle, die Sie einsehen können, um Probleme mit Ihren Anwendungs- oder Konfigurationsdateien zu beheben. Protokolle, die vom Webserver, dem Anwendungsserver und den Skripten der Elastic Beanstalk-Plattform erstellt und lokal auf einzelnen Instances gespeichert CloudFormation werden. Sie können diese über die Environment Management Console oder die EB CLI ganz einfach abrufen. Sie können Ihre Umgebung auch so konfigurieren, dass Protokolle in Echtzeit an Amazon

CloudWatch Logs gestreamt werden. Weitere Informationen finden Sie unter Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen.

Umgebungsbenachrichtigungen

Sie können Ihre Elastic Beanstalk Umgebung so konfigurieren, dass Sie Amazon Simple Notification Service (Amazon SNS) verwenden, um Sie über wichtige Ereignisse zu benachrichtigen, die Ihre Anwendung betreffen. Geben Sie während oder nach der Erstellung der Umgebung eine E-Mail-Adresse an, um E-Mails zu erhalten, AWS wenn ein Fehler auftritt oder wenn sich der Zustand Ihrer Umgebung ändert. Weitere Informationen finden Sie unter Elastic Beanstalk-Umgebungsbenachrichtigungen mit Amazon SNS.

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Alarmen beobachten Sie eine einzelne Metrik über einen von Ihnen festgelegten Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Amazon SNS SNS-Thema oder eine AWS Auto Scaling Richtlinie gesendet. CloudWatch Alarme lösen keine Aktionen aus, da sie sich in einem bestimmten Status befinden. Stattdessen rufen Alarme Aktionen auf, wenn sich der Status geändert hat und für eine bestimmte Anzahl von Zeiträumen aufrechterhalten wurde. Weitere Informationen finden Sie unter Elastic Beanstalk mit Amazon verwenden CloudWatch.

AWS CloudTrail Logs

CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Service in Elastic Beanstalk ausgeführt wurden. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Elastic Beanstalk gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln. Weitere Informationen finden Sie unter Elastic Beanstalk API-Aufrufe protokollieren mit AWS CloudTrail.

AWS X-Ray Debuggen

X-Ray ist ein AWS Dienst, der Daten über die Anfragen sammelt, die Ihre Anwendung bedient, und anhand dieser Daten eine Service-Map erstellt, anhand derer Sie Probleme mit Ihrer Anwendung und Optimierungsmöglichkeiten identifizieren können. Sie können die AWS Elastic Beanstalk Konsole oder eine Konfigurationsdatei verwenden, um den X-Ray-Daemon auf den Instanzen in Ihrer Umgebung auszuführen. Weitere Informationen finden Sie unter AWS X-Ray Debugging konfigurieren.

Compliance-Validierung für Elastic Beanstalk

Die Sicherheit und Konformität von AWS Elastic Beanstalk wird von externen Prüfern im Rahmen mehrerer AWS Compliance-Programme bewertet. Dazu gehören SOC, PCI, FedRAMP, HIPAA und andere. AWS bietet unter AWS Services in Scope by Compliance Program eine häufig aktualisierte Liste der AWS Services im Rahmen bestimmter Compliance-Programme.

Prüfberichte von Drittanbietern stehen Ihnen zum Herunterladen zur Verfügung unter AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen in AWS Artifact.

Weitere Informationen zu AWS Compliance-Programmen finden Sie unter AWS Compliance-Programme.

Welche Compliance-Verpflichtungen Sie bei der Verwendung von Elastic Beanstalk haben, hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihrer Organisation und den geltenden Gesetzen und Vorschriften ab. Falls Ihre Nutzung von Elastic Beanstalk der Einhaltung von Standards wie HIPAA, PCI oder FedRAMP unterliegt, finden Sie hier Ressourcen, die Ihnen helfen: AWS

- Kurzanleitungen zu Sicherheit und Compliance — Implementierungsleitfäden, in denen architektonische Überlegungen erörtert und Schritte für die Implementierung von sicherheits- und Compliance-orientierten Basisumgebungen beschrieben werden. AWS
- Architecting for HIPAA Security and Compliance on Amazon Web Services — Ein Whitepaper, das beschreibt, wie Unternehmen HIPAA-konforme Anwendungen erstellen AWS können.
- AWS Compliance-Ressourcen — Eine Sammlung von Compliance-Arbeitsmappen und Leitfäden, die möglicherweise auf Ihre Branche und Ihren Standort zutreffen.
- AWS Config: Ein Service, der die Konformität Ihrer Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften bewertet.
- AWS Security Hub CSPM— Ein umfassender Überblick über Ihren Sicherheitsstatus innerhalb der Branche AWS , anhand dessen Sie überprüfen können, ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

Resilienz in Elastic Beanstalk

AWS Elastic Beanstalk verwaltet und automatisiert die Nutzung der AWS globalen Infrastruktur in Ihrem Namen. Wenn Sie Elastic Beanstalk verwenden, profitieren Sie von den Verfügbarkeits- und Fehlertoleranzmechanismen, die es AWS bietet.

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones.

AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind.

Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

Infrastruktursicherheit in Elastic Beanstalk

Als verwalteter Dienst AWS Elastic Beanstalk wird er durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die auf unserer Website mit bewährten Methoden für Sicherheit, Identität und Compliance beschrieben sind.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Elastic Beanstalk zuzugreifen. Clients müssen Transport Layer Security (TLS) 1.2 oder höher unterstützen. Clients müssen außerdem Cipher Suites mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Plattformen unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS-Security-Token-Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

Konfigurations- und Schwachstellenanalyse in Elastic Beanstalk

AWS und unsere Kunden sind gemeinsam dafür verantwortlich, ein hohes Maß an Sicherheit und Konformität der Softwarekomponenten zu erreichen. AWS Elastic Beanstalk hilft Ihnen dabei, Ihren Teil des Modells der gemeinsamen Verantwortung umzusetzen, indem es eine Funktion für verwaltete Updates bereitstellt. Diese Funktion wendet automatisch Patches und kleinere Aktualisierungen für eine von Elastic Beanstalk unterstützte Plattformversion an.

Weitere Informationen finden Sie unter Modell der übergreifenden Verantwortlichkeit für die Wartung der Elastic Beanstalk-Plattform.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

Bewährte Sicherheitsmethoden für Elastic Beanstalk

AWS Elastic Beanstalk bietet mehrere Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter AWS Elastic Beanstalk Sicherheit.

Bewährte Methoden für vorbeugende Sicherheitsmaßnahmen

Vorbeugende Sicherheitskontrollen versuchen, Vorfälle zu verhindern, bevor sie auftreten.

Implementieren des Zugriffs mit geringsten Berechtigungen

Elastic Beanstalk bietet AWS Identity and Access Management (IAM) verwaltete Richtlinien für Instanzprofile, Servicerollen und IAM-Benutzer. Diese verwalteten Richtlinien geben alle Berechtigungen an, die möglicherweise für den korrekten Betrieb Ihrer Umgebung und Anwendung erforderlich sind.

Ihre Anwendung benötigt möglicherweise nicht alle Berechtigungen in unseren verwalteten Richtlinien. Sie können sie anpassen und nur die Berechtigungen erteilen, die für die Instances Ihrer Umgebung, den Elastic Beanstalk-Service und Ihre Benutzer zum Ausführen ihrer Aufgaben erforderlich sind. Dies ist besonders relevant für Benutzerrichtlinien, in denen unterschiedliche Benutzerrollen möglicherweise unterschiedliche Berechtigungsanforderungen haben. Die Implementierung der geringstmöglichen Zugriffsrechte ist eine grundlegende Voraussetzung zum

Reduzieren des Sicherheitsrisikos und der Auswirkungen, die aufgrund von Fehlern oder böswilligen Absichten entstehen könnten.

Schützen Sie sensible Anwendungsdaten

Wenn Ihre Anwendung auf vertrauliche Informationen wie Anmeldeinformationen, API-Schlüssel oder Konfigurationsdaten zugreifen muss, gehen Sie aus Sicherheitsgründen wie folgt vor:

- Rufen Sie sensible Daten direkt aus dem AWS Secrets Manager AWS Systems Manager Parameterspeicher ab, indem Sie den jeweiligen Code SDKs oder APIs Ihren Anwendungscode verwenden. Dies bietet die sicherste und flexibelste Möglichkeit, auf vertrauliche Informationen zuzugreifen.
- Wenn Sie sensible Daten aus dem AWS Secrets Manager AWS Systems Manager Parameterspeicher als Umgebungsvariablen übergeben (siehe Abrufen von Geheimnissen in Umgebungsvariablen), beschränken Sie den Zugriff sorgfältig auf EC2 Schlüsselpaare und konfigurieren Sie die entsprechenden IAM-Rollen mit den geringsten Rechten für Ihre Instances.
- Drucken, protokollieren oder veröffentlichen Sie niemals vertrauliche Daten in Ihrem Anwendungscode, da diese Werte in Protokolldateien oder Fehlermeldungen landen könnten, die für nicht autorisierte Benutzer sichtbar sein könnten.

Aktualisieren Sie Ihre Plattformen regelmäßig

Elastic Beanstalk veröffentlicht regelmäßig neue Plattformversionen, um alle Plattformen zu aktualisieren. Neue Plattformversionen bieten Betriebssystem-, Laufzeit-, Anwendungsserver- und Webserver-Updates sowie Updates für Elastic Beanstalk-Komponenten. Viele dieser Plattformaktualisierungen enthalten wichtige Sicherheitskorrekturen. Stellen Sie sicher, dass Ihre Elastic Beanstalk-Umgebungen auf einer unterstützten Plattformversion ausgeführt werden (in der Regel die neueste Version für Ihre Plattform). Details hierzu finden Sie unter Aktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung.

Die einfachste Möglichkeit, die Plattform Ihrer Umgebung auf dem neuesten Stand zu halten, besteht darin, die Umgebung so zu konfigurieren, dass verwaltete Plattformaktualisierungen verwendet werden.

IMDSv2 Auf Umgebungsinstanzen durchsetzen

Amazon Elastic Compute Cloud (Amazon EC2) -Instances in Ihren Elastic Beanstalk Beanstalk-Umgebungen verwenden den Instance-Metadaten-Service (IMDS), eine On-Instance-Komponente,

um sicher auf Instance-Metadaten zuzugreifen. IMDS unterstützt zwei Methoden für den Datenzugriff: und. IMDSv1 IMDSv2 IMDSv2 verwendet sitzungsorientierte Anfragen und behebt verschiedene Arten von Sicherheitslücken, die beim Versuch, auf das IMDS zuzugreifen, genutzt werden könnten. Weitere Informationen zu den Vorteilen von finden Sie unter Verbesserungen IMDSv2, um den Instanz-Metadatendienst eingehender zu schützen. EC2

IMDSv2 ist sicherer, daher ist es eine gute Idee, die Verwendung von IMDSv2 für Ihre Instances zu erzwingen. Um dies durchzusetzen IMDSv2, stellen Sie sicher, dass alle Komponenten Ihrer Anwendung Unterstützung bieten IMDSv2, und deaktivieren Sie sie anschließend IMDSv1. Weitere Informationen finden Sie unter the section called “IMDS”.

Bewährte Methoden für aufdeckende Sicherheitsmaßnahmen

Aufdeckende Sicherheitskontrollen identifizieren Sicherheitsverstöße, nachdem sie aufgetreten sind. Sie können Ihnen dabei helfen, potenzielle Sicherheitsbedrohungen oder -vorfälle zu erkennen.

Implementieren der Überwachung

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Sicherheit, Verfügbarkeit und Leistung Ihrer Elastic Beanstalk Beanstalk-Lösungen. AWS bietet verschiedene Tools und Dienste, die Sie bei der Überwachung Ihrer AWS Dienste unterstützen.

Es folgen einige Beispiele für zu überwachende Elemente:

- CloudWatch Amazon-Metriken für Elastic Beanstalk — Richten Sie Alarme für wichtige Elastic Beanstalk-Metriken und für die benutzerdefinierten Metriken Ihrer Anwendung ein. Details hierzu finden Sie unter Elastic Beanstalk mit Amazon verwenden CloudWatch.
- AWS CloudTrail Einträge — Verfolge Aktionen, die sich auf die Verfügbarkeit auswirken könnten, wie oder. `UpdateEnvironment` `TerminateEnvironment` Details hierzu finden Sie unter Elastic Beanstalk API-Aufrufe protokollieren mit AWS CloudTrail.

Aktivieren AWS Config

AWS Config bietet eine detaillierte Ansicht der Konfiguration der AWS Ressourcen in Ihrem Konto. Sie können Ressourcenbeziehungen betrachten, einen Verlauf der Konfigurationsänderungen anzeigen und feststellen, wie sich Beziehungen und Konfigurationen im Lauf der Zeit ändern.

Sie können AWS Config damit Regeln definieren, mit denen Ressourcenkonfigurationen im Hinblick auf Datenkonformität bewertet werden. AWS Config Regeln stellen die idealen

Konfigurationseinstellungen für Ihre Elastic Beanstalk Beanstalk-Ressourcen dar. Wenn eine Ressource gegen eine Regel verstößt und als nicht konform gekennzeichnet wird, AWS Config kann ich Sie mithilfe eines Amazon Simple Notification Service (Amazon SNS) -Themas benachrichtigen. Details hierzu finden Sie unter Elastic Beanstalk Beanstalk-Ressourcen finden und verfolgen mit AWS Config.

Elastic Beanstalk Service-Rollen, Instanzprofile und Benutzerrichtlinien

Rollen sind Entitäten, die Sie mit AWS Identity and Access Management (IAM) erstellen, um Berechtigungen anzuwenden. Es gibt Rollen, die erforderlich sind, damit Ihre Elastic Beanstalk Beanstalk-Umgebung ordnungsgemäß funktioniert. Sie haben auch die Möglichkeit, Ihre eigenen benutzerdefinierten Richtlinien und Rollen zu erstellen, die Sie Benutzern oder Gruppen zuweisen können.

Erforderliche Rollen für Ihre Elastic Beanstalk Beanstalk-Umgebung

Wenn Sie eine Umgebung erstellen, werden Sie AWS Elastic Beanstalk aufgefordert, die folgenden AWS Identity and Access Management (IAM-) Rollen anzugeben:

- Servicerolle: Elastic Beanstalk übernimmt eine Servicerolle, um andere in AWS-Services Ihrem Namen zu nutzen.
- Instance-Profil Elastic Beanstalk wendet ein Instance-Profil auf die Amazon EC2 EC2-Instances in Ihrer Umgebung an. Mit dieser Aktion können sie erforderliche Aufgaben ausführen, z. B. das Abrufen von Informationen von Amazon Simple Storage Service (Amazon S3) und das Hochladen von Protokollen auf S3.

Erstellen Sie die Servicerolle und die EC2-Instance-Profilrolle

Wenn Ihr AWS Konto kein EC2-Instanzprofil oder keine Servicerolle hat, müssen Sie mit dem IAM-Dienst jeweils eines davon erstellen. Anschließend können Sie das EC2-Instanzprofil und die Servicerolle neuen Umgebungen zuweisen, die Sie erstellen. Der Assistent zum Erstellen einer Umgebung führt Sie zum IAM-Dienst, sodass Sie diese Rollen mit den erforderlichen Berechtigungen erstellen können.

Optionale Richtlinien und Rollen zur Verwaltung Ihrer Elastic Beanstalk Beanstalk-Umgebung

Sie können optional Benutzerrichtlinien erstellen und diese auf IAM-Benutzer und -Gruppen in Ihrem Konto anwenden. Auf diese Weise können die Benutzer Elastic Beanstalk Beanstalk-Anwendungen und -Umgebungen erstellen und verwalten. Sie können Elastic Beanstalk auch verwaltete Richtlinien

für Vollzugriff und Lesezugriff für Benutzer oder Gruppen zuweisen. Weitere Informationen zu diesen Richtlinien finden Sie unter the section called “Benutzerrichtlinien”.

Sie können Ihre eigenen Instance-Profile und Benutzerrichtlinien für erweiterte Szenarien erstellen. Wenn Ihre Instances Zugriff auf Services benötigen, die nicht in den Standardrichtlinien enthalten sind, können Sie eine neue Richtlinie erstellen oder weitere Richtlinien zum Standard hinzufügen. Wenn die verwaltete Richtlinie für Ihre Anforderungen zu offen ist, können Sie auch strengere Benutzerrichtlinien erstellen. Weitere Informationen zu AWS Berechtigungen finden Sie im IAM-Benutzerhandbuch.

Elastic Beanstalk-Servicerolle

Eine Servicerolle ist die IAM-Rolle, die Elastic Beanstalk annimmt, wenn andere Services in Ihrem Namen aufgerufen werden. Elastic Beanstalk verwendet beispielsweise eine Service-Rolle, wenn es Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing und Amazon EC2 Auto Scaling APIs aufruft, um Informationen zu sammeln. Die Servicerolle, die Elastic Beanstalk verwendet, ist diejenige, die Sie beim Erstellen der Elastic-Beanstalk-Umgebung angegeben haben.

Es gibt zwei verwaltete Richtlinien, die mit der Servicerolle verbunden sind. Diese Richtlinien stellen die Berechtigungen bereit, mit denen Elastic Beanstalk auf die erforderlichen AWS Ressourcen zugreifen kann, um Ihre Umgebungen zu erstellen und zu verwalten. Eine verwaltete Richtlinie bietet Berechtigungen für die erweiterte Zustandsüberwachung und die Unterstützung von Amazon SQS auf der Worker-Ebene, und eine andere bietet zusätzliche Berechtigungen, die für verwaltete Plattform-Updates erforderlich sind.

`AWSElasticBeanstalkEnhancedHealth`

Diese Richtlinie erteilt Berechtigungen für die Überwachung der Instance- und Umgebungsintegrität durch Elastic Beanstalk. Sie umfasst auch Amazon-SQS-Aktionen, damit Elastic Beanstalk die Warteschlangenaktivität für Worker-Umgebungen überwachen kann. Den Inhalt dieser verwalteten Richtlinie finden Sie auf der AWSElasticBeanstalkEnhancedHealthSeite im Referenzhandbuch für AWS verwaltete Richtlinien.

`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Diese Richtlinie erteilt Elastic Beanstalk Berechtigungen, um Umgebungen in Ihrem Namen zu aktualisieren, um verwaltete Plattformaktualisierungen durchzuführen. Den Inhalt dieser verwalteten

Richtlinie finden Sie AWSElasticBeanstalkManagedUpdatesCustomerRolePolicyauf der Seite im Referenzhandbuch für AWS verwaltete Richtlinien.

Service Level-Berechtigungsgruppierungen

Diese Richtlinie ist in Anweisungen gruppiert, die auf den bereitgestellten Berechtigungen basieren.

- *ElasticBeanstalkPermissions*— Diese Gruppe von Berechtigungen dient zum Aufrufen der Elastic Beanstalk-Serviceaktionen (Elastic APIs Beanstalk).
- *AllowPassRoleToElasticBeanstalkAndDownstreamServices* – Diese Berechtigungsgruppe ermöglicht es, jede Rolle an Elastic Beanstalk und an andere nachgelagerte Services wie CloudFormation zu übergeben.
- *ReadOnlyPermissions* — Diese Berechtigungsgruppe dient zum Sammeln von Informationen über die laufende Umgebung.
- **OperationPermissions* — Gruppen mit diesem Benennungsmuster dienen zum Aufrufen der notwendigen Vorgänge zur Durchführung von Plattformaktualisierungen.
- **BroadOperationPermissions* — Gruppen mit diesem Benennungsmuster dienen zum Aufrufen der notwendigen Vorgänge zur Durchführung von Plattformaktualisierungen. Sie enthalten auch umfassende Berechtigungen zur Unterstützung von Legacy-Umgebungen.
- **TagResource*— Gruppen mit diesem Benennungsmuster sind für Aufrufe vorgesehen, die Tags tag-on-create APIs an Ressourcen anhängen, die in einer Elastic Beanstalk Beanstalk-Umgebung erstellt werden.

Sie können eine Elastic-Beanstalk-Umgebung mit einem der folgenden Ansätze erstellen. In jedem Abschnitt wird beschrieben, wie der Ansatz mit der Servicerolle umgeht.

Elastic-Beanstalk-Konsole

Wenn Sie eine Umgebung mit der Elastic-Beanstalk-Konsole erstellen, werden Sie von Elastic Beanstalk aufgefordert, eine Servicerolle namens `aws-elasticbeanstalk-service-role` zu erstellen. Wenn diese Rolle über Elastic Beanstalk erstellt wird, enthält sie eine Vertrauensrichtlinie, die Elastic Beanstalk erlaubt, die Servicerolle zu übernehmen. Die beiden verwalteten Richtlinien, die weiter oben in diesem Thema beschrieben wurden, sind ebenfalls mit der Rolle verknüpft.

Elastic Beanstalk Command Line Interface (EB CLI)

Sie können eine Umgebung mit dem Befehl the section called “eb create” des Elastic Beanstalk Command Line Interface (EB CLI) erstellen. Wenn Sie keine Servicerolle über die Option `--`

`service-role` angeben. Elastic Beanstalk erstellt dieselbe Standard-Servicerolle `aws-elasticbeanstalk-service-role`. Wenn die standardmäßige Servicerolle bereits vorhanden ist, verwendet Elastic Beanstalk diese für die neue Umgebung. Wenn diese Rolle über Elastic Beanstalk erstellt wird, enthält sie eine Vertrauensrichtlinie, die Elastic Beanstalk erlaubt, die Servicerolle zu übernehmen. Die beiden verwalteten Richtlinien, die weiter oben in diesem Thema beschrieben wurden, sind ebenfalls mit der Rolle verknüpft.

Elastic-Beanstalk-API

Sie können über die Aktion `CreateEnvironment` der Elastic Beanstalk Beanstalk-API eine Umgebung erstellen. Wenn Sie keine Servicerolle angeben, erstellt Elastic Beanstalk eine servicegebundene Überwachungsrolle. Dies ist eine einzigartige Art von Servicerolle, die von Elastic Beanstalk so vordefiniert ist, dass sie alle Berechtigungen enthält, die der Service benötigt, um andere in AWS-Services Ihrem Namen aufzurufen. Die serviceverknüpfte Rolle ist Ihrem Konto zugeordnet. Sie wird von Elastic Beanstalk einmalig erstellt und dann zum Erstellen weiterer Umgebungen wiederverwendet. Sie können die serviceverknüpfte Überwachungsrolle Ihres Kontos auch im Voraus mit IAM erstellen. Wenn Ihr Konto eine servicegebundene Überwachungsrolle hat, können Sie damit eine Umgebung mit der Elastic Beanstalk-Konsole, der Elastic Beanstalk- oder der EB CLI erstellen. Weitere Informationen zur Verwendung von serviceverknüpften Rollen mit Elastic-Beanstalk-Umgebungen finden Sie unter Servicegebundene Rollen für Elastic Beanstalk verwenden.

Weitere Informationen zu Servicerollen finden Sie unter Rollen von Elastic Beanstalk Service verwalten.

Elastic Beanstalk-Instance-Profil

Ein Instance-Profil ist eine IAM-Rolle, die für in der Elastic-Beanstalk-Umgebung gestartete Amazon-EC2-Instances gilt. Wenn Sie eine Elastic-Beanstalk-Umgebung erstellen, geben Sie das Instance-Profil an, das verwendet wird, wenn Ihre EC2-Instances Folgendes tun:

- Anwendungsversionen von Amazon Simple Storage Service (Amazon S3) abrufen
- Protokolle in Amazon S3 schreiben
- In integrierten AWS X-Ray -Umgebungen Debugging-Daten in X-Ray hochladen
- In von Amazon ECS verwalteten Docker-Umgebungen Container-Bereitstellungen mit Amazon Elastic Container Service (Amazon ECS) koordinieren
- In Worker-Umgebungen aus einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange lesen

- In Worker-Umgebungen die Leader-Wahl mit Amazon DynamoDB durchführen
- Veröffentlichen Sie in Mitarbeiterumgebungen Instanzzustandsmetriken auf Amazon CloudWatch

Verwaltete Richtlinien

Elastic Beanstalk bietet eine Reihe von verwalteten Richtlinien, mit denen die EC2-Instances in Ihrer Umgebung die erforderlichen Operationen ausführen können. Die folgenden verwalteten Richtlinien sind für grundlegende Anwendungssfälle erforderlich.

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

Wenn Ihre Webanwendung Zugriff auf weitere zusätzliche Dienste benötigt AWS-Services, fügen Sie dem Instance-Profil Anweisungen oder verwaltete Richtlinien hinzu, die den Zugriff auf diese Dienste ermöglichen. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zum Instance-Standardprofil.

Ein EC2-Instanzprofil erstellen

Wenn Ihr AWS Konto kein EC2-Instanzprofil hat, müssen Sie eines mithilfe des IAM-Dienstes erstellen. Anschließend können Sie das EC2-Instance-Profil neuen Umgebungen zuweisen, die Sie erstellen. Die Schritte Umgebung erstellen in der Elastic Beanstalk Beanstalk-Konsole bieten Ihnen Zugriff auf die IAM-Konsole, sodass Sie ein EC2-Instance-Profil mit den erforderlichen Berechtigungen erstellen können.

Sie können auch ein EC2-Instance-Profil erstellen, indem Sie direkt auf die IAM-Konsole zugreifen, ohne die Elastic Beanstalk-Konsole verwenden zu müssen. Ausführliche Schritte zum Erstellen eines Elastic Beanstalk EC2-Instance-Profils in der IAM-Konsole finden Sie unter. Erstellen eines Instance-Profils

Elastic Beanstalk-Benutzerrichtlinie

Erstellen Sie IAM-Benutzer für jeden Benutzer, der Elastic Beanstalk verwendet, damit Sie Ihr Stammkonto nicht verwenden oder Anmeldeinformationen weitergeben müssen. Als bewährte Sicherheitsmethode sollten Sie diesen Benutzern nur Berechtigungen für den Zugriff auf Dienste und Funktionen erteilen, die sie benötigen.

Elastic Beanstalk erfordert Berechtigungen nicht nur für die eigenen API-Aktionen, sondern auch für einige andere AWS -Services. Elastic Beanstalk verwendet Benutzerberechtigungen, um Ressourcen in einer Umgebung zu starten. Zu diesen Ressourcen gehören EC2-Instances, ein Elastic Load Balancing Load Balancer und eine Auto-Scaling-Gruppe. Elastic Beanstalk verwendet Benutzerberechtigungen auch, um Protokolle und Vorlagen in Amazon Simple Storage Service (Amazon S3) zu speichern, Benachrichtigungen an Amazon SNS zu senden, Instance-Profile zuzuweisen und Metriken zu veröffentlichen. CloudWatch Elastic Beanstalk benötigt CloudFormation Berechtigungen, um Ressourcenbereitstellungen und Updates zu orchestrieren. Es sind auch Amazon RDS-Berechtigungen zum Erstellen von Datenbanken (bei Bedarf) und Amazon SQS-Berechtigungen zum Erstellen von Warteschlangen für Worker-Umgebungen erforderlich.

Weitere Informationen zu Benutzerrichtlinien finden Sie unter Elastic Beanstalk-Benutzerrichtlinien verwalten.

Tutorials und Beispiele

Sprach- und Framework-spezifische Tutorials sind im AWS Elastic Beanstalk Entwicklerhandbuch verteilt. Neue und aktualisierte Tutorials werden dieser Liste nach ihrer Veröffentlichung hinzugefügt. Die neuesten Aktualisierungen werden zuerst angezeigt.

Diese Tutorials richten sich an fortgeschrittene Benutzer und enthalten keine Anweisungen für grundlegende Schritte, wie z. B. die Anmeldung bei AWS. Wenn Sie Elastic Beanstalk zum ersten Mal verwenden AWS , schauen Sie sich die Getting Started Walkthrough an, um Ihre erste Elastic Beanstalk Beanstalk-Umgebung zum Laufen zu bringen.

- Ruby on Rails – Bereitstellen einer Rails-Anwendung in Elastic Beanstalk
- Ruby und Sinatra – Bereitstellen einer Sinatra-Anwendung in Elastic Beanstalk
- PHP und MySQL HA-Konfiguration – Bereitstellen einer PHP-Anwendung mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank für Elastic Beanstalk
- PHP und Laravel – Bereitstellen einer Laravel-Anwendung in Elastic Beanstalk
- PHP und CakePHP – Bereitstellen einer CakePHP-Anwendung in Elastic Beanstalk
- PHP und Drupal HA-Konfiguration – Bereitstellen einer Drupal-Website mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank für Elastic Beanstalk
- PHP- und HA-Konfiguration - WordPress Bereitstellung einer WordPress Website mit hoher Verfügbarkeit mit einer externen Amazon RDS-Datenbank auf Elastic Beanstalk
- Node.js mit DynamoDB HA-Konfiguration – Bereitstellen einer Node.js-Anwendung mit DynamoDB in Elastic Beanstalk
- ASP.NET Core – QuickStart: Stellen Sie eine ASP.NET-Anwendung auf Elastic Beanstalk bereit
- Python und Flask – Bereitstellen einer Flask-Anwendung in Elastic Beanstalk
- Python und Django – Bereitstellen einer Django-Anwendung in Elastic Beanstalk
- Node.js und Express – Eine Node.js Express-Anwendung auf Elastic Beanstalk bereitstellen
- Docker, PHP und nginx – Erstellen einer von ECS verwalteten Docker-Umgebung mit der Elastic Beanstalk Beanstalk-Konsole

Unter den nachfolgenden Links können Sie die von Elastic Beanstalk verwendeten Beispielanwendungen herunterladen, um eine Umgebung ohne Quell-Bundle zu erstellen:

- Docker – docker.zip

- Docker für mehrere Container — 2.zip docker-multicontainer-v
- Vorkonfigurierter Docker (Glassfish) — 1.zip docker-glassfish-v
- Go – go.zip
- Corretto – corretto.zip
- Tomcat – tomcat.zip
- .NET Core unter Linux — .zip dotnet-core-linux
- .NET Core — dotnet-asp-windows .zip
- Node.js – nodejs.zip
- PHP – php.zip
- Python – python.zip
- Ruby – ruby.zip

Umfangreichere Beispielanwendungen, die die Verwendung zusätzlicher Web-Frameworks, Bibliotheken und Tools zeigen, sind als Open-Source-Projekte verfügbar unter GitHub:

- Load-Balancing WordPress (Tutorial) — Konfigurationsdateien für die WordPress sichere Installation und Ausführung in einer Elastic Beanstalk Beanstalk-Umgebung mit Lastenausgleich.
- Drupal mit Lastenausgleich (Tutorial) — Konfigurationsdateien und Anweisungen für die sichere Installation und Ausführung von Drupal in einer Elastic Beanstalk Beanstalk-Umgebung mit Lastenausgleich.
- Scorekeep — RESTful Web-API, die das Spring-Framework verwendet und eine Schnittstelle für die Erstellung und AWS SDK für Java Verwaltung von Benutzern, Sitzungen und Spielen bereitstellt. Die API ist mit einer Angular 1.5-Web-App gebündelt, von der die API über HTTP eingesetzt wird. Beinhaltet Zweige, die die Integration mit Amazon Cognito und Amazon Relational Database Service zeigen. AWS X-Ray

  Die Anwendung nutzt Funktionen der Java SE-Plattform, um Abhängigkeiten herunterzuladen und auf Instances zu erstellen, und minimiert so die Größe des Quell-Bundles. Die Anwendung umfasst auch nginx-Konfigurationsdateien, mit denen die Standardkonfiguration überschrieben wird, sodass die Bereitstellung der Frontend-Webanwendung statisch auf Port 80 über den Proxy erfolgt und Anforderungen über Pfade unter `/api` an die auf `localhost:5000` ausgeführte API weitergeleitet werden.
- Hat es Schlangen? - Tomcat-Anwendung, die die Verwendung von RDS in einer Java EE-Webanwendung in Elastic Beanstalk zeigt. Das Projekt zeigt die Verwendung von Servlets, Simple

Tag Support JSPs, Tag Files, JDBC, SQL, Log4J, Bootstrap, Jackson und Elastic Beanstalk Konfigurationsdateien.

- Locust Load Generator – Dieses Projekt zeigt die Verwendung von Java SE-Plattformfunktionen zur Installation und Ausführung von Locust. Dieses Tool für die Lastgenerierung ist in Python geschrieben. Das Projekt umfasst Konfigurationsdateien zur Installation und Konfiguration von Locust, ein Build-Skript zur Konfiguration einer DynamoDB-Tabelle sowie eine Procfile-Datei für die Locust-Ausführung.
- Share Your Thoughts (Tutorial) – PHP-Anwendung, mit der die Nutzung von MySQL auf Amazon RDS, Composer und Konfigurationsdateien gezeigt wird.
- Eine Beispielanwendung New Startup (Tutorial) — Node.js, die die Verwendung von DynamoDB, dem AWS SDK für JavaScript in Node.js, der npm-Paketverwaltung und Konfigurationsdateien zeigt.

Migration von IIS-Anwendungen zu Elastic Beanstalk

AWS Elastic Beanstalk bietet einen optimierten Migrationspfad für Ihre Windows-Anwendungen, die auf Internetinformationsdiensten (IIS) ausgeführt werden. Die in diesem Kapitel beschriebene Migrationsfunktion reduziert den Zeitaufwand und die Komplexität, die normalerweise mit Cloud-Migrationen verbunden sind, erheblich und hilft Ihnen, die Anwendungsfunktionalität und die Konfigurationsintegrität während der Umstellung auf aufrechtzuerhalten. AWS

Die Operation eb migrate

Verwenden Sie den eb migrate Befehl in der Elastic Beanstalk Command Line Interface (EB CLI), um Ihre IIS-Anwendungen automatisch zu erkennen, zu verpacken und auf dem bereitzustellen. AWS Cloud Der Prozess behält die Anwendungsfunktionalität bei und bewahrt Ihre Konfigurationen, einschließlich Bindungen, Anwendungspools und Authentifizierungseinstellungen.

In den folgenden Schritten wird der Prozess zusammengefasst, den der `eb migrate` Vorgang zur Umstellung Ihrer Anwendung auf Folgendes durchführt: AWS Cloud

1. Entdecken Sie IIS-Websites und ihre Konfigurationen.
2. Inhalt und Konfiguration der Paketanwendung.
3. Erstellen Sie eine Elastic Beanstalk Beanstalk-Umgebung und -Anwendung.
4. Stellen Sie die Anwendung mit beibehaltenen Einstellungen bereit.

Optionen für die Ausführung von Arbeitsabläufen und Standorten

Der eb migrate Befehl bietet Optionen für flexible Migrationsworkflows und Ausführungsplätze. Führen Sie den Befehl standardmäßig auf dem Zielserver aus, der die Anwendung enthält, die Sie zu Elastic Beanstalk migrieren möchten. Wenn Sie Befehle nicht direkt auf dem Anwendungsserver ausführen können, verwenden Sie die `remote` Option, um den Befehl von einem Bastion-Host aus auszuführen, der eine Verbindung zu dem Zielserver herstellt, der Ihre Anwendung und Konfigurationen enthält. Um die Migration in zwei Schritten abzuschließen, können Sie das Migrationspaket auch generieren, ohne es mithilfe der `archive-only` Option bereitzustellen, und es dann später nach Belieben mithilfe der `archive` Option bereitstellen.

Referenzinformationen zu diesem eb migrate Befehl finden Sie unterthe section called “eb migrate”.

Topics

Die folgenden Themen enthalten detaillierte Informationen zur Migration von IIS-Anwendungen zu Elastic Beanstalk:

- the section called “Voraussetzungen”- Machen Sie sich mit der Software, dem Zugriff und den Berechtigungen vertraut, die für die Migration Ihrer Windows-Anwendungen in Umgebungen erforderlich sind. AWS Elastic Beanstalk
- the section called “Glossar zur Migration”- Verstehen Sie, wie IIS-Komponenten Elastic Beanstalk-Ressourcen zugeordnet werden
- the section called “Zuordnung der Migration von IIS zu Elastic Beanstalk”- Verstehen Sie, wie IIS-Komponenten Elastic Beanstalk-Ressourcen zugeordnet werden
- the section called “Grundlegende IIS-Migrationen”- Erfahren Sie, wie Sie grundlegende Migrationen durchführen
- the section called “Fortgeschrittene Migrationsszenarien”- Bewältigen Sie komplexe Migrationsszenarien
- the section called “Security configuration (Sicherheitskonfiguration)”- Konfigurieren Sie die Sicherheitseinstellungen während der Migration
- the section called “Netzwerkkonfiguration”- Netzwerk- und Portkonfigurationen verwalten
- the section called “Problembehebung und Diagnose”- Beheben Sie häufig auftretende Migrationsprobleme
- the section called “Migrationsoptionen: EB CLI im Vergleich zu MGN”- Vergleichen Sie zwei primäre Migrationsoptionen.

Voraussetzungen

Stellen Sie vor der Verwendung des eb migrate Befehls sicher, dass Ihre Umgebung die folgenden Anforderungen erfüllt:

IIS-Installation und -Version

Auf dem Server, von dem Sie migrieren, müssen Internetinformationsdienste (IIS) Version 7.0 oder höher ausgeführt werden. IIS 10.0 auf Windows Server 2016 oder höher bietet die kompatibelste Umgebung für die Migration.

Führen Sie den folgenden Befehl aus, um Ihre IIS-Version zu überprüfen:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\InetStp\"
...
```

```
SetupString             : IIS 10.0
VersionString           : Version 10.0
...
```

Anforderungen für Windows Server

In Ihrer Quellumgebung sollte Windows Server 2016 oder höher ausgeführt werden, um eine optimale Kompatibilität zu gewährleisten. Elastic Beanstalk unterstützt diese Windows Server-Versionen als Zielplattformen:

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016

EB CLI-Installation

- Standard-Workflow (ohne `--remote` Option):
  - Python und das Elastic Beanstalk Command Line Interface (EB CLI) müssen auf dem Server installiert sein, der die Anwendung enthält, die Sie zu Elastic Beanstalk migrieren möchten. Dies ist zwar nicht erforderlich, wir empfehlen jedoch, die EB-CLI in einer `virtualenv` Sandbox zu installieren, wie unter beschrieben. the section called “In virtualenv installieren”
- Verwenden der `--remote` Option:
  - Python und das Elastic Beanstalk Command Line Interface (EB CLI) müssen auf Ihrem Bastion-Host installiert sein. Dies ist zwar nicht erforderlich, wir empfehlen jedoch, die EB-CLI in einer `virtualenv` Sandbox zu installieren, wie unter beschrieben. the section called “In virtualenv installieren”

Erforderliche -Berechtigungen

Sie benötigen die folgenden Anmeldeinformationen und Berechtigungen:

- Administratorrechte auf dem IIS-Quellserver oder auf dem Bastion-Host (falls Sie die `--remote` Option verwenden).
- AWS Anmeldeinformationen mit Berechtigungen zum Erstellen und Verwalten von Elastic Beanstalk Beanstalk-Ressourcen

Web Deploy 3.6

Das Microsoft Web Deploy Tool (Version 3.6 oder höher) muss auf Ihrem Quellserver oder auf dem Bastion-Host installiert sein (falls Sie die `--remote` Option verwenden). Dieses Tool wird von verwendet, um Ihre Anwendungen eb migrate zu verpacken.

Führen Sie den folgenden Befehl aus, um die Installation zu überprüfen:

:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\IIS
 Extensions\MSDeploy\3" -Name InstallPath

InstallPath  : C:\Program Files\IIS\Microsoft Web Deploy V3\
...
```

Installationsanweisungen finden Sie unter Installation und Konfiguration von Web Deploy auf IIS 8.0 oder höher auf der Microsoft Windows-Produktdokumentationswebsite.

Netzwerkanforderungen

- Standard-Workflow (ohne *--remote* Option):
  - Ihr Quellserver muss über ausgehenden Internetzugang zu AWS Diensten verfügen.
- Verwenden der *--remote* Option:
  - Ihr Quellserver muss über ausgehenden Internetzugang zu AWS Diensten verfügen.
  - Konfigurieren Sie die richtigen Regeln für den Zugriff auf Sicherheitsgruppen, die eine ausgehende Netzwerkverbindung von Ihrem Bastion-Host und eine eingehende Verbindung zum Remote-Computer ermöglichen. Stellen Sie sicher, dass die IP des Bastion-Hosts über TCP auf Port 22 zugelassen ist, um auf den Remote-Computer zuzugreifen.
  - Stellen Sie sicher, dass Ihr SSH-Client sowohl auf Ihrem Remote-Computer als auch auf Ihrem Bastion-Host installiert ist und läuft.
  - Stellen Sie sicher, dass Ihre Firewall-Konfiguration die entsprechenden Regeln enthält, die Port 22 öffnen oder eine Verbindung zum Client ermöglichen.
  - Testen Sie Ihre Verbindung, indem Sie manuell per SSH vom Bastion-Host aus auf den Remote-Host zugreifen, bevor Sie eine Migration versuchen.

Glossar zur Migration

Dieses Glossar enthält Definitionen für wichtige Begriffe und Konzepte im Zusammenhang mit IIS, Elastic Beanstalk und der Migration von IIS-Anwendungen zu Elastic Beanstalk.

Windows-, IIS- und .NET-Begriffe

IIS

Internet Information Services, eine Webserver-Software, die von Microsoft für die Verwendung mit Windows Server entwickelt wurde. IIS hostet Websites, Webanwendungen und Webdienste und bietet eine Plattform für die Ausführung von ASP.NET und anderen Webtechnologien. Während der Migration zu Elastic Beanstalk werden IIS-Sites und ihre Konfigurationen gepackt und auf Windows Server-Instanzen in der AWS Cloud bereitgestellt.

IIS-Versionen 7.0 und höher werden für die Migration unterstützt, wobei IIS 10.0 auf Windows Server 2016 oder höher die kompatibelste Umgebung bietet.

.NET Framework

Eine von Microsoft entwickelte Softwareentwicklungsplattform zum Erstellen und Ausführen von Windows-Anwendungen. Sie bietet eine umfangreiche Klassenbibliothek namens Framework Class Library (FCL) und unterstützt die Sprachinteroperabilität zwischen mehreren Programmiersprachen.

Bei der Migration zu Elastic Beanstalk werden Anwendungen, die auf dem.NET Framework basieren, weiterhin auf derselben Version des Frameworks in der Cloud-Umgebung ausgeführt. Elastic Beanstalk unterstützt mehrere Versionen von.NET Framework (4.x) auf seinen Windows Server-Plattformen.

.NET Core

Ein plattformübergreifender Open-Source-Nachfolger des.NET-Frameworks, der modularer und leichter konzipiert wurde. .NET Core (jetzt einfach.NET 5 und höher genannt) ermöglicht es Entwicklern, Anwendungen zu erstellen, die unter Windows, Linux und macOS ausgeführt werden.

Bei der Migration von auf .NET Core basierenden Anwendungen zu Elastic Beanstalk können Sie je nach den Anforderungen und Abhängigkeiten Ihrer Anwendung zwischen Windows Server-Plattformen oder Linux-basierten Plattformen wählen.

Common Language Runtime (CLR)

Die Komponente für virtuelle Maschinen von.NET Framework, die die Ausführung von.NET-Programmen verwaltet. Die CLR bietet Dienste wie Speicherverwaltung, Typsicherheit, Ausnahmebehandlung, Garbage-Collection und Threadverwaltung.

Bei der Migration zu Elastic Beanstalk ist die entsprechende CLR-Version automatisch auf der von Ihnen ausgewählten Windows Server-Plattform verfügbar, sodass die Kompatibilität mit den Anforderungen Ihrer Anwendung gewährleistet ist.

Website

Ein logischer Container in IIS, der eine Webanwendung oder einen Webdienst darstellt, identifiziert durch eine eindeutige Bindung von IP-Adresse, Port und Host-Header. Jede IIS-Site hat ihren eigenen Anwendungspool, Bindungen und Konfigurationseinstellungen und kann eine oder mehrere Anwendungen enthalten.

Anwendung

Eine Gruppierung von Inhalts- und Codedateien innerhalb einer IIS-Site, die Anfragen für einen bestimmten URL-Bereich verarbeitet. Anwendungen in IIS können ihre eigenen Konfigurationseinstellungen haben, die normalerweise in `web.config` Dateien gespeichert werden.

Bei der Migration zu Elastic Beanstalk werden Anwendungen mit ihrer Pfadstruktur und ihren Konfigurationseinstellungen beibehalten. Der Migrationsprozess stellt sicher, dass verschachtelte Anwendungen ihre Hierarchie und URL-Pfade in der Cloud-Umgebung beibehalten.

ApplicationPool

Eine IIS-Funktion, die Webanwendungen isoliert, um die Sicherheit, Zuverlässigkeit und Leistungsverwaltung zu verbessern. Anwendungspools definieren die Laufzeitumgebung für Anwendungen, einschließlich der.NET Framework-Version, des Pipeline-Modus und der Identitätseinstellungen.

VirtualDirectory

Eine Verzeichniszuordnung in IIS, mit der Inhalte von einem Speicherort außerhalb des Stammverzeichnisses der Site bereitgestellt werden können. Virtuelle Verzeichnisse ermöglichen es Ihnen, Inhalte an verschiedenen physischen Standorten zu organisieren und Benutzern gleichzeitig eine einheitliche URL-Struktur zu bieten.

Bei der Migration zu Elastic Beanstalk werden virtuelle Verzeichnisse mit ihren Pfadzuordnungen beibehalten. Der eb migrate Befehl erstellt die erforderliche Verzeichnisstruktur und Konfigurationen in der Cloud-Umgebung, um dieselben URL-Pfade beizubehalten.

ARR

Application Request Routing, eine IIS-Erweiterung, die Lastenausgleichs- und Proxyfunktionen für Webserver bietet. ARR ermöglicht URL-basiertes Routing, HTTP-Anforderungsweiterleitung und Lastverteilung auf mehrere Server.

Während der Migration zu Elastic Beanstalk werden ARR-Konfigurationen durch die Installation von ARR-Funktionen auf EC2-Instances und die Konfiguration der entsprechenden Routing-Regeln beibehalten. Bei komplexen Routing-Szenarien kann der Migrationsprozess auch Application Load Balancer Balancer-Regeln nutzen, um ähnliche Funktionen zu implementieren.

URL umschreiben

Ein IIS-Modul, das angeforderte Änderungen auf der URLs Grundlage definierter Regeln vornimmt, bevor sie die Webanwendung erreichen. URL Rewrite ermöglicht URL-Manipulation, Umleitung und Inhaltsbereitstellung auf der Grundlage von Mustern und Bedingungen.

Bei der Migration zu Elastic Beanstalk werden URL-Rewrite-Regeln aus Ihren `web.config` Dateien nach Möglichkeit in ALB-Routing-Regeln übersetzt oder in der IIS-Konfiguration auf den EC2-Instances beibehalten. Dadurch wird sichergestellt, dass URL-Muster und Umleitungen in der Cloud-Umgebung weiterhin wie erwartet funktionieren.

msdeploy.exe

Ein Befehlszeilentool, das für die Bereitstellung von Webanwendungen und Websites auf IIS-Servern verwendet wird. Es wird auch als Web Deploy bezeichnet und bietet die Möglichkeit, Webanwendungen, Websites und Serverkonfigurationen zu verpacken, zu synchronisieren und bereitzustellen.

Der eb migrate Befehl verwendet Web Deploy (Version 3.6 oder höher), um Ihre Anwendungen während der Migration zu Elastic Beanstalk zu verpacken. Dieses Tool muss auf Ihrem Quellserver installiert sein, damit der Migrationsprozess ordnungsgemäß funktioniert.

Physischer Pfad

Der tatsächliche Speicherort im Dateisystem, an dem die Inhaltsdateien einer IIS-Website oder -Anwendung gespeichert sind. Physikalische Pfade können auf lokale Verzeichnisse, Netzwerkfreigaben oder andere Speicherorte verweisen, auf die der IIS-Server Zugriff hat.

Während der Migration zu Elastic Beanstalk werden physische Pfade den entsprechenden Speicherorten auf den EC2-Instances in Ihrer Umgebung zugeordnet. Beim Migrationsprozess

wird die Inhaltsstruktur beibehalten und gleichzeitig sichergestellt, dass alle Dateien ordnungsgemäß in der Cloud-Umgebung bereitgestellt werden.

ApplicationHost.config

Die Stammkonfigurationsdatei für IIS, die serverweite Einstellungen definiert und die Konfiguration für alle Sites, Anwendungen und virtuellen Verzeichnisse enthält. Diese Datei befindet sich im `%windir%\System32\inetsrv\config` Verzeichnis und steuert das allgemeine Verhalten des IIS-Servers.

Bei der Migration zu Elastic Beanstalk `applicationHost.config` werden die relevanten Einstellungen aus extrahiert und auf die IIS-Konfiguration auf den EC2-Instances in Ihrer Umgebung angewendet. Dadurch wird sichergestellt, dass die serverweiten Einstellungen während der Migration beibehalten werden.

web.config

Eine XML-basierte Konfigurationsdatei, die in ASP.NET-Anwendungen verwendet wird, um Anwendungseinstellungen, Sicherheit und Verhalten auf Anwendungs- oder Verzeichnisebene zu steuern. `web.config`Dateien können Einstellungen für Authentifizierung, Autorisierung, Sitzungsstatus, Kompilierung und benutzerdefinierte Anwendungsparameter enthalten.

Während der Migration zu Elastic Beanstalk werden `web.config` Dateien beibehalten und zusammen mit Ihrer Anwendung bereitgestellt. Der Migrationsprozess stellt sicher, dass anwendungsspezifische Konfigurationen in der Cloud-Umgebung weiterhin wie erwartet funktionieren.

DefaultDocument

Eine IIS-Funktion, die die Standarddatei angibt, die bereitgestellt werden soll, wenn ein Benutzer ein Verzeichnis anfordert, ohne einen Dateinamen anzugeben. Standarddokumente sind standardmäßig aktiviert, und IIS 7 definiert die folgenden Standarddokumentdateien in der `applicationHost.config` Datei als serverweite Standardeinstellungen: Default.htm, Default.asp, Index.htm, Index.html, Iisstart.htm.

Bei der Migration zu Elastic Beanstalk werden die Standarddokumenteinstellungen in der IIS-Konfiguration auf den EC2-Instances beibehalten, wodurch sichergestellt wird, dass Verzeichnisanfragen in der Cloud-Umgebung konsistent verarbeitet werden.

System.WebServer

Ein Konfigurationsabschnitt in `web.config` oder`applicationHost.config`, der IIS-spezifische Einstellungen für Module, Handler und anderes Serververhalten enthält. In diesem

Abschnitt wird gesteuert, wie IIS Anfragen verarbeitet, Module verwaltet und Serverfunktionen konfiguriert.

Während der Migration zu Elastic Beanstalk werden System.WebServer-Konfigurationen in den `web.config` Dateien Ihrer Anwendung beibehalten und auf die IIS-Installation auf den EC2-Instances in Ihrer Umgebung angewendet. Dadurch wird sichergestellt, dass IIS-spezifische Verhaltensweisen in der Cloud-Umgebung beibehalten werden.

Allgemeine Geschäftsbedingungen von Elastic Beanstalk

Plattform

Eine Kombination aus Betriebssystem-, Programmiersprachen-Runtime-, Webserver-, Anwendungsserver- und Elastic Beanstalk Beanstalk-Komponenten, die den Software-Stack für die Ausführung von Anwendungen definieren.

Für Windows-Migrationen bietet Elastic Beanstalk Plattformen, die auf Windows Server 2016, 2019 und 2022 mit IIS und verschiedenen .NET Framework-Versionen basieren, um die Kompatibilität mit Ihrer Quellumgebung sicherzustellen.

SolutionStack

Eine vordefinierte Plattformkonfiguration in Elastic Beanstalk, die das Betriebssystem, die Laufzeit und andere Komponenten spezifiziert, die für die Ausführung einer Anwendung erforderlich sind. Konzeptionell identisch mit einer Plattform und wird synonym für den Betrieb von Umgebungen verwendet.

Während der Migration wählt der eb migrate Befehl auf der Grundlage der Konfiguration Ihrer Quellumgebung einen geeigneten Lösungsstapel aus, um die Kompatibilität mit Ihren IIS-Anwendungen sicherzustellen.

CreateEnvironment

Eine Elastic Beanstalk Beanstalk-API-Aktion, die eine neue Umgebung zum Hosten einer Anwendungsversion erstellt. Diese API wird vom eb migrate Befehl verwendet, um die erforderlichen AWS Ressourcen für Ihre migrierte Anwendung bereitzustellen.

Der Migrationsprozess konfiguriert die entsprechenden Umgebungsparameter auf der Grundlage Ihrer IIS-Quellumgebung, einschließlich Instanztyp, Umgebungsvariablen und Optionseinstellungen.

CreateApplicationVersion

Eine Elastic Beanstalk Beanstalk-API-Aktion, die eine neue Anwendungsversion aus einem in Amazon S3 gespeicherten Quellpaket erstellt. Der eb migrate Befehl verwendet diese API, um Ihre verpackte IIS-Anwendung als Version in Elastic Beanstalk zu registrieren.

Während der Migration werden Ihre Anwendungsdateien und die Konfiguration gepackt, auf Amazon S3 hochgeladen und vor der Bereitstellung als Anwendungsversion registriert.

DescribeEvents

Eine Elastic Beanstalk Beanstalk-API-Aktion, die eine Liste von Ereignissen für eine Umgebung abruft, einschließlich Bereitstellungen, Konfigurationsänderungen und Betriebsproblemen. Der eb migrate Befehl verwendet diese API, um den Fortschritt Ihrer Migration zu überwachen.

Sie können den eb events Befehl auch nach der Migration verwenden, um den Verlauf der Ereignisse in Ihrer Umgebung anzuzeigen.

DescribeEnvironmentHealth

Eine Elastic Beanstalk Beanstalk-API-Aktion, die detaillierte Gesundheitsinformationen zu den Instances und anderen Komponenten einer Umgebung bereitstellt. Diese API wird verwendet, um den Zustand Ihrer migrierten Anwendung nach der Bereitstellung zu überprüfen.

Nach der Migration können Sie den eb health Befehl verwenden, um den Status Ihrer Umgebung zu überprüfen und Probleme zu identifizieren, die behoben werden müssen.

HealthD

Ein Monitoring-Agent in Elastic Beanstalk, der Metriken sammelt, Logs überwacht und den Integritätsstatus für EC2-Instances in einer Umgebung meldet. HealthD bietet erweiterte Statusberichte für Ihre migrierten Anwendungen.

Nach der Migration überwacht HealthD die Leistung Ihrer Anwendung, die Ressourcenauslastung und die Erfolgsquoten von Anfragen und bietet so einen umfassenden Überblick über den Zustand Ihrer Umgebung.

Protokolle bündeln

Eine Funktion in Elastic Beanstalk, die Protokolle von EC2-Instances komprimiert und zur zentralen Speicherung und Analyse auf Amazon S3 hochlädt. Diese Funktion hilft Ihnen bei der Behebung von Problemen mit Ihren migrierten Anwendungen.

Nach der Migration können Sie den eb logs Befehl verwenden, um Protokolle aus Ihrer Umgebung abzurufen und anzuzeigen.

aws-windows-deployment-manifest.json

Eine Datei, die den Inhalt, die Abhängigkeiten und die Konfiguration eines Softwarepakets oder einer Anwendung beschreibt. Dieses Manifest wird während des Migrationsprozesses generiert, um zu definieren, wie Ihre IIS-Anwendungen auf Elastic Beanstalk bereitgestellt werden sollen.

Abschnitt „benutzerdefiniertes Manifest“

Ein Abschnitt innerhalb `aws-windows-deployment-manifest.json` dieses Bereichs ermöglicht die benutzerdefinierte Steuerung der Anwendungsbereitstellung. Dieser Abschnitt enthält PowerShell Skripts und Befehle, die während des Bereitstellungsprozesses ausgeführt werden.

Während der Migration werden benutzerdefinierte Manifestabschnitte generiert, um bestimmte Aspekte Ihrer IIS-Konfiguration zu behandeln, z. B. die Einrichtung virtueller Verzeichnisse, die Rechteverwaltung und die Konfiguration des Anwendungspools.

EB CLI

Ein Befehlszeilentool, das Befehle für die Erstellung, Konfiguration und Verwaltung von Elastic Beanstalk Beanstalk-Anwendungen und -Umgebungen bereitstellt. Die EB-CLI enthält den eb migrate Befehl speziell für die Migration von IIS-Anwendungen zu Elastic Beanstalk.

Nach der Migration können Sie die EB CLI weiterhin verwenden, um Ihre Umgebung zu verwalten, Updates bereitzustellen, den Zustand zu überwachen und andere administrative Aufgaben auszuführen.

Optionseinstellungen

Konfigurationswerte, die definieren, wie Elastic Beanstalk AWS Ressourcen in Ihrer Umgebung bereitstellt und konfiguriert. Die Optionseinstellungen sind in Namespaces organisiert, die verschiedene Komponenten Ihrer Umgebung repräsentieren, z. B. Load Balancer, Instances und Umgebungsprozesse.

Während der Migration generiert der eb migrate Befehl die entsprechenden Optionseinstellungen auf der Grundlage Ihrer IIS-Konfiguration, um sicherzustellen, dass Ihre Cloud-Umgebung den Funktionen Ihrer Quellumgebung entspricht. Weitere Informationen finden Sie unter Konfigurationsoptionen im Elastic Beanstalk Developer Guide.

aws:elbv2:listener:default

Ein Elastic Beanstalk Beanstalk-Konfigurationsnamespace für den Standard-Listener auf einem Application Load Balancer. Während der Migration wird dieser Namespace auf der Grundlage Ihrer IIS-Site-Bindungen konfiguriert, um eine korrekte Weiterleitung des Datenverkehrs sicherzustellen.

Der Standard-Listener verarbeitet in der Regel den HTTP-Verkehr auf Port 80, der dann gemäß den Routingregeln an Ihre Anwendungsinstanzen weitergeleitet wird.

aws:elbv2:listener:listener_port

Ein Elastic Beanstalk Beanstalk-Konfigurationsnamespace für einen bestimmten Listener-Port auf einem Application Load Balancer. Dieser Namespace wird verwendet, um zusätzliche Listener für Ihre migrierten Anwendungen zu konfigurieren, z. B. HTTPS auf Port 443.

Während der Migration werden Listener auf der Grundlage der Portbindungen Ihrer IIS-Sites erstellt, wodurch sichergestellt wird, dass Ihre Anwendungen über dieselben Ports wie in Ihrer Quellumgebung zugänglich bleiben.

aws:elbv2:listenerrule:rule_name

Ein Elastic Beanstalk Beanstalk-Konfigurationsnamespace zur Definition von Routing-Regeln für einen Application Load Balancer Balancer-Listener. Diese Regeln legen anhand von Pfadmustern oder Host-Headern fest, wie eingehende Anfragen an verschiedene Zielgruppen weitergeleitet werden.

Während der Migration werden Listener-Regeln so erstellt, dass sie der URL-Struktur Ihrer IIS-Anwendungen entsprechen und sicherstellen, dass Anfragen an die richtigen Anwendungspfade weitergeleitet werden.

aws:elasticbeanstalk:environment:process:default

Ein Elastic Beanstalk Beanstalk-Konfigurationsnamespace für den Standardprozess in einer Umgebung. Dieser Namespace definiert, wie der standardmäßige Webanwendungsprozess konfiguriert ist, einschließlich der Einstellungen für die Integritätsprüfung, Portzuordnungen und Proxyeinstellungen.

Während der Migration wird der Standardprozess auf der Grundlage der Einstellungen Ihrer primären IIS-Website konfiguriert, sodass eine ordnungsgemäße Zustandsüberwachung und Bearbeitung von Anfragen gewährleistet ist.

aws:elasticbeanstalk:environment:process:process_name

Ein Elastic Beanstalk Beanstalk-Konfigurationsnamespace für einen bestimmten benannten Prozess in einer Umgebung. Dieser Namespace ermöglicht es Ihnen, mehrere Prozesse mit unterschiedlichen Konfigurationen zu definieren, ähnlich wie bei mehreren Anwendungspools in IIS.

Während der Migration können zusätzliche Prozesse erstellt werden, die unterschiedliche Site-Bindungen aus Ihrer Quellumgebung repräsentieren.

> **Note**
>
> Weitere Informationen zu einigen der in diesem Thema beschriebenen Begriffe finden Sie in den folgenden Ressourcen:
>
> - Elastic Beanstalk API-Aktionen —AWS Elastic Beanstalk API-Referenz
> - Elastic Beanstalk-Plattformen, einschließlich unterstützter Plattformversionen — Unterstützte Plattformen im AWS Elastic Beanstalk Plattformhandbuch
> - Namespaces für die Elastic Beanstalk Beanstalk-Konfiguration — in diesem Handbuch Allgemeine Optionen für alle Umgebungen
> - Die EB-CLI oder bestimmte EB-CLI-Befehle — Einrichtung der EB-Befehlszeilenschnittstelle (EB CLI) zur Verwaltung von Elastic Beanstalk in diesem Handbuch

Python-Begriffe

pip

Das Paketinstallationsprogramm für Python, das zur Installation und Verwaltung von in Python geschriebenen Softwarepaketen verwendet wird. Die EB-CLI wird mit pip installiert und aktualisiert.

Während des Migrationsprozesses wird pip verwendet, um das EB-CLI-Paket und seine Abhängigkeiten auf Ihrem Quellserver zu installieren und die für die Migration erforderlichen Tools bereitzustellen.

PyPI

Python Package Index, das offizielle Repository für Python-Softwarepakete von Drittanbietern, aus dem pip Pakete abruft und installiert. Die EB-CLI und ihre Abhängigkeiten werden auf PyPI gehostet.

Bei der Installation der EB-CLI für die Migration stellt pip eine Verbindung zu PyPI her, um die erforderlichen Pakete herunterzuladen und zu installieren.

virtualenv

Ein Tool zum Erstellen isolierter Python-Umgebungen, das es verschiedenen Projekten ermöglicht, ihre eigenen Abhängigkeiten und Pakete ohne Konflikte zu haben. Die Verwendung von virtualenv wird bei der Installation der EB-CLI empfohlen, um Konflikte mit anderen Python-Anwendungen zu vermeiden.

Durch das Erstellen einer virtuellen Umgebung vor der Installation der EB CLI wird sichergestellt, dass die Migrationstools über eine saubere, isolierte Umgebung mit den richtigen Abhängigkeiten verfügen.

pywin32

Eine Reihe von Python-Erweiterungen, die den Zugriff auf viele APIs Windows-Betriebssysteme ermöglichen und so die Interaktion mit dem Windows-Betriebssystem und seinen Komponenten ermöglichen. Die EB-CLI verwendet pywin32, um während der Migration mit Windows-spezifischen Funktionen zu interagieren.

Während des Migrationsprozesses wird pywin32 verwendet, um auf die IIS-Konfiguration, die Windows-Registrierungseinstellungen und andere Systeminformationen zuzugreifen, die für das ordnungsgemäße Verpacken und Migrieren Ihrer Anwendungen erforderlich sind.

python.net

Ein Paket, das Python-Code die Interaktion mit .NET Framework- und .NET Core-Anwendungen ermöglicht. Diese Integration ermöglicht es der EB CLI, während des Migrationsprozesses mit .NET-Komponenten zu arbeiten.

Der Migrationsprozess verwendet möglicherweise Pythonnet, um mit .NET-Assemblys und -Komponenten zu interagieren, wenn Sie Ihre Anwendungen für die Bereitstellung in Elastic Beanstalk analysieren und verpacken.

Durchführen grundlegender IIS-Migrationen

Dieser Abschnitt führt Sie durch den Prozess der Migration Ihrer IIS-Anwendungen zu Elastic Beanstalk mithilfe des Befehls. eb migrate

Erkunden Sie Ihre IIS-Umgebung

Bevor Sie Änderungen vornehmen, sollten Sie wissen, welche Ressourcen auf Ihrem Server vorhanden sind. Erkunden Sie zunächst Ihre IIS-Siteseb migrate explore, indem Sie Folgendes ausführen, wie im folgenden Beispiel gezeigt:

```
PS C:\migrations_workspace> eb migrate explore
```

Mit diesem Befehl werden Ihre IIS-Sites angezeigt. Sehen Sie sich die folgende Liste an:

```
Default Web Site
Intranet
API.Internal
Reports
```

Für eine detaillierte Ansicht der Konfiguration der einzelnen Sites, einschließlich Bindungen, Anwendungen und virtuellen Verzeichnissen, fügen Sie die `--verbose` Option hinzu, wie in diesem Beispiel gezeigt:

```
PS C:\migrations_workspace> eb migrate explore --verbose
```

Die folgende Liste enthält die umfassenden Informationen über Ihre Umgebung, die der Befehl bereitstellt:

```
1: Default Web Site:
  - Bindings:
    - *:80:www.example.com
    - *:443:www.example.com
  - Application '/':
    - Application Pool: DefaultAppPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\inetpub\wwwroot
        - Logon Method: ClearText
```

```
  - Application '/api':
    - Application Pool: ApiPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\websites\api
        - Logon Method: ClearText
2: Intranet:
...
3. API.Internal:
...
4. Reports:
...
```

Die Discovery-Ausgabe verstehen

Die ausführliche Ausgabe enthält die folgenden wichtigen Informationen für die Migrationsplanung:

Sites

In der Discovery-Ausgabe werden alle IIS-Sites auf Ihrem Server aufgeführt. Jede Site wird anhand ihres Namens (z. B. „Standardwebsite“, „Intranet“, „API.Internal“) identifiziert und fortlaufend nummeriert. Wenn auf einem Server mehrere Sites vorhanden sind, kann der `eb migrate` Befehl je nach Ihrer Migrationsstrategie jede Site einzeln oder zusammen paketieren und bereitstellen.

Bindungen

Protokollbindungen geben Aufschluss darüber, welche Protokolle (HTTP/HTTPS) Ihre Websites verwenden und auf welchen Ports sie betrieben werden. Die Bindungsinformationen beinhalten Host-Header-Anforderungen, die domänenbasierte Routing-Konfigurationen definieren.

Anwendungen

Anwendungspfade zeigen sowohl Stamm- als auch verschachtelte Anwendungsstrukturen in Ihrer IIS-Konfiguration. Anwendungspoolzuweisungen geben an, wie Ihre Anwendungen aus Sicherheits- und Ressourcenmanagementgründen voneinander isoliert sind.

Virtuelle Verzeichnisse

Physikalische Pfadzuordnungen geben an, wo sich Ihre Inhalte im Dateisystem befinden. Die Authentifizierungseinstellungen weisen spezielle Zugriffsanforderungen auf, die auch nach der Migration beibehalten werden müssen.

Vorbereitung der Migration

Machen Sie sich mit Ihrer Umgebung vertraut und stellen Sie sicher, dass Ihr Server die Voraussetzungen erfüllt. Überprüfen Sie zunächst Ihre IIS-Version mit dem folgenden PowerShell Befehl:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\InetStp\" -Name
 MajorVersion
```

Sie benötigen IIS 7.0 oder höher. Das Migrationstool verwendet Web Deploy 3.6 zum Verpacken Ihrer Anwendungen. Überprüfen Sie die Installation mit dem folgenden Befehl:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\IIS Extensions
\MSDeploy\3" -Name InstallPath
```

Wenn Web Deploy nicht auf Ihrem Server installiert ist, können Sie es von der Downloadseite des Microsoft Web Platform Installers herunterladen.

Ihre erste Migration

Beginnen wir mit einer grundlegenden Migration der Standardwebsite. Das folgende Beispiel zeigt den einfachsten Befehl,eb migrate.

```
PS C:\migrations_workspace> eb migrate
```

Dieser Befehl initiiert eine Reihe automatisierter Schritte, die in der folgenden Beispielausgabe dargestellt werden:

```
Identifying VPC configuration of this EC2 instance (i-0123456789abcdef0)
  id: vpc-1234567890abcdef0
  publicip: true
  elbscheme: public
  ec2subnets: subnet-123,subnet-456,subnet-789
  securitygroups: sg-123,sg-456
  elbsubnets: subnet-123,subnet-456,subnet-789

Using .\migrations\latest to contain artifacts for this migration run.
Generating source bundle for sites, applications, and virtual directories...
  Default Web Site/ -> .\migrations\latest\upload_target\DefaultWebSite.zip
```

Das Migrationstool erstellt ein strukturiertes Verzeichnis, das Ihre Bereitstellungsartefakte enthält. Die folgende Liste zeigt die Verzeichnisstruktur:

```
C:\migration_workspace\
### .\migrations\latest\
    ### upload_target\
        ### DefaultWebSite.zip
        ### aws-windows-deployment-manifest.json
        ### ebmigrateScripts\
            ### site_installer.ps1
            ### permission_handler.ps1
            ### >other helper scripts<
```

Steuerung der Migration

Um mehr Kontrolle über den Migrationsprozess zu erhalten, können Sie mit dem folgenden Befehl genau angeben, welche Websites migriert werden sollen:

```
PS C:\migrations_workspace> eb migrate --sites "Default Web Site,Intranet"
```

Sie können auch den Umgebungsnamen und den Anwendungsnamen anpassen, wie im folgenden Beispielbefehl gezeigt:

```
PS C:\migrations_workspace> eb migrate `
    --sites "Default Web Site" `
    --application-name "CorporateApp" `
    --environment-name "Production"
```

Eine vollständige Liste der Optionen finden Sie unterthe section called “eb migrate”.

Überwachen des Fortschritts

eb migrateStellt während der Migration Statusaktualisierungen in Echtzeit bereit. Sehen Sie sich das folgende Ausgabebeispiel an:

```
...
Creating application version
Creating environment... This may take a few minutes

2024-03-18 18:12:15    INFO    Environment details for: Production
  Application name: CorporateApp
```

```
  Region: us-west-2
  Deployed Version: app-230320_153045
  Environment ID: e-abcdef1234
  Platform: 64bit Windows Server 2019 v2.7.0 running IIS 10.0
  Tier: WebServer-Standard-1.0
  CNAME: production.us-west-2.elasticbeanstalk.com
  Updated: 2024-03-20 15:30:45
2025-03-18 18:12:17    INFO    createEnvironment is starting.
2025-03-18 18:12:19    INFO    Using elasticbeanstalk-us-east-1-180301529717 as Amazon
 S3 storage bucket for environment data.
2025-03-18 18:12:40    INFO    Created security group named: sg-0fdd4d696a26b086a
2025-03-18 18:12:48    INFO    Environment health has transitioned to Pending.
 Initialization in progress (running for 7 seconds). There are no instances.
...
2025-03-18 18:23:59    INFO    Application available at EBMigratedEnv-arrreal3.us-
east-1.elasticbeanstalk.com.
2025-03-18 18:24:00    INFO    Successfully launched environment: EBMigratedEnv-
arrreal3
```

Überprüfung der Migration

Sobald die Umgebung bereit ist, bietet Elastic Beanstalk mehrere Möglichkeiten, Ihr Deployment zu verifizieren.

Greifen Sie auf Ihre Anwendung zu

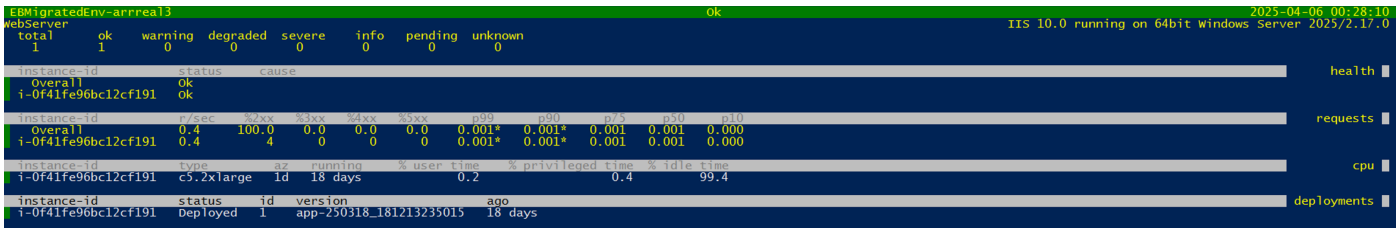
Öffnen Sie Ihre Anwendungs-URL (CNAME) in einem Webbrowser, um zu überprüfen, ob sie ordnungsgemäß funktioniert.

Überprüfen Sie den Zustand der Umgebung

Verwenden Sie den eb health Befehl, um den Zustand Ihrer Umgebung zu überprüfen.

```
PS C:\migrations_workspace> eb health
```

Das folgende Bildschirmbild zeigt den Zustand der Instanz, die Kennzahlen zur Reaktion der Anwendung und die Auslastung der Systemressourcen.



Verwenden Sie den eb logs Befehl, um auf Protokolle zuzugreifen und Probleme zu beheben:

```
PS C:\migrations_workspace> eb logs --zip
```

Der eb logs Befehl lädt Protokolle in das `.elasticbeanstalk/logs` Verzeichnis herunter. Weitere Informationen finden Sie unter the section called “CloudWatch Logs”.

Connect zu Instanzen her

Wenn Sie während der Migration ein key pair angegeben haben, können Sie zur direkten Fehlerbehebung über RDP eine Verbindung zu Ihren Instances herstellen.

Greifen Sie auf die Elastic Beanstalk Beanstalk-Konsole zu

Sie können den Zustand, die Protokolle und die Konfigurationseigenschaften der Umgebung über die Umgebungsmanagementkonsole für diese Umgebung einsehen.

Verwaltung von Migrationsartefakten

Der eb migrate Befehl erstellt während des Migrationsprozesses lokale Artefakte. Diese Artefakte enthalten vertrauliche Informationen und können im Laufe der Zeit viel Speicherplatz beanspruchen. Verwenden Sie den cleanup Unterbefehl, um diese Artefakte zu verwalten, wie im folgenden Beispiel gezeigt:

```
PS C:\migrations_workspace> eb migrate cleanup
Are you sure you would like to cleanup older artifacts within ./migrations/? (Y/N):
```

Um die Bereinigung ohne Bestätigung zu erzwingen, verwenden Sie die --force folgende Option:

```
PS C:\migrations_workspace> eb migrate cleanup --force
```

Beim Bereinigungsprozess wird die letzte erfolgreiche Migration im Verzeichnis beibehalten und ältere `./migrations/latest` Migrationsverzeichnisse entfernt

Netzwerkkonfiguration und Porteinstellungen

In diesem Abschnitt werden Netzwerkkonfigurationsoptionen für IIS-Migrationen behandelt, einschließlich VPC-Einstellungen, Portkonfigurationen und Bereitstellungen an mehreren Standorten.

VPC-Konfiguration

Der eb migrate Befehl bietet flexible VPC-Konfigurationsoptionen für Ihre Elastic Beanstalk Beanstalk-Umgebung. Das Tool kann entweder VPC-Einstellungen von einer EC2-Quellinstanz erkennen oder benutzerdefinierte VPC-Konfigurationen über Befehlszeilenparameter akzeptieren. Lesen SieVerwenden von Elastic Beanstalk mit Amazon VPC, um zu erfahren, wie Elastic Beanstalk mit VPC konfiguriert wird.

Automatische VPC-Erkennung

Wenn sie auf einer EC2-Instance eb migrate ausgeführt wird, erkennt und verwendet sie automatisch die VPC-Konfiguration aus den EC2-Instances der Quellumgebung. Die folgende Beispielausgabe veranschaulicht die von ihr erkannten Konfigurationsinformationen:

```
PS C:\migrations_workspace > eb migrate
Identifying VPC configuration of this EC2 instance (i-0123456789abcdef0):
  id: vpc-1234567890abcdef0
  publicip: true
  elbscheme: public
  ec2subnets: subnet-123,subnet-456,subnet-789
  securitygroups: sg-123,sg-456
  elbsubnets: subnet-123,subnet-456,subnet-789
...
```

Die erkannte Konfiguration umfasst:

- VPC-ID
- Einstellungen für die öffentliche IP-Zuweisung
- Load Balancer-Schema (öffentlich/privat)
- Subnetzzuweisungen für EC2-Instances
- Zuordnungen von Sicherheitsgruppen
- Subnetzzuweisungen für den Load Balancer

Lokale oder nicht in der Cloud befindliche Hosts AWS

Wenn der Elastic Beanstalk-Service auf einem lokalen Server oder einem AWS Nicht-Cloud-Host eb migrate ausgeführt wird, verwendet er die Standard-VPC in Ihrem Konto. AWS Die folgende Liste zeigt ein Beispiel für einen Befehl und eine Ausgabe:

```
PS C:\migrations_worspace> eb migrate `
      -k windows-test-pem `
      --region us-east-1 `
      -a EBMigratedEnv `
      -e EBMigratedEnv-test2 `
      --copy-firewall-config
Determining EB platform based on host machine properties
Using .\migrations\latest to contain artifacts for this migration run.
...
```

Lesen Sie Verwenden von Elastic Beanstalk mit Amazon VPC weiter, um zu erfahren, wie Elastic Beanstalk die Standard-VPC für Ihre Umgebung konfiguriert.

Benutzerdefinierte VPC-Konfiguration

Stellen Sie für jede Quellumgebung (EC2, lokal oder nicht in der AWS Cloud), in der Sie bestimmte VPC-Einstellungen benötigen, eine VPC-Konfigurationsdatei wie die im folgenden Beispiel bereit:

```
{
    "id": "vpc-12345678",
    "publicip": "true",
    "elbscheme": "public",
    "ec2subnets": ["subnet-a1b2c3d4", "subnet-e5f6g7h8"],
    "securitygroups": "sg-123456,sg-789012",
    "elbsubnets": ["subnet-a1b2c3d4", "subnet-e5f6g7h8"]
}
```

Wenden Sie diese Konfiguration mit dem folgenden Befehl an:

```
PS C:\migrations_workspace> eb migrate --vpc-config vpc-config.json
```

> **Note**
>
> Die VPC-Konfigurationsdatei erfordert das `id` Feld, das die VPC-ID angibt. Alle anderen Felder sind optional und Elastic Beanstalk verwendet Standardwerte für alle Felder, die Sie nicht angeben.

> **Important**
>
> Bei der Migration werden alle vorhandenen VPC-Einstellungen aus der Quellumgebung ignoriert, wenn Sie den *`--vpc-config`* Parameter angeben. Wenn Sie diesen Parameter verwenden, verwendet die Migration nur die VPC-Einstellungen, die in der Konfigurationsdatei angegeben sind, die Sie übergeben. Die Verwendung dieses Parameters setzt das Standardverhalten beim Erkennen der VPC-Konfiguration der Quell-Instance oder beim Verwenden der Standard-VPC außer Kraft.

Verwenden Sie den `--vpc-config` Parameter in diesen Szenarien:

- Wenn Sie Nicht-EC2-Umgebungen migrieren, die keine erkennbaren VPC-Einstellungen haben
- Wenn Sie zu einer anderen VPC migrieren als der, die von der Quellumgebung verwendet wird
- Wenn Sie Subnetzauswahlen oder Sicherheitsgruppenkonfigurationen anpassen müssen
- Wenn die automatische Erkennung die gewünschten VPC-Einstellungen nicht korrekt identifiziert
- Wenn Sie von einer lokalen Umgebung migrieren und nicht die Standard-VPC verwenden möchten

Konfiguration der Netzwerksicherheit

eb migrateÖffnet standardmäßig Port 80 auf Zielinstanzen, kopiert jedoch keine anderen Windows-Firewallregeln vom Quellcomputer. Verwenden Sie den folgenden Befehl, um alle Firewallkonfigurationen einzubeziehen:

```
PS C:\migrations_workspace> eb migrate --copy-firewall-config
```

Dieser Befehl führt die folgenden Aktionen aus:

- Identifiziert Ports, die von IIS-Site-Bindungen verwendet werden
- Ruft die entsprechenden Firewallregeln ab
- Generiert PowerShell Skripts, um Regeln auf Zielinstanzen neu zu erstellen
- Behält alle DENY-Regeln für Port 80 vom Quellcomputer bei (andernfalls ist Port 80 standardmäßig zulässig)

Stellen Sie sich einen Anwendungsfall vor, bei dem Ihr Quellcomputer über die im folgenden Beispiel angegebenen Firewallregeln verfügt:

```
# Source machine firewall configuration
Get-NetFirewallRule | Where-Object {$_.Enabled -eq 'True'} | Get-NetFirewallPortFilter
 | Where-Object {$_.LocalPort -eq 80 -or $_.LocalPort -eq 443 -or $_.LocalPort -eq
 8081}
# Output shows rules for ports 80, 443, and 8081
```

Bei der Migration wird ein Skript (`modify_firewall_config.ps1`) erstellt, das die folgende Konfiguration enthält:

```
New-NetFirewallRule -DisplayName "Allow Web Traffic" -Direction Inbound -Action Allow -
Protocol TCP -LocalPort 80,443
New-NetFirewallRule -DisplayName "Allow API Traffic" -Direction Inbound -Action Allow -
Protocol TCP -LocalPort 8081
```

Das Migrationstool führt automatisch die folgenden Aktionen aus:

- Extrahiert HTTP/HTTPS Ports aus allen IIS-Site-Bindungen
- Verwendet die Windows-Firewall INetFwPolicy2-Schnittstelle, um Firewallregeln aufzulisten
- Filtert Regeln so, dass sie nur Regeln einschließen, die explizit auf die angegebenen Ports verweisen
- Verarbeitet nur HTTP- und HTTPS-Site-Bindungen und die zugehörigen Firewallregeln
- Behält die Regeleigenschaften wie Anzeigename, Aktion, Protokoll und Aktivierungsstatus bei
- Behandelt sowohl einzelne Ports als auch Portbereiche in Firewallregeln
- Fügt das Firewall-Konfigurationsskript zum Bereitstellungsmanifest hinzu

Load Balancer-Konfiguration

Sie können die Load Balancer Balancer-Konfiguration über das `--vpc-config` Argument angeben. Die folgenden Beispiele veranschaulichen die Parameter.

Auswahl des Schemas

Wählen Sie zwischen öffentlichen und privaten Load Balancer-Schemata:

```
{
    "id": "vpc-12345678",
    "elbscheme": "private",
    "elbsubnets": ["subnet-private1", "subnet-private2"]
```

```
}
```

Verteilung über Subnetze

Um eine hohe Verfügbarkeit zu gewährleisten, verteilen Sie die Load Balancer-Subnetze auf die Availability Zones:

```
{
    "elbsubnets": [
        "subnet-az1", // Availability Zone 1
        "subnet-az2", // Availability Zone 2
        "subnet-az3"  // Availability Zone 3
    ]
}
```

> **Note**
>
> Elastic Beanstalk unterstützt zwar die Erstellung von Umgebungen mit Application Load Balancers, Network Load Balancers und Classic Load Balancers, der eb migrate Befehl unterstützt jedoch nur Application Load Balancers. Weitere Informationen zu Load-Balancer-Typen finden Sie unter Load Balancer Ihrer Elastic-Beanstalk-Umgebung.

Bereitstellungen an mehreren Standorten mit Portkonfigurationen

Der eb migrate Befehl verarbeitet komplexe IIS-Bereitstellungen an mehreren Standorten, bei denen Anwendungen möglicherweise Abhängigkeiten gemeinsam haben oder nicht standardmäßige Ports verwenden. Betrachten Sie das folgende Beispiel für eine typische Unternehmenseinrichtung mit mehreren Standorten:

```
<!-- IIS Configuration -->
<sites>
    <site name="Default Web Site" id="1">
        <bindings>
            <binding protocol="http" bindingInformation="*:80:www.example.com" />
        </bindings>
    </site>
    <site name="InternalAPI" id="2">
        <bindings>
```

```
            <binding protocol="http" bindingInformation="*:8081:api.internal" />
        </bindings>
    </site>
    <site name="ReportingPortal" id="3">
        <bindings>
            <binding protocol="http" bindingInformation="*:8082:reports.internal" />
        </bindings>
    </site>
</sites>
```

Verwenden Sie den folgenden Beispielbefehl und die folgenden Parameter, um diese Konfiguration zu migrieren:

```
PS C:\migrations_workspace> eb migrate `
    --sites "Default Web Site,InternalAPI,ReportingPortal" `
    --copy-firewall-config `
    --instance-type "c5.large"
```

Der eb migrate Befehl erstellt ein Bereitstellungspaket, das die Identität und Konfiguration der einzelnen Standorte beibehält. Der Befehl generiert eine`aws-windows-deployment-manifest.json`, die definiert, wie diese Sites bereitgestellt werden sollen. Das folgende Beispiel zeigt eine generierte JSON-Datei:

```
{
    "manifestVersion": 1,
    "deployments": {
        "msDeploy": [
            {
                "name": "DefaultWebSite",
                "parameters": {
                    "appBundle": "DefaultWebSite.zip",
                    "iisPath": "/",
                    "iisWebSite": "Default Web Site"
                }
            }
        ],
        "custom": [
            {
                "name": "InternalAPI",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\\install_site_InternalAPI.ps1"
```

```
                    },
                    "restart": {
                        "file": "ebmigrateScripts\\restart_site_InternalAPI.ps1"
                    },
                    "uninstall": {
                        "file": "ebmigrateScripts\\uninstall_site_InternalAPI.ps1"
                    }
                }
            },
            {
                "name": "ReportingPortal",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\\install_site_ReportingPortal.ps1"
                    },
                    "restart": {
                        "file": "ebmigrateScripts\\restart_site_ReportingPortal.ps1"
                    },
                    "uninstall": {
                        "file": "ebmigrateScripts\\uninstall_site_ReportingPortal.ps1"
                    }
                }
            }
        ]
    }
}
```

Der Migrationsprozess erstellt die folgenden Application Load Balancer Balancer-Listener-Regeln, die Ihre ursprüngliche Routing-Logik beibehalten:

- Der Verkehr über Port 80 leitet zur Standardwebsite weiter
- Port 8081 leitet den Verkehr zur InternalAPI weiter
- Port 8082 Verkehrswege zu ReportingPortal

Gemeinsame Konfiguration und Abhängigkeiten

Wenn Websites Konfigurationen oder Abhängigkeiten gemeinsam nutzen, werden diese Beziehungen angemessen eb migrate behandelt. Sehen Sie sich das folgende Beispiel an, in dem mehrere Standorte eine gemeinsame Konfiguration verwenden:

```
<!-- Shared configuration in applicationHost.config -->
```

```
<location path="Default Web Site">
    <system.webServer>
        <asp enableSessionState="true" />
        <caching enabled="true" enableKernelCache="true" />
    </system.webServer>
</location>
```

Der Migrationsprozess umfasst die folgenden Aufgaben:

1. Identifiziert gemeinsam genutzte Konfigurationen an mehreren Standorten
2. Generiert entsprechende PowerShell Skripts, um diese Einstellungen anzuwenden
3. Behält die Konfigurationshierarchie und Verer

Best Practices

Wir empfehlen, dass Sie die bewährten Methoden für die Netzwerkkonfiguration Ihrer migrierten Anwendung befolgen. Die folgenden Gruppierungen bieten zusammenfassende Richtlinien.

VPC-Design

- Folgen Sie den Best AWS Practices für VPC-Design
- Verwenden Sie separate Subnetze für Load Balancer und EC2-Instances
- Implementieren Sie die richtigen Routentabellen und NACLs
- Ziehen Sie VPC-Endpunkte für Dienste in Betracht AWS

Hohe Verfügbarkeit

- Bereitstellen über mehrere Availability Zones hinweg
- Verwenden Sie mindestens zwei Subnetze für Load Balancer
- Konfigurieren Sie die auto-scaling für AZs
- Implementieren Sie angemessene Gesundheitschecks

Sicherheit

- Befolgen Sie die bewährten Sicherheitsmethoden
- Verwenden Sie Sicherheitsgruppen als primäre Zugriffskontrolle
- Implementieren Sie Netzwerk-Zugriffskontrolllisten (ACLs) für zusätzliche Sicherheit
- VPC-Flow-Logs überwachen

Fehlerbehebung

Zu den häufigsten Problemen mit der Netzwerkkonfiguration gehören die folgenden Bereiche. Nach jedem Thema finden Sie Beispielbefehle, mit denen Sie weitere Informationen zur Netzwerkkonfiguration und zum Zustand Ihrer Umgebung erhalten können.

Konfiguration des Subnetzes

```
# Verify subnet availability
PS C:\migrations_workspace> aws ec2 describe-subnets --subnet-ids subnet-id

# Check available IP addresses
PS C:\migrations_workspace>aws ec2 describe-subnets --subnet-ids subnet-id --query
 'Subnets[].AvailableIpAddressCount'
```

Zugriff auf Sicherheitsgruppen

```
# Verify security group rules
PS C:\migrations_workspace> aws ec2 describe-security-groups --group-ids sg-id

# Test network connectivity
PS C:\migrations_workspace> aws ec2 describe-network-interfaces --filters
 Name=group-id,Values=sg-id
```

Zustand des Load Balancers

```
# Check load balancer health
PS C:\migrations_workspace> aws elbv2 describe-target-health --target-group-arn
 arn:aws:elasticloadbalancing:region:account-id:targetgroup/group-name/group-id
```

Sicherheitskonfigurationen und IAM-Rollen

Der eb migrate Befehl verwaltet AWS Sicherheitskonfigurationen mithilfe von IAM-Rollen, Instanzprofilen und Servicerollen. Das Verständnis dieser Komponenten gewährleistet eine angemessene Zugriffskontrolle und Einhaltung der Sicherheitsbestimmungen während der Migration.

Konfiguration des Instanzprofils

Ein Instance-Profil dient als Container für eine IAM-Rolle, die Elastic Beanstalk an EC2-Instances in Ihrer Umgebung anhängt. Bei der Ausführung können Sie eb migrate ein benutzerdefiniertes Instance-Profil angeben:

```
PS C:\migrations_workspace> eb migrate --instance-profile "CustomInstanceProfile"
```

Wenn Sie kein Instanzprofil angeben, eb migrate erstellt ein Standardprofil mit diesen Berechtigungen:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-*",
                "arn:aws:s3:::elasticbeanstalk-*/*"
            ]
        }
    ]
}
```

Verwaltung von Servicerollen

Eine Servicerolle ermöglicht es Elastic Beanstalk, AWS Ressourcen in Ihrem Namen zu verwalten. Geben Sie während der Migration mit dem folgenden Befehl eine benutzerdefinierte Servicerolle an:

```
PS C:\migrations_workspace> eb migrate --service-role "CustomServiceRole"
```

Falls nicht angegeben, eb migrate wird eine Standard-Servicerolle `aws-elasticbeanstalk-service-role` mit einer Vertrauensrichtlinie erstellt, die es Elastic Beanstalk ermöglicht, die Rolle zu übernehmen. Diese Servicerolle ist für Elastic Beanstalk unverzichtbar, um den Zustand Ihrer Umgebung zu überwachen und verwaltete Plattform-Updates durchzuführen. Für die Servicerolle sind zwei verwaltete Richtlinien erforderlich:

- `AWSElasticBeanstalkEnhancedHealth`- Ermöglicht Elastic Beanstalk die Überwachung des Instance- und Umgebungszustands mithilfe des erweiterten Zustandsberichtssystems
- `AWSElasticBeanstalkManagedUpdates`- Ermöglicht Elastic Beanstalk, verwaltete Plattform-Updates durchzuführen, einschließlich der Aktualisierung von Umgebungsressourcen, wenn eine neue Plattformversion verfügbar ist

Mit diesen Richtlinien verfügt die Servicerolle über folgende Berechtigungen:

- Auto Scaling Scaling-Gruppen erstellen und verwalten
- Application Load Balancer erstellen und verwalten
- Logs auf Amazon hochladen CloudWatch
- EC2-Instances verwalten

Weitere Informationen zu Servicerollen finden Sie Elastic Beanstalk-Servicerolle im Elastic Beanstalk Developer Guide.

Sicherheitsgruppenkonfiguration

Der eb migrate Befehl konfiguriert automatisch Sicherheitsgruppen auf der Grundlage Ihrer IIS-Site-Bindungen. Wenn Ihre Quellumgebung beispielsweise Websites enthält, die die Ports 80, 443 und 8081 verwenden, ergibt sich die folgende Konfiguration:

```
<site name="Default Web Site">
    <bindings>
        <binding protocol="http" bindingInformation="*:80:" />
        <binding protocol="https" bindingInformation="*:443:" />
    </bindings>
</site>
<site name="InternalAPI">
    <bindings>
        <binding protocol="http" bindingInformation="*:8081:" />
    </bindings>
```

```
</site>
```

Der Migrationsprozess schließt die folgenden Aktionen ab:

- Erstellt eine Load Balancer-Sicherheitsgruppe, die eingehenden Datenverkehr über die Ports 80 und 443 aus dem Internet ermöglicht (0.0.0.0/0)
- Erstellt eine EC2-Sicherheitsgruppe, die den Datenverkehr vom Load Balancer zulässt
- Konfiguriert zusätzliche Ports (wie 8081), falls angegeben `--copy-firewall-config`

Standardmäßig ist der Application Load Balancer für öffentlichen Zugriff über das Internet konfiguriert. Wenn Sie dieses Verhalten anpassen müssen, z. B. den Zugriff auf bestimmte IP-Bereiche einschränken oder einen privaten Load Balancer verwenden, können Sie die standardmäßige VPC- und Sicherheitsgruppenkonfiguration mit dem folgenden Parameter überschreiben: `--vpc-config`

```
PS C:\migrations_workspace> eb migrate --vpc-config vpc-config.json
```

Mit der folgenden `vpc-config.json` Konfiguration wird beispielsweise ein privater Load Balancer in einem privaten Subnetz erstellt:

```
{
    "id": "vpc-12345678",
    "publicip": "false",
    "elbscheme": "internal",
    "ec2subnets": ["subnet-private1", "subnet-private2"],
    "elbsubnets": ["subnet-private1", "subnet-private2"]
}
```

Weitere Informationen zu VPC-Konfigurationsoptionen finden Sie unterVPC-Konfiguration.

Integration von SSL-Zertifikaten

Integrieren Sie bei der Migration von Websites mit HTTPS-Bindungen SSL-Zertifikate über AWS Certificate Manager (ACM):

```
PS C:\migrations_workspace> eb migrate --ssl-certificates
 "arn:aws:acm:region:account:certificate/certificate-id"
```

Mit dieser Konfiguration werden die folgenden Aktionen abgeschlossen:

- Ordnet das Zertifikat dem Application Load Balancer zu
- Hält die HTTPS-Terminierung am Load Balancer aufrecht
- Behält die interne HTTP-Kommunikation zwischen dem Load Balancer und den EC2-Instances bei

Windows-Authentifizierung

Bei Anwendungen, die die Windows-Authentifizierung eb migrate verwenden, werden die Authentifizierungseinstellungen der Anwendung `web.config` wie folgt beibehalten:

```
<configuration>
    <system.webServer>
        <security>
            <authentication>
                <windowsAuthentication enabled="true">
                    <providers>
                        <add value="Negotiate" />
                        <add value="NTLM" />
                    </providers>
                </windowsAuthentication>
            </authentication>
        </security>
    </system.webServer>
</configuration>
```

> **Important**
>
> Der eb migrate Befehl kopiert keine Benutzerprofile oder Konten aus Ihrer Quellumgebung auf die Elastic Beanstalk Beanstalk-Zielinstanzen. Alle benutzerdefinierten Benutzerkonten oder Gruppen, die Sie auf Ihrem Quellserver erstellt haben, müssen nach der Migration in der Zielumgebung neu erstellt werden.

Integrierte Windows-Konten wie `IUSR` und Gruppen wie `IIS_IUSRS` sowie alle anderen integrierten Konten und Gruppen sind standardmäßig in den Windows Server-Zielinstanzen enthalten. Weitere Informationen zu integrierten IIS-Konten und -Gruppen finden Sie in der Microsoft-Dokumentation unter Grundlegendes zu integrierten Benutzer- und Gruppenkonten in IIS.

Wenn Ihre Anwendung auf benutzerdefinierten Windows-Benutzerkonten oder der Active Directory-Integration basiert, müssen Sie diese Aspekte nach Abschluss der Migration separat konfigurieren.

Bewährte Methoden und Problembehebung

Rollenverwaltung

Implementieren Sie bewährte AWS IAM-Praktiken bei der Rollenverwaltung für Ihre Elastic Beanstalk Beanstalk-Umgebungen:

Erstellung und Verwaltung von Rollen

- Erstellen Sie Rollen, sofern möglich, mithilfe AWS verwalteter Richtlinien
- Folgen Sie den Best Practices für IAM-Sicherheit
- Verwenden Sie den AWS Policy Generator für benutzerdefinierte Richtlinien
- Implementieren Sie Berechtigungsgrenzen für zusätzliche Sicherheit

Überwachung und Prüfung

Aktivieren Sie AWS CloudTrail diese Option, um die Rollennutzung zu überwachen:

- Folgen Sie dem AWS CloudTrail Benutzerhandbuch
- Konfigurieren Sie die CloudWatch Logs-Integration für die Echtzeitüberwachung
- Richten Sie Benachrichtigungen für nicht autorisierte API-Aufrufe ein

Regelmäßiger Überprüfungsprozess

Richten Sie einen vierteljährlichen Überprüfungszyklus ein, um die folgenden Aufgaben zu erledigen:

- Prüfen Sie ungenutzte Berechtigungen mit IAM Access Analyzer
- Entfernen Sie veraltete Berechtigungen
- Aktualisieren Sie Rollen auf der Grundlage der Prinzipien der geringsten Rechte

Verwaltung von Zertifikaten

Implementieren Sie diese Verfahren für SSL/TLS Zertifikate in Ihren Elastic Beanstalk Beanstalk-Umgebungen:

Lebenszyklus von Zertifikaten

- AWS Certificate ManagerFür die Zertifikatsverwaltung verwenden
- Aktivieren Sie die automatische Verlängerung für von ACM ausgestellte Zertifikate
- Richten Sie Ablaufbenachrichtigungen ein

Sicherheitsstandards

- Verwenden Sie TLS 1.2 oder höher
- Befolgen Sie die AWS Sicherheitsrichtlinien für HTTPS-Listener
- Implementieren Sie bei Bedarf HTTP Strict Transport Security (HSTS)

Verwaltung der Sicherheitsgruppe

Implementieren Sie die folgenden bewährten Methoden für Sicherheitsgruppen:

Verwaltung von Regeln

- Dokumentieren Sie alle Anforderungen an benutzerdefinierte Ports
- Verwenden Sie VPC Flow Logs, um den Verkehr zu überwachen
- Verwenden Sie nach Möglichkeit Referenzregeln für Sicherheitsgruppen anstelle von IP-Bereichen

Regelmäßige Prüfung

Richten Sie monatliche Überprüfungen ein, um die folgenden Aufgaben zu erledigen:

- Identifizieren und entfernen Sie ungenutzte Regeln
- Überprüfen Sie source/destination die Anforderungen
- Suchen Sie nach sich überschneidenden Regeln

Protokollierung und Überwachung

Für eine effektive Sicherheitsüberwachung konfigurieren Sie die folgenden Protokolle:

Windows-Ereignisprotokolle auf EC2-Instances

```
# Review Security event log
PS C:\migrations_workspace> Get-EventLog -LogName Security -Newest 50

# Check Application event log
PS C:\migrations_workspace> Get-EventLog -LogName Application -Source "IIS*"
```

CloudWatch Integration von Protokollen

Konfigurieren Sie den CloudWatch Logs-Agent so, dass er Windows-Ereignisprotokolle CloudWatch zur zentralen Überwachung und Alarmierung streamt.

Bei anhaltenden Problemen sammeln Sie diese Protokolle und wenden Sie sich AWS Support mit den folgenden Informationen an:

- Umgebungs-ID
- Bereitstellungs-ID (falls zutreffend)
- Relevante Fehlermeldungen
- Zeitplan der Sicherheitsänderungen

Grundlegendes zur Migrationszuordnung von IIS zu Elastic Beanstalk

Die Migration von IIS zu Elastic Beanstalk beinhaltet die Zuordnung Ihrer lokalen Windows-Serverkonfiguration zu AWS Cloud-Ressourcen. Das Verständnis dieser Zuordnung ist entscheidend für erfolgreiche Migrationen und das Management nach der Migration.

IIS-Sites und -Anwendungen in Elastic Beanstalk

In IIS stellt eine Website eine Sammlung von Webanwendungen und virtuellen Verzeichnissen mit jeweils eigener Konfiguration und eigenem Inhalt dar. Bei der Migration zu Elastic Beanstalk werden diese Komponenten wie folgt transformiert:

IIS-Websites

Ihre IIS-Websites werden zu Anwendungen innerhalb von Elastic Beanstalk. Die Konfiguration jeder Website, einschließlich ihrer Bindungen, Anwendungspools und Authentifizierungseinstellungen, wird über das Deployment-Manifest () von Elastic Beanstalk beibehalten. `aws-windows-deployment-manifest.json`

Wenn Sie beispielsweise mehrere Websites wie Default Web Site und haben IntranetSite, werden die Inhalte und die Konfiguration jeder Site in eb migrate Paketen zusammengefasst, wobei die Isolierung beibehalten wird.

Der Befehl erstellt entsprechende Application Load Balancer (ALB) -Listener-Regeln für die Bearbeitung von Routing-Anfragen an Ihre Anwendungen. Außerdem werden Sicherheitsgruppen konfiguriert, um einen ordnungsgemäßen Portzugriff auf der Grundlage Ihrer ursprünglichen IIS-Bindungen sicherzustellen.

Anwendungspools

IIS-Anwendungspools bieten Funktionen zur Isolierung von Arbeitsprozessen, zur Laufzeitverwaltung und zum Recycling Ihrer Anwendungen. In Elastic Beanstalk werden diese Umgebungsprozessen zugeordnet, die über den `aws:elasticbeanstalk:environment:process` Namespace definiert und über IIS auf den Instances konfiguriert wurden. EC2

Bei der Migration werden wichtige Einstellungen für den Anwendungspool beibehalten, darunter die folgenden:

- Konfigurationen des Prozessmodells — Identität (ApplicationPoolIdentity,, oder benutzerdefinierte Konten) NetworkService, Einstellungen für Leerlauf-Timeouts und Intervalle für die Wiederverwertung von Prozessen
- .NET CLR-Versionseinstellungen — Behält Ihre angegebene .NET Framework-Version (v2.0, v4.0 oder No Managed Code) bei, um die Anwendungskompatibilität sicherzustellen
- Verwalteter Pipeline-Modus — Behält die Einstellungen für den integrierten oder klassischen Pipeline-Modus bei, um Ihre Architektur für die HTTP-Anforderungsverarbeitung beizubehalten
- Erweiterte Einstellungen — Warteschlangenlänge, CPU-Grenzwerte, Schwellenwerte für den Rapid-Fail-Schutz und Startzeitlimits

Der eb migrate Befehl behält die Zuordnungen zwischen Sites und Anwendungspools während der Migration zu Ihrer Elastic Beanstalk Beanstalk-Umgebung bei.

Wenn Ihre Anwendungspools benutzerdefinierte Wiederverwendungszeitpläne (bestimmte Zeiten oder Speichergrenzwerte) verwenden, werden diese mithilfe von PowerShell Skripts im Bereitstellungspaket implementiert, die die entsprechenden IIS-Einstellungen auf den Instances konfigurieren. EC2

Webseiten-Bindungen

IIS-Website-Bindungen, die definieren, wie Clients auf Ihre Anwendungen zugreifen, werden in die folgenden Application Load Balancer (ALB) -Konfigurationen umgewandelt:

- Port-Bindungen werden den entsprechenden ALB-Listener-Regeln zugeordnet
- Host-Header-Konfigurationen werden in ALB-Routing-Regeln übersetzt
- SSL-fähige Websites verwenden AWS Certificate Manager (ACM) für die Zertifikatsverwaltung

Verwaltung virtueller Verzeichnisse und Anwendungspfade

Virtuelle IIS-Verzeichnisse und -Anwendungen ermöglichen die Zuordnung von URL-Pfaden zu physischen Verzeichnissen. Elastic Beanstalk hält diese Beziehungen mithilfe der folgenden Konstrukte aufrecht:

Virtuelle Verzeichnisse

Beim Migrationsprozess werden die physischen Pfade Ihrer virtuellen Verzeichnisse im Bereitstellungspaket beibehalten.

Pfadzuordnungen werden in der IIS-Konfiguration auf den EC2 Instanzen konfiguriert, sodass sichergestellt wird, dass Ihre URL-Struktur nach der Migration intakt bleibt.

2 Physische Pfade des Laufwerks

> **Important**
>
> Standardmäßig stellen Elastic Beanstalk Beanstalk-Windows-Umgebungen nur das Laufwerk C:\ (Root-Volume) bereit. In der aktuellen Version werden Anwendungen, deren Inhalt sich auf den Laufwerken „D:\“, E:\ usw. befindet, für die Migration nicht unterstützt.

Der eb migrate Befehl erkennt automatisch physische Pfade, die sich auf den Laufwerken befinden, und warnt Sie vor möglichen Problemen, wie im folgenden Beispiel:

```
ERROR: Detected physical paths on drive D:\ which are not supported in the current
 version:
  - D:\websites\intranet
  - D:\shared\images

Migration of content from non-system drives is not supported. Please relocate this
 content to the C:\ drive before migration. Otherwise, select only those sites that
 are on C:\.
```

Wenn Ihre Anwendung von Laufwerken abhängig ist, müssen Sie Ihre Anwendung so ändern, dass sie vor der Migration den gesamten Inhalt auf Laufwerk C:\ speichert.

Verschachtelte Anwendungen

Unter Websites verschachtelte Anwendungen werden mit den richtigen Pfadkonfigurationen und den entsprechenden Anwendungspoolzuweisungen bereitgestellt. Beim Migrationsprozess

werden alle `web.config` Einstellungen beibehalten, sodass sichergestellt wird, dass anwendungsspezifische Konfigurationen in der Cloud-Umgebung weiterhin erwartungsgemäß funktionieren.

URL-Umschreibung und Routing von Anwendungsanfragen (ARR)

Wenn Ihre IIS-Bereitstellung URL Rewrite oder Application Request Routing (ARR) verwendet, werden diese Konfigurationen anhand der folgenden Regeln und Konfigurationen eb migrate behandelt:

Regeln für das Umschreiben von URLs

Regeln zum Umschreiben von URLs aus Ihren `web.config` Dateien werden nach Möglichkeit in ALB-Routing-Regeln übersetzt. Der folgende Eintrag wird beispielsweise zu einer ALB-Listener-Regel, die den Datenverkehr auf der Grundlage von Host-Headern und Pfadmustern leitet. :

```
<!-- Original IIS URL Rewrite Rule -->
<rule name="Redirect to WWW" stopProcessing="true">
    <match url="(.*)" />
    <conditions>
        <add input="{HTTP_HOST}" pattern="^example.com$" />
    </conditions>
    <action type="Redirect" url="http://www.example.com/{R:1}" />
</rule>
```

Weiterleitung von Anwendungsanfragen

ARR-Konfigurationen werden durch die Installation von ARR-Funktionen auf EC2 Instanzen beibehalten. Der Migrationsprozess umfasst die folgenden Aufgaben:

- Konfiguriert die Proxyeinstellungen so, dass sie zu Ihrer Quellumgebung passen
- Behält die mit ARR verknüpften Regeln zum Umschreiben von URLs bei

Struktur des Migrationsartefakts

Bei der Ausführung wird ein strukturiertes Verzeichnis erstellteb migrate, das alle erforderlichen Bereitstellungskomponenten enthält. In der folgenden Liste wird die Verzeichnisstruktur beschrieben:

```
C:\migration_workspace\
```

```
### .\migrations\latest\
    ### upload_target\
        ### [SiteName].zip                 # One ZIP per IIS site
        ### aws-windows-deployment-manifest.json
        ### ebmigrateScripts\
            ### site_installer.ps1         # Site installation scripts
            ### arr_configuration.ps1      # ARR configuration scripts
            ### permission_handler.ps1     # Permission management
            ### firewall_config.ps1        # Windows Firewall rules
```

Die `aws-windows-deployment-manifest.json` Datei ist die zentrale Konfigurationsdatei, die Elastic Beanstalk anweist, wie Sie Ihre Anwendungen bereitstellen. Sehen Sie sich die folgende Beispielstruktur an:

```
{
    "manifestVersion": 1,
    "deployments": {
        "msDeploy": [
            {
                "name": "Primary Site",
                "parameters": {
                    "appBundle": "DefaultWebSite.zip",
                    "iisPath": "/",
                    "iisWebSite": "Default Web Site"
                }
            }
        ],
        "custom": [
            {
                "name": "ConfigureARR",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\\arr_configuration.ps1"
                    },
                    "uninstall": {
                        "file": "ebmigrateScripts\\noop.ps1"
                    },
                    "restart": {
                        "file": "ebmigrateScripts\\noop.ps1"
                    }
                }
            }
        ]
```

```
    }
}
```

Dieses Manifest stellt die folgenden Ergebnisse für Ihre Migration sicher:

- Anwendungen werden bereitgestellt, um IIS-Pfade zu korrigieren
- Benutzerdefinierte Konfigurationen werden angewendet
- Standortspezifische Einstellungen werden beibehalten
- Die Reihenfolge der Bereitstellung wird beibehalten

Fortgeschrittene Migrationsszenarien

In diesem Abschnitt werden erweiterte Migrationsszenarien für komplexe IIS-Bereitstellungen behandelt.

Migrationen an mehreren Standorten mit Application Request Routing (ARR)

Der eb migrate Befehl erkennt ARR-Konfigurationen während der Migration automatisch und behält sie bei. Wenn er die ARR-Einstellungen in Ihrem IIS identifiziert`applicationHost.config`, generiert er die erforderlichen PowerShell Skripts, um ARR auf den EC2 Zielinstanzen neu zu installieren und zu konfigurieren.

Erkennung der ARR-Konfiguration

Der Migrationsprozess untersucht drei wichtige Konfigurationsabschnitte in IIS:

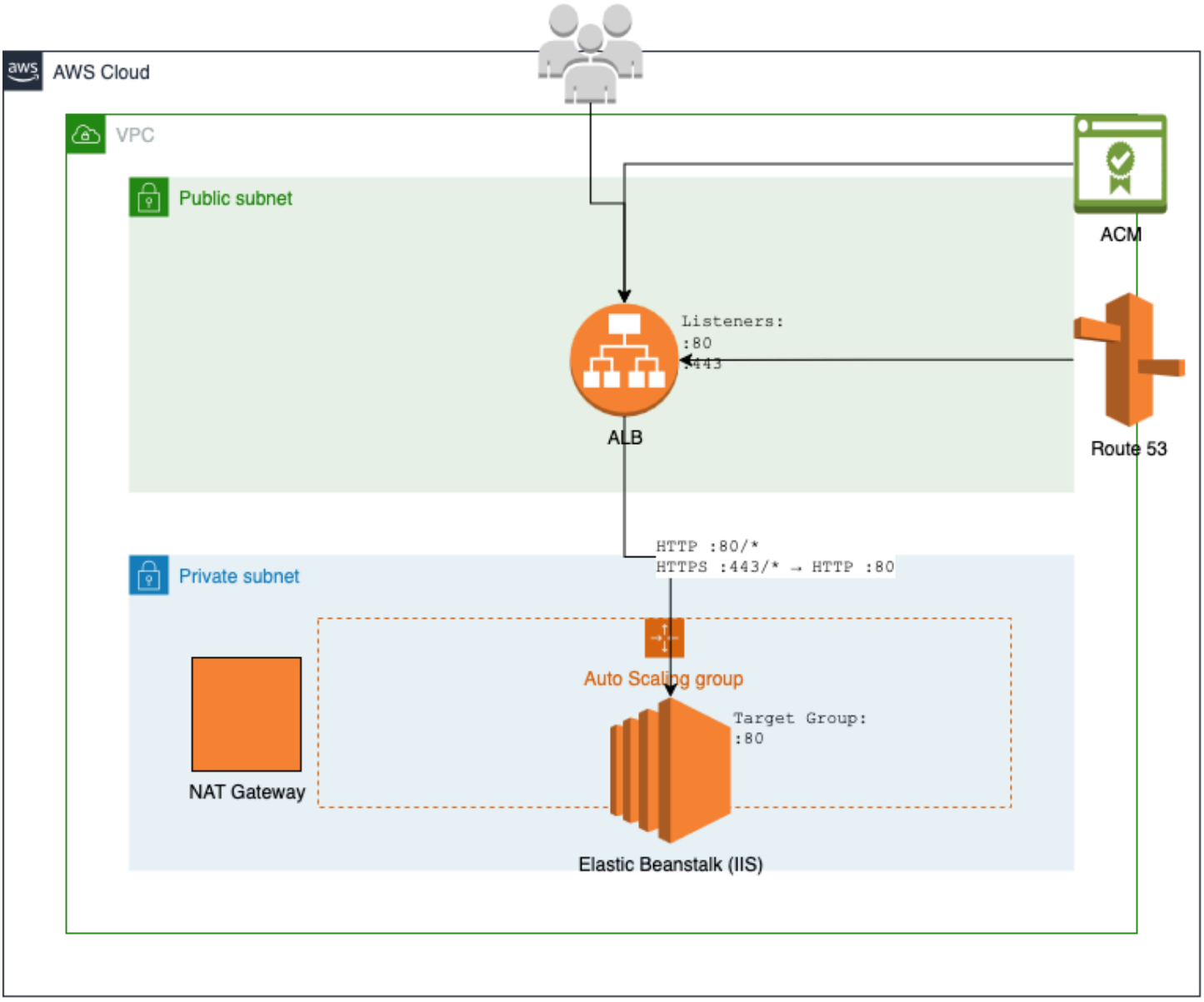
- `system.webServer/proxy`: Kern-ARR-Proxyeinstellungen
- `system.webServer/rewrite`: Regeln für das Umschreiben von URLs
- `system.webServer/caching`: Konfiguration zwischenspeichern

Stellen Sie sich zum Beispiel eine gängige ARR-Konfiguration vor, bei der ein auf Port 80 `RouterSite` laufendes Objekt Anfragen an die Ports 8081 bzw. 8082 `APIService` `AdminPortal` weiterleitet und auf diesen Ports läuft:

```
<!-- Original IIS ARR Configuration -->
```

```
<rewrite>
    <rules>
        <rule name="Route to API" stopProcessing="true">
            <match url="^api/(.*)$" />
            <action type="Rewrite" url="http://backend:8081/api/{R:1}" />
        </rule>
        <rule name="Route to Admin" stopProcessing="true">
            <match url="^admin/(.*)$" />
            <action type="Rewrite" url="http://backend:8082/admin/{R:1}" />
        </rule>
    </rules>
</rewrite>
```

Das folgende Diagramm zeigt, wie diese Regeln hinter Port 80 im IIS-Server versteckt sind und nicht über die Sicherheitsgruppen offengelegt werden. EC2 Nur Port 80 ist für den Application Load Balancer zugänglich und der gesamte Datenverkehr von dort wird an Port 80 an die Zielgruppe weitergeleitet.



Mit dem folgenden Befehl kann diese Konfiguration migriert werden:

```
PS C:\migrations_workspace> eb migrate --sites "RouterSite,APIService,AdminPortal" `
    --copy-firewall-config
```

ARR-Migrationsprozess

Beim Migrationsprozess wird Ihre ARR-Konfiguration in mehreren Schritten beibehalten.

Export der Konfiguration

Das Tool exportiert Ihre vorhandenen ARR-Einstellungen aus den drei wichtigsten Konfigurationsabschnitten in separate XML-Dateien, die im `ebmigrateScripts` Verzeichnis gespeichert sind:

```
ebmigrateScripts\
### arr_config_proxy.xml
### arr_config_rewrite.xml
### arr_config_caching.xml
```

Installationsskripten

Für das ARR-Setup werden zwei PowerShell Skripte generiert:

1. `arr_msi_installer.ps1`: Lädt das ARR-Modul herunter und installiert es
2. `arr_configuration_importer_script.ps1`: Importiert Ihre exportierte ARR-Konfiguration

Integration des Bereitstellungsmanifests

Die Skripts werden durch Einträge in den folgenden Bereichen in den Bereitstellungsprozess integriert`aws-windows-deployment-manifest.json`:

```
{
    "manifestVersion": 1,
    "deployments": {
        "custom": [
            {
                "name": "WindowsProxyFeatureEnabler",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\
\windows_proxy_feature_enabler.ps1"
                    }
                }
            },
            {
                "name": "ArrConfigurationImporterScript",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\
\arr_configuration_importer_script.ps1"
```

```
                    }
                }
            }
        ]
    }
}
```

Integration des Load Balancers

Der Migrationsprozess übersetzt Ihre ARR-Regeln nach Möglichkeit in Application Load Balancer (ALB) -Listener-Regeln. Die obige ARR-Konfiguration führt beispielsweise zu ALB-Regeln, die den Datenverkehr auf der Grundlage von URL-Pfadmustern weiterleiten und gleichzeitig das interne Routing auf den Instances beibehalten. EC2

Die resultierende Umgebung behält Ihre ARR-Routing-Logik bei und nutzt gleichzeitig AWS die elastische Infrastruktur. Ihre Anwendungen funktionieren weiterhin wie zuvor, wobei ARR das interne Routing übernimmt, während der Application Load Balancer die externe Datenverkehrsverteilung verwaltet.

Migrationen an mehreren Standorten ohne ARR mithilfe von hostbasiertem Routing

Application Request Routing (ARR) ist zwar ein gängiger Ansatz für die Verwaltung mehrerer Sites in IIS, aber Sie können Bereitstellungen mit mehreren Standorten auch ohne ARR direkt zu Elastic Beanstalk migrieren, indem Sie die hostbasierten Routing-Funktionen des Application Load Balancers nutzen. Dieser Ansatz kann die Komplexität reduzieren und die Leistung verbessern, indem eine zusätzliche Routing-Ebene entfällt.

Überblick über das hostbasierte Routing

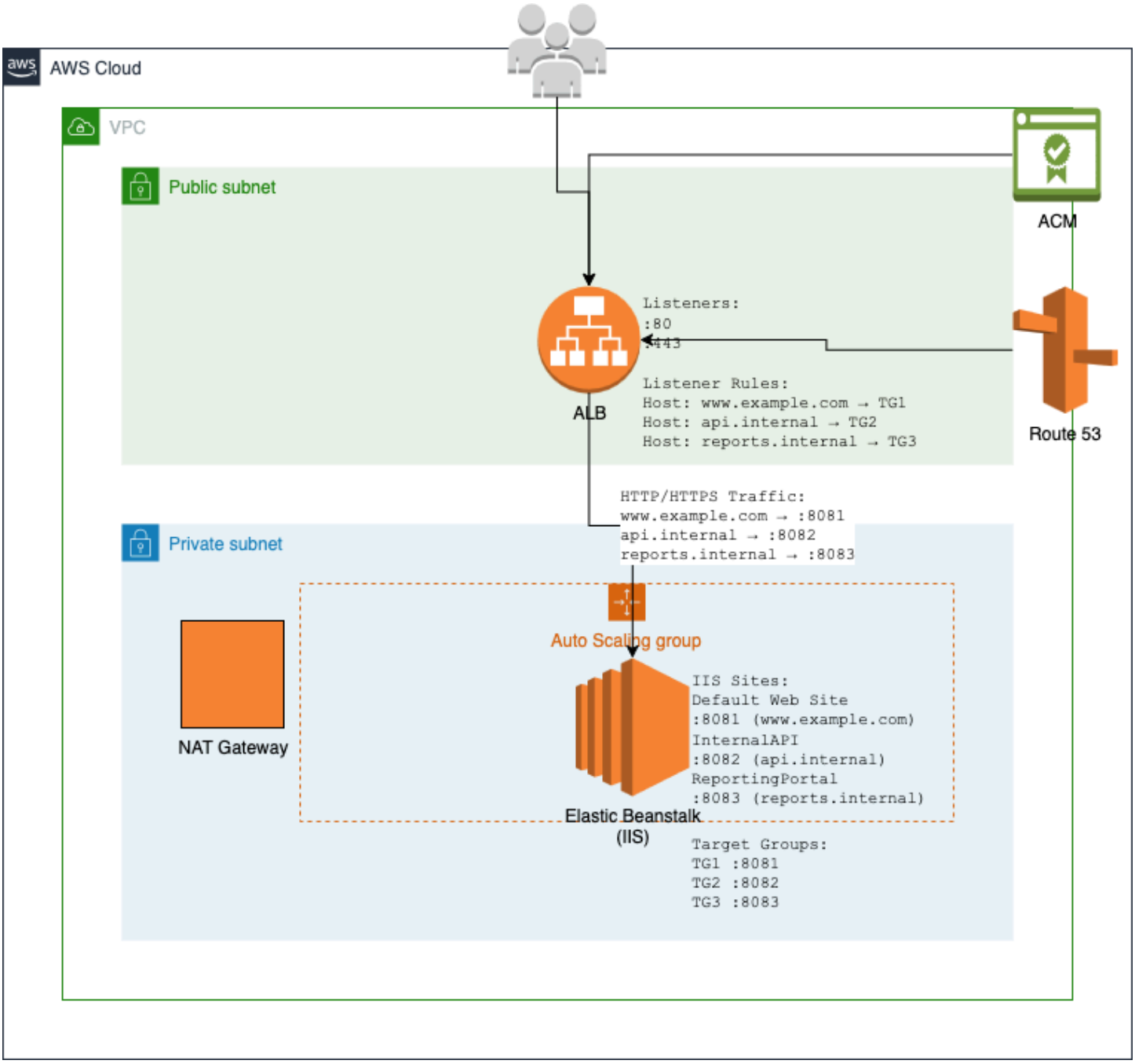
Bei diesem Ansatz wird jede IIS-Site außerhalb der EC2 Instance verfügbar gemacht, und der Application Load Balancer leitet den Datenverkehr basierend auf dem Host-Header direkt an den entsprechenden Port weiter. Dadurch entfällt die Notwendigkeit von ARR, während gleichzeitig die Trennung zwischen Ihren Anwendungen gewahrt bleibt.

Stellen Sie sich eine IIS-Konfiguration mit mehreren Standorten mit drei Standorten vor, von denen jeder über eine eigene Hostnamenbindung verfügt:

```
<sites>
```

```
    <site name="Default Web Site" id="1">
        <bindings>
            <binding protocol="http" bindingInformation="*:8081:www.example.com" />
        </bindings>
    </site>
    <site name="InternalAPI" id="2">
        <bindings>
            <binding protocol="http" bindingInformation="*:8082:api.internal" />
        </bindings>
    </site>
    <site name="ReportingPortal" id="3">
        <bindings>
            <binding protocol="http" bindingInformation="*:8083:reports.internal" />
        </bindings>
    </site>
</sites>
```

Diese Sites sind über die Sicherheitsgruppen an den Ports 8081, 8082 und 8083 verfügbar. EC2 Der Application Load Balancer leitet sie auf der Grundlage der Load Balancer-Listener-Regelkonfiguration weiter.



Migrationsprozess

Um diese Konfiguration ohne ARR auf Elastic Beanstalk zu migrieren, verwenden Sie den eb migrate Befehl im folgenden Beispiel:

```
PS C:\migrations_workspace> eb migrate --sites "Default Web
 Site,InternalAPI,ReportingPortal"
```

Der Migrationsprozess konfiguriert den Application Load Balancer automatisch mit hostbasierten Routing-Regeln, die den Traffic auf der Grundlage des Host-Headers an die entsprechende Zielgruppe weiterleiten. Jede Zielgruppe leitet den Traffic an den entsprechenden Port auf Ihren Instances weiter: EC2

1. Host-Header www.example.com → Zielgruppe auf Port 8081
2. Host-Header api.internal → Zielgruppe auf Port 8082
3. Host-Header reports.internal → Target Group auf Port 8083

SSL/TLS-Konfiguration

Gehen Sie wie folgt vor, um Ihre Anwendungen mit SSL/TLS zu sichern:

1. Fordern Sie Zertifikate für Ihre Domains über AWS Certificate Manager(ACM) an.
2. Konfigurieren Sie HTTPS-Listener auf Ihrem Application Load Balancer mithilfe dieser Zertifikate.
3. Aktualisieren Sie Ihre Umgebungskonfiguration so, dass sie HTTPS-Listener mit den folgenden Einstellungen für Konfigurationsoptionen enthält.

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: HTTPS
    SSLCertificateId: arn:aws:acm:region:account-id:certificate/certificate-id
    InstancePort: 80
    InstanceProtocol: HTTP
```

Bei dieser Konfiguration erfolgt die SSL-Terminierung am Load Balancer, und der Datenverkehr wird über HTTP an Ihre Instances weitergeleitet. Dies vereinfacht die Zertifikatsverwaltung und gewährleistet gleichzeitig sichere Verbindungen zu Clients.

Best Practices

Sicherheitsgruppen

Konfigurieren Sie Sicherheitsgruppen so, dass eingehender Datenverkehr nur an den Ports zugelassen wird, die von Ihren IIS-Sites (in diesem Beispiel 8081, 8082, 8083) aus der Application Load Balancer Balancer-Sicherheitsgruppe verwendet werden.

Health checks (Zustandsprüfungen)

Konfigurieren Sie Integritätsprüfungen für jede Zielgruppe, um sicherzustellen, dass der Datenverkehr nur an fehlerfreie Instances weitergeleitet wird. Erstellen Sie Endpunkte für Integritätsprüfungen für jede Anwendung, falls sie noch nicht vorhanden sind.

Überwachen

Richten Sie CloudWatch Alarme ein, um den Zustand und die Leistung jeder Zielgruppe separat zu überwachen. Auf diese Weise können Sie Probleme identifizieren, die für einzelne Anwendungen spezifisch sind.

Skalierung

Berücksichtigen Sie bei der Konfiguration von Auto Scaling-Richtlinien die Ressourcenanforderungen aller Anwendungen. Wenn eine Anwendung erheblich unterschiedliche Ressourcenanforderungen hat, sollten Sie erwägen, sie in eine separate Umgebung zu migrieren.

Verwaltung virtueller Verzeichnisse

Der eb migrate Befehl behält virtuelle Verzeichnisstrukturen bei der Migration Ihrer IIS-Anwendungen zu Elastic Beanstalk bei.

Standardkonfiguration für Berechtigungen

eb migrateLegt bei der Migration virtueller Verzeichnisse einen Basissatz von Berechtigungen fest, indem ReadAndExecute Zugriff gewährt wird auf:

- IIS_IUSRS
- IUSR
- Authentifizierte Benutzer

Stellen Sie sich zum Beispiel eine typische virtuelle Verzeichnisstruktur vor:

```
<site name="CorporatePortal">
    <application path="/" applicationPool="CorporatePortalPool">
        <virtualDirectory path="/" physicalPath="C:\sites\portal" />
        <virtualDirectory path="/shared" physicalPath="C:\shared\content" />
        <virtualDirectory path="/reports" physicalPath="D:\reports" />
    </application>
```

```
</site>
```

Kennwortgeschützte virtuelle Verzeichnisse

Wenn eb migrate kennwortgeschützte virtuelle Verzeichnisse gefunden werden, werden Warnungen ausgegeben und ein manuelles Eingreifen ist erforderlich.

Das folgende Konfigurationsbeispiel führt zu der Warnantwort, die dem Beispiel folgt.

```
<virtualDirectory path="/secure"
                  physicalPath="C:\secure\content"
                  userName="DOMAIN\User"
                  password="[encrypted]" />
```

```
[WARNING] CorporatePortal/secure is hosted at C:\secure\content which is password-
protected and won't be copied.
```

Um den Kennwortschutz aufrechtzuerhalten, erstellen Sie ein benutzerdefiniertes Bereitstellungsskript wie das Folgende:

```
# PS C:\migrations_workspace> cat secure_vdir_config.ps1

$vdirPath = "C:\secure\content"
$siteName = "CorporatePortal"
$vdirName = "secure"
$username = "DOMAIN\User"
$password = "SecurePassword"

# Ensure directory exists
if (-not (Test-Path $vdirPath)) {
    Write-Host "Creating directory: $vdirPath"
    New-Item -Path $vdirPath -ItemType Directory -Force
}

# Configure virtual directory with credentials
Write-Host "Configuring protected virtual directory: $vdirName"
New-WebVirtualDirectory -Site $siteName -Name $vdirName `
    -PhysicalPath $vdirPath -UserName $username -Password $password

# Set additional permissions as needed
$acl = Get-Acl $vdirPath
```

```
$rule = New-Object System.Security.AccessControl.FileSystemAccessRule(
    $username, "ReadAndExecute", "ContainerInherit,ObjectInherit", "None", "Allow"
)
$acl.AddAccessRule($rule)
Set-Acl $vdirPath $acl
```

Fügen Sie dieses Skript zu Ihrer Bereitstellung hinzu, indem Sie es in das Manifest aufnehmen:

```
{
    "manifestVersion": 1,
    "deployments": {
        "custom": [
            {
                "name": "SecureVirtualDirectory",
                "scripts": {
                    "install": {
                        "file": "secure_vdir_config.ps1"
                    }
                }
            }
        ]
    }
}
```

Benutzerdefiniertes Berechtigungsmanagement

Der eb migrate Befehl bietet ein Framework für benutzerdefinierte Berechtigungsskripten für Anwendungen, für die andere Berechtigungen als die Standardberechtigungen erforderlich sind.

```
$paths = @(
    "C:\sites\portal\uploads",
    "C:\shared\content"
)

foreach ($path in $paths) {
    if (-not (Test-Path $path)) {
        Write-Host "Creating directory: $path"
        New-Item -Path $path -ItemType Directory -Force
    }

    $acl = Get-Acl $path
```

```
    # Add custom permissions
    $customRules = @(
        # Application Pool Identity - Full Control
        [System.Security.AccessControl.FileSystemAccessRule]::new(
            "IIS AppPool\CorporatePortalPool",
            "FullControl",
            "ContainerInherit,ObjectInherit",
            "None",
            "Allow"
        ),
        # Custom Service Account
        [System.Security.AccessControl.FileSystemAccessRule]::new(
            "NT SERVICE\CustomService",
            "Modify",
            "ContainerInherit,ObjectInherit",
            "None",
            "Allow"
        )
    )

    foreach ($rule in $customRules) {
        $acl.AddAccessRule($rule)
    }

    Set-Acl $path $acl
    Write-Host "Custom permissions applied to: $path"
}
```

Best Practices

Folgen Sie diesen bewährten Methoden, um Ihre Migration zu planen, auszuführen, zu überwachen und zu verifizieren.

Planung vor der Migration

Dokumentieren Sie bestehende Berechtigungen und Authentifizierungsanforderungen vor der Migration. Testen Sie benutzerdefinierte Berechtigungsskripts in einer Entwicklungsumgebung, bevor Sie sie in der Produktion einsetzen.

Verwaltung gemeinsam genutzter Inhalte

Stellen Sie bei Verzeichnissen mit gemeinsam genutzten Inhalten sicher, dass alle erforderlichen Dateisystemberechtigungen mithilfe benutzerdefinierter Skripts ordnungsgemäß konfiguriert

sind. Erwägen Sie die Verwendung von Amazon FSx for Windows File Server für gemeinsame Speicheranforderungen.

Überwachung und Überprüfung

Überwachen Sie die Anwendungsprotokolle nach der Migration, um den ordnungsgemäßen Zugriff auf virtuelle Verzeichnisse zu überprüfen. Achten Sie besonders auf die folgenden Bereiche:

- Zugriff auf die Identität des Anwendungspools
- Benutzerdefinierte Berechtigungen für Dienstkonten
- Netzwerk-Share-Konnektivität
- Authentication failures (Authentifizierungsfehler)

Benutzerdefinierte Einstellungen für den Anwendungspool

Der eb migrate Befehl kopiert standardmäßig keine benutzerdefinierten Anwendungspool-Einstellungen. Um benutzerdefinierte Anwendungspoolkonfigurationen beizubehalten, gehen Sie wie folgt vor, um einen benutzerdefinierten Manifestabschnitt zu erstellen und anzuwenden.

1. Erstellen Sie ein Archiv Ihrer Migrationsartefakte.

```
PS C:\migrations_workspace> eb migrate --archive
```

2. Erstellen Sie ein benutzerdefiniertes PowerShell Skript zur Konfiguration von Anwendungspools.

```
# PS C:\migrations_workspace> cat .\migrations\latest\upload_target
\customize_application_pool_config.ps1

$configPath = "$env:windir\System32\inetsrv\config\applicationHost.config"

[xml]$config = Get-Content -Path $configPath

$newPoolXml = @"
<!-- Original IIS Configuration -->
<applicationPools>
    <add name="CustomPool"
         managedRuntimeVersion="v4.0"
         managedPipelineMode="Integrated">
        <processModel identityType="SpecificUser"
                      userName="AppPoolUser"
```

```
                    password="[encrypted]" />
        <recycling>
            <periodicRestart time="00:00:00">
                <schedule>
                    <add value="02:00:00" />
                    <add value="14:00:00" />
                </schedule>
            </periodicRestart>
        </recycling>
    </add>
</applicationPools>
"@
$newPoolXmlNode = [xml]$newPoolXml

# Find the applicationPools section
$applicationPools = $config.SelectSingleNode("//configuration/
system.applicationHost/applicationPools")

# Import the new node into the document
$importedNode = $config.ImportNode($newPoolXmlNode.DocumentElement, $true)
$applicationPools.AppendChild($importedNode)

# Save the changes
$config.Save($configPath)

Write-Host "ApplicationHost.config has been updated successfully."
```

3. Aktualisieren Sie die `aws-windows-deployment-manifest.json` Datei so, dass sie Ihr benutzerdefiniertes Skript enthält.

```
{
    "manifestVersion": 1,
    "deployments": {
        ...
        "custom": [
            ...,
            {
                "name": "ModifyApplicationPoolConfig",
                "description": "Modify application pool configuration from source
 machine to remove",
                "scripts": {
                    "install": {
                        "file": "customize_application_pool_config.ps1"
```

```
                    },
                    "restart": {
                        "file": "ebmigrateScripts\\noop.ps1"
                    },
                    "uninstall": {
                        "file": "ebmigrateScripts\\noop.ps1"
                    }
                }
            }
        ]
    }
}
```

4. Erstellen Sie eine Umgebung mit dem aktualisierten Archivverzeichnis.

```
PS C:\migrations_workspace> eb migrate `
    --archive-dir '.\migrations\latest\upload_target\'
```

Das `--archive-dir` Argument weist eb migrate an, den zuvor erstellten Quellcode zu verwenden, um die Erstellung neuer Archive zu vermeiden.

Bereitstellung früherer Versionen

Der eb migrate führt eine Historie Ihrer Migrationen anhand von Verzeichnissen und Anwendungsversionen mit Zeitstempel in Elastic Beanstalk. Bei jeder Migration wird eine einzigartige ZIP-Datei erstellt, die bei Bedarf bereitgestellt werden kann.

```
PS C:\migrations_workspace> ls .\migrations\

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----l        3/18/2025  10:34 PM                latest
d-----        3/16/2025   5:47 AM                migration_1742104049.479849
d-----        3/17/2025   9:18 PM                migration_1742246303.18056
d-----        3/17/2025   9:22 PM                migration_1742246546.565739
...
d-----        3/18/2025  10:34 PM                migration_1742337258.30742
```

Der `latest` symbolische Link verweist immer auf das zuletzt erstellte Migrationsartefaktverzeichnis. Zusätzlich zu den relevanten Anwendungs- und Fehlerprotokollen enthält jedes

Migrationsartefaktverzeichnis auch eine `upload_target.zip` Datei, die Sie auf Elastic Beanstalk bereitstellen können.

```
PS C:\migrations_workspace> ls .\migrations\latest\

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        3/18/2025  10:34 PM                upload_target
-a----        3/18/2025  10:34 PM          13137 application.log
-a----        3/18/2025  10:34 PM              0 error.log
-a----        3/18/2025  10:34 PM        1650642 upload_target.zip
```

Sie können die Datei wie folgt bereitstellen: `upload_target.zip` eb migrate

```
PS C:\migrations_workspace> eb migrate --zip .\migrations\latest\upload_target.zip
```

Problembehebung und Diagnose

> **Testen Sie Amazon Q Developer CLI zur KI-gestützten Fehlerbehebung**
>
> Amazon Q Developer CLI kann Ihnen helfen, Umgebungsprobleme schnell zu beheben. Die Q CLI bietet Lösungen, indem sie den Umgebungsstatus überprüft, Ereignisse überprüft, Protokolle analysiert und klärende Fragen stellt. Weitere Informationen und detaillierte Anleitungen finden Sie in den Blogs unter Troubleshooting Elastic Beanstalk Environments with Amazon Q Developer CLI. AWS

Dieser Abschnitt enthält Anleitungen zur Behebung häufiger Probleme, die bei der Migration von IIS-Anwendungen zu Elastic Beanstalk auftreten können.

Ein EC2 Schlüsselpaar mit Ihrer Umgebung verknüpfen

Sie können sich mit einem Amazon-Schlüsselpaar sicher bei den Amazon Elastic Compute Cloud (Amazon EC2) -Instances anmelden, die für Ihre Elastic Beanstalk Beanstalk-Anwendung bereitgestellt wurden. EC2 Anweisungen zum Erstellen eines key pair finden Sie unter Erstellen eines Schlüsselpaars mit Amazon EC2 im EC2 Amazon-Benutzerhandbuch.

Die Angabe eines Schlüsselnamens für eb migrate hat zur Folge, dass Ihre Elastic Beanstalk Beanstalk-Umgebung mit dem Schlüsselpaar verknüpft wird. Aus Sicherheitsgründen wird dadurch

Port 3389 in der Sicherheitsgruppe Ihrer Instance nicht geöffnet. EC2 Sie können zusätzliche EC2 Sicherheitsgruppen zuordnen, die den Datenverkehr über Port 3389 auch eb config nach der ersten Migration zulassen.

```
PS C:\migrations_workspace> eb migrate  `
    --keyname "my-keypair"  `
    --verbose
```

Wenn Sie ein key pair erstellen, EC2 speichert Amazon eine Kopie Ihres öffentlichen Schlüssels. Wenn Sie es nicht mehr verwenden müssen, um eine Verbindung zu Umgebungsinstanzen herzustellen, können Sie es aus Amazon löschen EC2. Einzelheiten finden Sie unter Löschen Ihres Schlüsselpaars im EC2 Amazon-Benutzerhandbuch.

Weitere Informationen zum Herstellen einer Verbindung zu EC2 Windows-Amazon-Instances finden Sie unter Verbindung zu Windows-Instances herstellen.

Auf Protokolle zugreifen

Die EB CLI bietet eine eb logs Funktion, mit der Sie Logs aus einer Elastic Beanstalk Beanstalk-Umgebung abrufen können, ohne sich bei deren EC2 Instanzen anmelden zu müssen. Nach der Ausführung von können Sie den eb logs --zip Befehl ausführeneb migrate, mit dem Logs heruntergeladen und im `.elasticbeanstalk\logs` Verzeichnis gespeichert werden.

Alternativ können Sie Logs auch über die AWS Elastic Beanstalk Beanstalk-Konsole anzeigen. Weitere Informationen finden Sie unter Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen.

Zugreifen auf clientseitige Artefakte

Der eb migrate Befehl speichert Anwendungs- und Fehlerprotokolle, die von msdeploy internen Verzeichnissen für Migrationsartefakte generiert wurden.

```
./migrations/
### latest -> migration_20240308_123456/
### migration_20240308_123456/
    ### application.log
    ### error.log
    ### upload_target\
```

Überwachung des Zustands der Umgebung

Elastic Beanstalk unterstützt Sie mit den erweiterten Funktionen zur Gesundheitsüberwachung bei der Überwachung des Zustands. Es handelt sich um ein automatisiertes System zur Gesundheitsüberwachung, das kontinuierlich den Betriebsstatus von Anwendungsinstanzen verfolgt und dabei integrierte Messwerte wie CPU-Auslastung, Latenz, Anzahl der Anfragen und Antwortcodes nutzt.

Das System zur Gesundheitsüberwachung verwendet einen agentenbasierten Ansatz zur Erfassung von Daten auf Instanzebene und ist mit Protokollierung und Warnmeldungen in Echtzeit integriert. Elastic Load Balancing (ELB) und Auto Scaling reagieren dynamisch auf Änderungen des Gesundheitsstatus und sorgen so für hohe Verfügbarkeit und Fehlertoleranz. Erweiterte Überwachungsmodi, einschließlich erweiterter Statusberichte, bieten einen detaillierten Einblick in das Anwendungsverhalten und ermöglichen so eine proaktive Fehlerbehebung und automatische Wiederherstellungsmechanismen.

Führen Sie den EB eb health CLI-Befehl aus, um den Zustand der Umgebung anzuzeigen. Die folgenden Informationen werden angezeigt:

- Instance-Zustand
- Kennzahlen zur Reaktion von Anwendungen
- Auslastung der Systemressourcen
- Aktuelle Bereitstellungsereignisse

EC2 Leistungsoptimierung

eb migrateWählt standardmäßig den Instance-Typ c5.2xlarge aus, um ein optimales Ersterlebnis mit Elastic Beanstalk zu bieten. Sie können dieses Verhalten mit dem Argument überschreiben: --instance-type

```
PS C:\migrations_workspace> eb migrate `
    --instance-type "t3.large"
```

Für Produktionsumgebungen sollten Sie bei der Auswahl eines Instance-Typs die folgenden Faktoren berücksichtigen:

- Speicheranforderungen Ihrer Anwendungen

- CPU-Anforderungen für die Verarbeitung von Workloads
- Anforderungen an die Netzwerkleistung
- Ziele zur Kostenoptimierung

Konfiguration des EBS-Volumes

Standardmäßig erstellt Elastic Beanstalk nur ein Root-Block-Device-Volume (`C:\`) für Ihre Umgebung. Sie können zusätzliche Amazon Elastic Block Store-Snapshot-Volumes mit der folgenden --ebs-snapshots Option übergeben:

```
PS C:\migrations_workspace> eb migrate `
    --ebs-snapshots "snap-123456789abc"
```

Beispiele dafür, wie Sie Blockgeräte-Mappings mit Elastic Beanstalk konfigurieren können, finden Sie im Blogartikel Customize Ephemeral and EBS Volumes in Elastic Beanstalk Environments.

Für Anwendungen mit hohen Speicheranforderungen sollten Sie die folgenden Optionen in Betracht ziehen:

- EBS-Volumes für persistente Daten verwenden
- Implementierung von Amazon S3 für statische Inhalte
- Amazon FSx for Windows File Server für gemeinsam genutzte Dateisysteme verwenden

Häufige Probleme und Lösungen

Ereignis: Fehlende Web Deploy-Installation

Wenn Sie auf Fehler stoßen, die darauf zurückzuführen sind, dass Web Deploy nicht gefunden wurde, installieren Sie Web Deploy 3.6 oder höher über den Microsoft Web Platform Installer. Im folgenden Beispiel wird eine mögliche Fehlermeldung angezeigt.

```
Couldn't find msdeploy.exe. Follow instructions here: https://learn.microsoft.com/en-
us/iis/install/installing-publishing-technologies/installing-and-configuring-web-deploy
```

Ereignis: Berechtigungsprobleme während der Migration

Wenn Sie auf berechtigungsbezogene Fehler stoßen, stellen Sie sicher, dass Sie die EB-CLI mit Administratorrechten ausführen. Im folgenden Beispiel wird eine mögliche Fehlermeldung angezeigt.

```
[ERROR] Access to the path 'C:\inetpub\wwwroot\web.config' is denied.
```

Ereignis: Identitätsprobleme mit dem Anwendungspool

Wenn Ihre Anwendung aufgrund von Problemen mit der Identität des Anwendungspools nicht gestartet werden kann, erstellen Sie ein benutzerdefiniertes Skript, um die Identitäten des Anwendungspools zu konfigurieren, wie unter beschrieben. the section called “Benutzerdefinierte Einstellungen für den Anwendungspool”

Ereignis: Fehler bei der Konfiguration des SSL-Zertifikats

Wenn HTTPS-Bindungen nicht funktionieren, stellen Sie sicher, dass Sie mit dem eb mibrate `--ssl-certificates` Optionsparameter einen gültigen ACM-Zertifikat-ARN angegeben haben.

Ereignis: Timeout bei der Erstellung der Umgebung

Wenn bei der Erstellung der Umgebung ein Timeout auftritt, überprüfen Sie die CloudFormation Ereignisse in der AWS Management Console auf bestimmte Fehler bei der Ressourcenerstellung. Zu den häufigsten Ursachen gehören VPC-Konfigurationsprobleme oder Servicebeschränkungen.

Supportanfragen

Wenn Sie auf Probleme stoßen, die Sie nicht lösen können, AWS Support holen Sie sich vor der Kontaktaufnahme die folgenden Informationen ein:

- Umgebungs-ID (`eb status`)
- Anwendungsprotokolle (`eb logs --zip`)
- Migrationsartefakte von `.\migrations\latest\`
- IIS-Quellkonfiguration (Ausgabe von`eb migrate explore --verbose`)
- Detaillierte Fehlermeldungen

Weitere Informationen zur Fehlerbehebung bei Elastic Beanstalk finden Sie unter. Fehlerbehebung in Ihrer Elastic Beanstalk Beanstalk-Umgebung

Vergleich der Migrationsoptionen: EB CLI vs. AWS Application Migration Service

AWS bietet mehrere Pfade für die Migration von Windows-Anwendungen in die Cloud. In diesem Abschnitt werden zwei Hauptoptionen verglichen: der eb migrate Befehl in der EB-CLI und AWS Application Migration Service (MGN). Wenn Sie die Unterschiede zwischen diesen Ansätzen verstehen, können Sie die für Ihre spezifischen Bedürfnisse am besten geeignete Migrationsstrategie auswählen.

Vergleich der Migrationsoptionen

Feature	WEB-CLI (eb migrate)	AWS Application Migration Service (MGN)
Primärer Fokus	Migration von IIS-Websites und -Anwendungen auf Anwendungsebene	Rehosting ganzer Maschinen (physische, virtuelle oder Cloud-Server) auf Serverebene
Am besten geeignet für	IIS-Anwendungen, die Sie mit minimaler Neukonfiguration direkt zu Elastic Beanstalk migrieren möchten	Umfangreiche Migrationen mit vielen Servern oder komplexer Infrastruktur
Discovery-Ansatz	Erkennung von IIS-Websites, -Anwendungen und -Konfigurationen auf Anwendungsebene	Replikation ganzer Maschinen auf Serverebene, einschließlich Betriebssystem und Anwendungen
Zielumgebung	Erstellt und konfiguriert direkt Elastic Beanstalk Beanstalk-Umgebungen, die für Windows-Anwendungen optimiert sind	Erstellt EC2 Instanzen, für die eine zusätzliche Konfiguration erforderlich ist, um mit Elastic Beanstalk zu funktionieren
Beibehaltung der Konfiguration	Behält automatisch IIS-spezifische Konfigurationen (Sites, Anwendungspools, Bindungen) bei	Behält die gesamte Serverkonfiguration bei, die möglicherweise nicht benötigte Komponenten enthält
Bereitstellungsmodell	Schafft eine saubere Elastic Beanstalk-Umgebung, in der Ihre Anwendungen unter Verwendung der	Erstellt ein Replikat Ihres Quellservers, das möglicherweise für Cloud-Operationen optimiert werden muss

Feature	WEB-CLI (eb migrate)	AWS Application Migration Service (MGN)
	Best Practices von Elastic Beanstalk bereitgestellt werden	
Umfang der Migration	Ideal für gezielte Migrationen bestimmter Anwendungen	Konzipiert für groß angelegte Migrationen vieler Server
Schritte nach der Migration	Minimal; die Umgebung ist bereit für den Einsatz mit den Elastic Beanstalk -Management-Tools	Erfordert zusätzliche Schritte für die Integration mit Elastic Beanstalk, z. B. die Ausführung von SSM-Aktionen nach dem Start

Wann sollten die einzelnen Migrationsoptionen verwendet werden

Wählen Sie auseb migrate, wann Sie die folgenden Anforderungen haben:

- Sie möchten nicht ganze Server, sondern nur bestimmte IIS-Anwendungen migrieren
- Ihr Ziel ist es, Elastic Beanstalk als Ihre Anwendungsmanagement-Plattform einzuführen.
- Sie möchten die Funktionen der verwalteten Plattform von Elastic Beanstalk wie einfache Skalierung, Bereitstellung und Überwachung nutzen
- Sie bevorzugen eine saubere Bereitstellung, die den AWS Best Practices für Cloud-native Abläufe folgt
- Sie möchten den Konfigurationsaufwand nach der Migration minimieren

Wählen Sie aus AWS Application Migration Service , wann Sie die folgenden Anforderungen haben:

- Sie müssen eine große Anzahl von Servern migrieren
- Sie haben komplexe Serverkonfigurationen, die exakt beibehalten werden müssen
- Ihre Anwendungen weisen Kompatibilitätsprobleme auf, die die Aufrechterhaltung der exakten Serverumgebung erfordern
- Sie möchten ein Lift-and-Shift-Verfahren mit minimalen Änderungen an Ihren Anwendungen durchführen
- Sie planen, Ihre Anwendungen nach der Migration umzugestalten oder zu optimieren

Vergleich der Migrations-Workflows

EB CLI (eb migrate) -Arbeitsablauf:

1. Installieren Sie die EB-CLI entweder auf Ihrem IIS-Quellserver oder einem Bastion-Host.
2. Führen Sie das eb migrate Programm aus, um IIS-Anwendungen zu entdecken.
3. Der Befehl packt Ihre Anwendungen und Konfigurationen.
4. Eine Elastic Beanstalk Beanstalk-Umgebung wird mit entsprechenden Ressourcen erstellt.
5. Ihre Anwendungen werden in der neuen Umgebung bereitgestellt.
6. Mit den Elastic Beanstalk Beanstalk-Tools können Sie Ihre Anwendungen sofort verwalten.

AWS Application Migration Service Arbeitsablauf:

1. Installieren Sie den AWS Replication Agent auf den Quellservern.
2. Konfigurieren und testen Sie die Datenreplikation.
3. Starten Sie Testinstanzen, um die Funktionalität zu überprüfen.
4. Planen Sie die Umstellung auf. AWS
5. Starten Sie Produktionsinstanzen.
6. Führen Sie Aktionen nach dem Start aus, um sie für die Cloud zu optimieren.
7. Wenn Elastic Beanstalk die Zielplattform ist, ist für die Integration mit Elastic Beanstalk eine zusätzliche Konfiguration erforderlich.

Schlussfolgerung

Elastic Beanstalk ist das bevorzugte Ziel für Windows-Plattformanwendungen und bietet eine verwaltete Umgebung AWS, die die Bereitstellung, Skalierung und Verwaltung vereinfacht. Der eb migrate Befehl bietet einen direkten Pfad zu Elastic Beanstalk für IIS-Anwendungen mit automatischer Erkennung und Konfiguration, bei der Ihre Anwendungseinstellungen beibehalten werden.

Es AWS Application Migration Service bietet zwar leistungsstarke Funktionen für umfangreiche Servermigrationen, für die Integration mit Elastic Beanstalk sind jedoch zusätzliche Schritte erforderlich. Für die meisten IIS-Anwendungsmigrationen, bei denen Elastic Beanstalk die Zielplattform ist, eb migrate bietet dies einen optimierteren Ansatz, der dem Managed-Service-Modell von Elastic Beanstalk entspricht.

Wählen Sie den Migrationsansatz, der Ihren spezifischen Anforderungen am besten entspricht, und berücksichtigen Sie dabei Faktoren wie Umfang, Komplexität und die von Ihnen gewünschte Endstatusarchitektur. AWS

Weitere Informationen zu finden Sie AWS Application Migration Service unter Was ist AWS Application Migration Service? im AWS Application Migration Service Benutzerhandbuch.

Fehlerbehebung in Ihrer Elastic Beanstalk Beanstalk-Umgebung

Dieses Kapitel enthält Anleitungen zur Behebung von Problemen mit Ihrer Elastic-Beanstalk-Umgebung. Diese umfasst das Folgende:

- Eine Einführung in AWS Systems Manager, einschließlich eines Verfahrens zur Ausführung vordefinierter Elastic Beanstalk Beanstalk-Runbooks, die Schritte und Empfehlungen zur Fehlerbehebung ausgeben.
- Allgemeine Hinweise zu Aktionen und Ressourcen, die Sie verwenden können, wenn sich der Status Ihrer Umgebung verschlechtert.
- Spezifische Tipps zur Fehlerbehebung, geordnet nach Kategorien.

Wenn sich der Zustand Ihrer Umgebung auf Rot ändert, empfehlen wir Ihnen, zunächst das AWS Systems Manager Tool zu verwenden, das vordefinierte Runbooks enthält, um Probleme mit Elastic Beanstalk zu beheben. Weitere Informationen finden Sie unter Verwenden des Systems-Manager-Tools.

> **Testen Sie Amazon Q Developer CLI für KI-gestützte Problembehebung**
>
> Amazon Q Developer CLI kann Ihnen helfen, Umgebungsprobleme schnell zu beheben. Die Q CLI bietet Lösungen, indem sie den Umgebungsstatus überprüft, Ereignisse überprüft, Protokolle analysiert und klärende Fragen stellt. Weitere Informationen und detaillierte Anleitungen finden Sie in den Blogs unter Troubleshooting Elastic Beanstalk Environments with Amazon Q Developer CLI. AWS

Themen

- Mithilfe von KI-gestützter Umgebungsanalyse
- AWS Systems Manager Elastic Beanstalk Beanstalk-Runbooks verwenden
- Allgemeine Hinweise zur Fehlerbehebung in Ihrer Elastic Beanstalk Beanstalk-Umgebung
- Umgebungen, die mit Umgebungsvariablen auf Geheimnisse und Parameter zugreifen
- Umgebungserstellung und Instance-Starts
- Bereitstellungen

- Gesundheit
- Umgebungen mit konfigurierten Dual-Stack-Load Balancern
- Konfiguration
- Fehlerbehebung bei Docker-Containern
- Häufig gestellte Fragen
- Häufige Fehler
- Fehler bei der Bereitstellung

Mithilfe von KI-gestützter Umgebungsanalyse

Elastic Beanstalk bietet KI-gestützte Analysen, um die Grundursachen zu identifizieren und Lösungen für umweltbedingte Gesundheitsprobleme zu empfehlen. Sie können auf diese Funktion über die KI-Analyseschaltfläche der Konsole zugreifen, indem Sie die Elastic Beanstalk Beanstalk-API über die oder durch die AWS CLI Ausführung eb logs --analyze mit der EB-CLI verwenden. Bei der Analyse werden die Protokolle, Ereignisse und der Instanzstatus Ihrer Umgebung untersucht, um kritische Probleme zu identifizieren und Empfehlungen step-by-step zur Fehlerbehebung zu geben.

Weitere Informationen finden Sie unter KI-gestützte Umgebungsanalyse.

AWS Systems Manager Elastic Beanstalk Beanstalk-Runbooks verwenden

Sie können Systems Manager verwenden, um Probleme in Ihren Elastic-Beanstalk-Umgebungen zu beheben. Um Ihnen den Einstieg zu erleichtern, stellt Systems Manager vordefinierte Automatisierungs-Runbooks für Elastic Beanstalk bereit. Ein Automatisierungs-Runbook ist eine Art von Systems-Manager-Dokument, das Aktionen definiert, die auf den Instances und anderen AWS -Ressourcen Ihrer Umgebung ausgeführt werden sollen.

Bei dem Dokument `AWSSupport-TroubleshootElasticBeanstalk` handelt es sich um ein Automatisierungs-Runbook, mit dem Sie eine Reihe gängiger Probleme identifizieren können, die Ihre Elastic-Beanstalk-Umgebung beeinträchtigen können. Zu diesem Zweck werden Komponenten Ihrer Umgebung überprüft, darunter die folgenden: EC2-Instances, die VPC, der CloudFormation Stack, Load Balancer, Auto Scaling Scaling-Gruppen und die Netzwerkkonfiguration, die mit Sicherheitsgruppenregeln, Routentabellen und verknüpft ist. ACLs

Es gibt auch die Möglichkeit, gebündelte Protokolldateien aus Ihrer Umgebung zu AWS -Support hochzuladen.

Weitere Informationen finden Sie unter `AWSSupport-TroubleshootElasticBeanstalk` in der Referenz zum AWS Systems Manager -Automation-Runbook.

Verwenden Sie Systems Manager, um **`AWSSupport-TroubleshootElasticBeanstalk`** Runbook auszuführen

> **Note**
>
> Führen Sie dieses Verfahren in derselben Umgebung aus, in der AWS-Region sich Ihre Elastic Beanstalk Beanstalk-Umgebung befindet.

1. Öffnen Sie die AWS Systems Manager-Konsole.
2. Wählen Sie im Navigationsbereich im Abschnitt Änderungsmanagement Automatisierung aus.
3. Wählen Sie Automatisierung ausführen.
4. Auf der Registerkarte Owned by Amazon geben Sie in das Suchfeld für Automatisierungsdokumente `AWSSupport-TroubleshootElasticBeanstalk` ein.
5. Wählen Sie die AWSSupport-TroubleshootElasticBeanstalkKarte aus und klicken Sie dann auf Weiter.
6. Wählen Sie Ausführen aus.
7. Im Abschnitt Eingabeparameter:
   a. Wählen Sie aus der AutomationAssumeRoleDropdownliste den ARN der Rolle aus, mit der Systems Manager Aktionen in Ihrem Namen ausführen kann.
   b. Geben Sie für ApplicationNameden Namen der Elastic Beanstalk Beanstalk-Anwendung ein.
   c. Geben Sie als Umgebungsname die Elastic-Beanstalk-Umgebung ein.
   d. (Optional) Geben Sie für S3 einen Link einUploaderLink, wenn Ihnen ein AWS Support-Techniker einen S3-Link für die Protokollerfassung zur Verfügung gestellt hat.
8. Wählen Sie Ausführen.

   Wenn einer der Schritte fehlschlägt, wählen Sie den Link in der Spalte Schritt-ID für den Schritt aus, der fehlgeschlagen ist. Daraufhin wird eine Ausführungsdetailseite für den Schritt angezeigt. In VerificationErrorMessagediesem Abschnitt wird eine Zusammenfassung der Schritte angezeigt, die Aufmerksamkeit erfordern. `IAMPermissionCheck` könnte

zum Beispiel eine Warnmeldung anzeigen. In diesem Fall könnten Sie überprüfen, ob die in der AutomationAssumeRoleDropdownliste ausgewählte Rolle über die erforderlichen Berechtigungen verfügt.

Nachdem alle Schritte erfolgreich abgeschlossen wurden, enthält die Ausgabe Schritte zur Fehlerbehebung und Empfehlungen, mit denen Sie Ihre Umgebung wieder in einen einwandfreien Zustand versetzen können.

Allgemeine Hinweise zur Fehlerbehebung in Ihrer Elastic Beanstalk Beanstalk-Umgebung

Fehlermeldungen können auf der Seite Ereignisse in der Konsole, in Protokollen oder auf der Seite Zustand erscheinen. Sie können auch Maßnahmen ergreifen, um eine beeinträchtigte Umgebung wiederherzustellen, die durch eine kürzlich erfolgte Änderung verursacht wurde. Ändert sich der Zustand der Umgebung zu Rot, führen Sie die folgenden Schritte aus:

- Fordern Sie eine KI-gestützte Analyse Ihrer Umgebung an. Weitere Informationen finden Sie unter KI-gestützte Umgebungsanalyse.
- Wenn bei einem Vorgang in Ihrer Umgebung ein Fehler zurückgegeben wird, der den Text enthältThe `stack` *`stack_id`* `associated with environment` *`environment-ID`* `is in` *`stack-status`* `state`, finden Sie Hilfe Wiederherstellung Ihrer Elastic Beanstalk Beanstalk-Umgebung aus einem ungültigen Zustand zur Fehlerbehebung unter.
- Wenn ein Vorgang in Ihrer Umgebung einen Fehler zurückgibt, der den Text enthält`Environment` *`environment-name`* `associated CloudFormation stack` *`stack_arn`* `does not exist`, beenden Sie Ihre Umgebung und erstellen Sie eine neue.
- Überprüfen Sie die letzten Ereignisse in der Umgebung. Hier finden Sie häufig Meldungen von Elastic Beanstalk zu Bereitstellungs-, Auslastungs- und Konfigurationsproblemen.
- Überprüfen Sie den aktuellen Umweltänderungsverlauf. Der Änderungsverlauf listet alle Konfigurationsänderungen auf, die an Ihren Umgebungen vorgenommen wurden, und enthält weitere Informationen, z. B. welche IAM-Benutzer Änderungen vorgenommen hat und welche Konfigurationsparameter festgelegt wurden.
- Rufen Sie die Protokolle auf und sehen Sie sich die letzten Protokolldateieinträge an. Die Webserverprotokolle können Informationen zu eingehenden Anforderungen und Fehlern enthalten.
- Stellen Sie die Verbindung zu einer Instance her und überprüfen Sie die Systemressourcen.
- Stellen Sie eine vorherige funktionierende Version der Anwendung wieder her.

- Machen Sie die letzten Konfigurationsänderungen rückgängig oder stellen Sie eine gespeicherte Konfiguration wieder her.
- Stellen Sie eine neue Umgebung bereit. Wenn diese Umgebung stabil wirkt, führen Sie einen CNAME-Austausch durch, um den Datenverkehr in die neue Umgebung umzuleiten, und setzen Sie die Fehlerbehebung in der vorherigen Umgebung fort.

Umgebungen, die mit Umgebungsvariablen auf Geheimnisse und Parameter zugreifen

Ereignis: Bei der Instanzbereitstellung konnten ein oder mehrere Geheimnisse nicht abgerufen werden

Diese Meldung weist darauf hin, dass Elastic Beanstalk eines oder mehrere der während der Anwendungsbereitstellung angegebenen Geheimnisse nicht abrufen konnte.

- Überprüfen Sie, ob die durch die ARN-Werte in Ihrer Umgebungsvariablenkonfiguration angegebenen Ressourcen vorhanden sind.
- Vergewissern Sie sich, dass Ihre Elastic Beanstalk EC2-Instance-Profilrolle über die erforderlichen IAM-Berechtigungen für den Zugriff auf die Ressourcen verfügt.
- Wenn dieses Ereignis durch den `RestartAppServer` Vorgang ausgelöst wurde, versuchen Sie nach der Behebung des Problems erneut, den `RestartAppServer` Anruf zu starten, um das Problem zu lösen.
- Wenn das Ereignis durch einen `UpdateEnvironment` Anruf ausgelöst wurde, wiederholen Sie den `UpdateEnvironment` Vorgang.

Beispiele für diese Befehle finden Sie unter AWS CLI Beispiele für Elastic Beanstalk. Weitere Informationen zu den API-Aktionen für diese Operationen finden Sie in der AWS Elastic Beanstalk API-Referenz.

Ereignis: Bei der Instanzbereitstellung wurden ein oder mehrere mehrzeilige Umgebungswerte erkannt, die für diese Plattform nicht unterstützt werden

Mehrzeilige Variablen werden für Amazon Linux 2-Plattformen nicht unterstützt, ausgenommen Docker- und ECS-verwaltete Docker-Plattformen. Die verfügbaren Optionen, um fortzufahren, finden Sie unter. Mehrzeilige Werte

Ereignis: CreateEnvironment schlägt fehl, wenn ein Geheimnis angegeben wird

Wenn ein `CreateEnvironment` Fehler auftritt und Sie Geheimnisse als Umgebungsvariablen haben, müssen Sie das zugrundeliegende Problem beheben und es dann verwenden`UpdateEnvironment`, um das Umgebungs-Setup abzuschließen. Verwenden Sie es nicht`RestartAppServer`, da es in dieser Situation nicht ausreicht, um die Umgebung zum Laufen zu bringen. Beispiele für diese Befehle finden Sie unter AWS CLI Beispiele für Elastic Beanstalk. Weitere Informationen zu den API-Aktionen für diese Operationen finden Sie in der AWS Elastic Beanstalk API-Referenz.

Umgebungserstellung und Instance-Starts

Ereignis: Umgebung konnte nicht gestartet werden

Dieses Ereignis tritt auf, wenn Elastic Beanstalk versucht, eine Umgebung zu starten und es dabei zu Fehlern kommt. Die vorherigen Ereignisse auf der Seite Events (Ereignisse) weisen auf die Ursache des Problems hin.

Ereignis: Die Erstellung der Umgebung wurde mit Befehls-Timeouts abgeschlossen. Versuchen Sie, die Timeout-Zeit zu erhöhen.

Falls Sie Konfigurationsdateien einsetzen, mit denen Befehle auf der Instance ausgeführt, große Dateien heruntergeladen oder Pakete installiert werden, kann die Anwendungsbereitstellung langwierig sein. Erhöhen Sie den Timeout-Wert für Befehle, damit die Anwendung im Rahmen der Bereitstellungen mehr Zeit zum Ausführen hat.

Ereignis: Die folgenden Ressourcen konnten nicht erstellt werden: [AWSEBInstanceLaunchWaitCondition]

Diese Nachricht gibt an, dass die Amazon EC2-Instances der Umgebung Elastic Beanstalk nicht über den erfolgreichen Start informiert haben. Das kann passieren, wenn die Instances keine Verbindung zum Internet haben. Wenn die Umgebungskonfiguration das Starten der Instances in einem privaten VPC-Subnetz vorsieht, stellen Sie sicher, dass das Subnetz über NAT verfügt, damit die Instances eine Verbindung zu Elastic Beanstalk herstellen können.

Ereignis: In dieser Region ist eine Servicerolle erforderlich. Bitte fügen Sie der Umgebung eine Servicerolle-Option hinzu.

Elastic Beanstalk nutzt eine Servicerolle zur Überwachung der Ressourcen in Ihrer Umgebung und zur Unterstützung von verwalteten Plattformaktualisierungen. Weitere Informationen finden Sie unter Rollen von Elastic Beanstalk Service verwalten.

Bereitstellungen

Problem: Anwendung steht während Bereitstellungen nicht zur Verfügung

Da Elastic Beanstalk ein Drop-in-Upgrade ausführt, kann es zu einer Ausfallzeit von einigen Sekunden kommen. Setzen Sie fortlaufende Bereitstellungen ein, um die Beeinträchtigung der Produktionsumgebungen aufgrund von Bereitstellungen zu minimieren.

Ereignis: Die AWS Elastic Beanstalk Beanstalk-Anwendungsversion konnte nicht erstellt werden

Möglicherweise ist das Quell-Bundle der Anwendung zu umfangreich oder das Anwendungsversionskontingent wurde erreicht.

Ereignis: Umgebungsaktualisierung wurde mit Befehls-Timeouts abgeschlossen. Versuchen Sie, die Timeout-Zeit zu erhöhen.

Falls Sie Konfigurationsdateien einsetzen, mit denen Befehle auf der Instance ausgeführt, große Dateien heruntergeladen oder Pakete installiert werden, kann die Anwendungsbereitstellung langwierig sein. Erhöhen Sie den Timeout-Wert für Befehle, damit die Anwendung im Rahmen der Bereitstellungen mehr Zeit zum Ausführen hat.

Gesundheit

Ereignis: CPU-Auslastung Über 95 %

Versuchen Sie, mehr Instances auszuführen, oder wählen Sie einen anderen Instance-Typ aus.

Ereignis: Elastic Load Balancer awseb — *myapp* Hat Zero Healthy Instances, ELB-Prozesse sind auf *X* Out-of-Instances nicht fehlerfrei. *Y* , ELB-Prozesse sind nicht auf allen Instances fehlerfrei.

Diese Meldungen weisen darauf hin, dass Ihre Load Balancer-Zustandsprüfungen bei einer oder mehreren Instances fehlschlagen. Wenn Ihre Anwendung zu funktionieren scheint, stellen Sie sicher, dass die URL für die Integritätsprüfung Ihrer Anwendung korrekt konfiguriert ist. Falls nicht, versuchen Sie Folgendes:

- Überprüfen Sie Ihre Anwendungsprotokolle (z. B./`var/log/web.stdout.log`) und `/var/log/eb-engine.log` auf Fehler, Startfehler oder Bereitstellungsprobleme.
- Überprüfen Sie die Proxyprotokolle in `/var/log/nginx/` (nginx) oder `/var/log/httpd/` (Apache) auf Verbindungsfehler.

- Stellen Sie sicher, dass Ihre Anwendung mit HTTP 200 auf den konfigurierten Health Check-Pfad reagiert und den erwarteten Port abhört.
- Stellen Sie sicher, dass Sicherheitsgruppen eingehenden Datenverkehr vom Load Balancer zu Instances am Health Check-Port zulassen. Überprüfen Sie in der EC2-Konsole die Regeln für eingehenden Datenverkehr in der Sicherheitsgruppe der Instance, um sicherzustellen, dass der Health Check-Port für die Sicherheitsgruppe des Load Balancers geöffnet ist.
- Überprüfen Sie die CPU- und Speicherauslastung anhand von CloudWatch Metriken (`CPUUtilization`,). `EnvironmentHealth` Eine Erschöpfung der Ressourcen kann zu Timeouts bei der Integritätsprüfung führen.

Rufen Sie für weitere Informationen die vollständigen Paketprotokolle ab oder aktivieren CloudWatch Sie das Protokollstreaming. Sie können auch das `AWSSupport-TroubleshootElasticBeanstalk`Automatisierungs-Runbook ausführen, um das Problem zu diagnostizieren.

Ereignis: Elastic Load Balancer awseb- *`myapp`* kann nicht gefunden werden

Der Load Balancer Ihrer Umgebung wurde möglicherweise entfernt. out-of-band Nehmen Sie Änderungen an den Umgebungsressourcen nur mit den von Elastic Beanstalk bereitgestellten Konfigurationsoptionen und Erweiterungen vor. Erstellen Sie Ihre Umgebung erneut oder starten Sie eine neue Umgebung.

Ereignis: EC2-Instance konnte nicht gestartet werden. Warten auf den Start einer neuen EC2 Instance ...

Möglicherweise sind derzeit keine Instance-Typen für die Umgebung verfügbar oder das Instance-Kontingent für Ihr Konto wurde erreicht. Überprüfen Sie in der Übersicht zum Servicestatus-, ob der Elastic Compute Cloud (Amazon EC2)-Service den Status "Grün" aufweist, oder fordern Sie eine Kontingenterhöhung an.

Ereignis: Keine der Instanzen sendet Daten.

Elastic Beanstalk empfängt von keiner Ihrer Instances Gesundheitsdaten. Dies kann durch Netzwerkverbindungsprobleme, Probleme mit Health Agents oder Ausfälle auf Systemebene verursacht werden.

- Überprüfen Sie die Netzwerkkonnektivität. Vergewissern Sie sich, dass Sicherheitsgruppen ausgehendes HTTPS (Port 443) zulassen. Stellen Sie für Instances in privaten Subnetzen sicher,

dass ein NAT-Gateway oder VPC-Endpunkte konfiguriert sind. Überprüfen Sie die Routentabellen auf korrekte ausgehende Routen und stellen Sie sicher, dass die DNS-Auflösung in Ihrer VPC ordnungsgemäß funktioniert.

- Überprüfen Sie den Health Monitoring Agent. Suchen Sie nach Fehlern bei `/var/log/healthd/daemon.log` Health Agents und `/var/log/messages` nach Fehlern auf Systemebene. Stellen Sie sicher, dass der Healthd-Dienst ausgeführt wird, indem Sie sich über SSH mit der Instanz verbinden und ihn ausführen. `systemctl status healthd`
- Prüfen Sie, ob Probleme auf Systemebene vorliegen, wie z. B. Ressourcenerschöpfung (Arbeitsspeicher, CPU, Festplattenspeicher), NTP/system Uhrsynchronisierung und Konnektivität des Instanz-Metadatendienstes.

Rufen Sie für weitere Informationen die vollständigen Paketprotokolle ab oder aktivieren CloudWatch Sie das Protokollstreaming. Sie können auch das `AWSSupport-TroubleshootElasticBeanstalk`Automatisierungs-Runbook ausführen, um das Problem zu diagnostizieren.

Umgebungen mit konfigurierten Dual-Stack-Load Balancern

Ereignis: In der VPC sind *`vpc_id`* keine IPv6 CIDR-Blöcke konfiguriert. IPv6 CIDR-Blöcke sind für den Dual-Stack-Load Balancer erforderlich. Bitte ordnen Sie Ihrer VPC einen IPv6 CIDR-Block zu, bevor Sie den Dual-Stack-Modus verwenden.

Ihrer VPC und allen Subnetzen müssen IPv6 CIDR-Blöcke zugeordnet sein. Dies ist eine der VPC-Voraussetzungen, die Sie erfüllen müssen, bevor Sie Ihren Load Balancer für die Dual-Stack-Unterstützung konfigurieren können. Weitere Informationen zur Ausführung dieser Aufgabe finden Sie weiter Voraussetzungen für Amazon VPC oben in diesem Thema.

Ereignis: In einem oder mehreren Subnetzen für VPC sind *`vpc_id`* keine IPv6 CIDR-Blöcke konfiguriert. IPv6 CIDR-Blöcke sind für die Subnetze erforderlich, die mit dem Dualstack Load Balancer verwendet werden. Bitte ordnen Sie IPv6 CIDR-Blöcke allen erforderlichen Subnetzen zu, bevor Sie den Dual-Stack-Modus verwenden.

Allen Subnetzen für Ihre VPC müssen IPv6 CIDR-Blöcke zugeordnet sein. Dies ist eine der VPC-Voraussetzungen, die Sie erfüllen müssen, bevor Sie Ihren Load Balancer für die Dual-Stack-Unterstützung konfigurieren können. Weitere Informationen zur Ausführung dieser Aufgabe finden Sie weiter Voraussetzungen für Amazon VPC oben in diesem Thema.

Fehler: Die *IpAddressType* Option kann nur auf Elastic Beanstalk Beanstalk-Umgebungen angewendet werden, die mit einem Application Load Balancer oder Network Load Balancer konfiguriert sind.

Diese Meldung weist darauf hin, dass es sich bei Ihrer Elastic Beanstalk Beanstalk-Umgebung möglicherweise um eine Single-Instance-Umgebung handelt oder dass sie einen Classic Load Balancer verwendet. Nur Umgebungen, die mit einem Application Load Balancer oder Network Load Balancer konfiguriert sind, können konfiguriert werden. `IpAddressType`

Konfiguration

Ereignis: Der der Umgebung *stack_id* zugeordnete Stack *environment-ID* befindet sich im Status *stack-status*

Der zugrunde liegende CloudFormation Stack Ihrer Umgebung befindet sich möglicherweise im Status *_FAILED. Dieser Status muss behoben werden, um den Elastic Beanstalk Beanstalk-Betrieb in Ihrer Umgebung fortsetzen zu können. Weitere Informationen finden Sie unter Wiederherstellung Ihrer Elastic Beanstalk Beanstalk-Umgebung aus einem ungültigen Zustand.

Ereignis: Es kann keine Elastic Beanstalk-Umgebung konfiguriert werden, die Werte für die Optionen "Elastic Load Balancing Target (Elastic Load Balancing-Ziel)" und "Application Healthcheck URL (URL für die Zustandsprüfung der Anwendung)" enthält.

Die Option `Target` im Namespace `aws:elb:healthcheck` ist veraltet. Entfernen Sie die Namespace-Option `Target` aus der Umgebung und führen Sie die Aktualisierung erneut aus.

Ereignis: ELB kann nicht an mehrere Subnetze in derselben AZ angehängt werden

Diese Meldung wird angezeigt, wenn Sie einen Load Balancer zwischen Subnetzen in derselben Availability Zone verschieben möchten. Sollen die Subnetze des Load Balancers geändert werden, muss dieser aus den ursprünglichen Availability Zones entfernt und mit den gewünschten Subnetzen wieder in diese eingefügt werden. Während des Vorgangs werden alle Ihre Instances zwischen diesen migriert AZs, was zu erheblichen Ausfallzeiten führt. Ziehen Sie stattdessen in Betracht, eine neue Umgebung zu erstellen und einen CNAME-Austausch durchzuführen.

Fehlerbehebung bei Docker-Containern

Ereignis: Docker-Image :latest: konnte nicht abgerufen werden. Ungültiger Repository-Name (), nur [a-z0-9-] sind zulässig. Weitere Details finden Sie in den Protokollfragmenten.

Prüfen Sie die Syntax der `dockerrun.aws.json`-Datei mit einer JSON-Validierung. Verifizieren Sie außerdem den dockerfile-Inhalt hinsichtlich der Anforderungen in Vorbereiten Ihres Docker-Images für die Bereitstellung auf Elastic Beanstalk.

Ereignis: Keine EXPOSE-Anweisung in Dockerfile gefunden, Bereitstellung abbrechen

Die `Dockerfile` oder `dockerrun.aws.json`-Datei deklariert den Container-Port nicht. Verwenden Sie die `EXPOSE`-Anweisung (`Dockerfile`) oder den `Ports`-Block (`dockerrun.aws.json`-Datei), um einen Port für eingehenden Datenverkehr verfügbar zu machen.

Ereignis: Authentifizierungsdaten *`repository`* konnten nicht heruntergeladen werden von *`bucket name`*

Der `dockerrun.aws.json` stellt ein ungültiges EC2-Schlüsselpaar and/or S3-Bucket für die `.dockercfg` Datei bereit. Oder das Instance-Profil hat keine GetObject Autorisierung für den S3-Bucket. Überprüfen Sie, ob die `.dockercfg`-Datei einen gültigen S3-Bucket und ein gültiges EC2-Schlüsselpaar enthält. Erteilen Sie Berechtigungen für die Aktion `s3:GetObject` für die IAM-Rolle im Instance-Profil. Details finden Sie unter Elastic Beanstalk Instance-Profile verwalten.

Ereignis: Aktivitätsausführung wegen Warnmeldung zur ungültigen "auth"-Konfigurationsdatei fehlgeschlagen

Die Authentifizierungsdatei (`config.json`) ist nicht ordnungsgemäß formatiert. Siehe Authentifizierung mit Image-Repositorys

Häufig gestellte Fragen

Frage: Wie kann ich meine Anwendungs-URL von myapp.us-west-2.elasticbeanstalk.com in www.myapp.com ändern?

Registrieren Sie bei einem DNS-Server einen CNAME-Datensatz, wie z. B. **`www.mydomain.com CNAME mydomain.elasticbeanstalk.com`**.

Frage: Wie gebe ich eine bestimmte Availability Zone für meine Elastic Beanstalk-Anwendung an?

Sie können eine bestimmte Availability Zone auswählen, indem Sie das CLI APIs, das Eclipse-Plug-In oder das Visual Studio-Plugin verwenden. Für Anweisungen zum Angeben einer Availability Zone mit der Elastic Beanstalk-Konsole siehe Auto Scaling Ihrer Elastic Beanstalk Beanstalk-Umgebungsinstanzen.

Frage: Wie kann ich den Instance-Typ meiner Umgebung ändern?

Um den Instance-Typ Ihrer Umgebung zu ändern, rufen Sie die Seite Bearbeiten in der Konfigurationskategorie Instances auf. Wählen Sie dann einen neuen Instance-Typ aus und wählen Sie Anwenden, um Ihre Umgebung zu aktualisieren. Danach beendet Elastic Beanstalk alle laufenden Instances und ersetzt sie durch neue.

Frage: Wie stelle ich fest, ob jemand Konfigurationsänderungen an einer Umgebung vorgenommen hat?

Um diese Informationen anzuzeigen, wählen Sie im Navigationsbereich der Elastic-Beanstalk-Konsole den Änderungsverlauf, um eine Liste der Konfigurationsänderungen für alle Umgebungen anzuzeigen. Diese Liste enthält das Datum und die Uhrzeit der Änderung, den Konfigurationsparameter und den Wert, in den sie geändert wurde, sowie den IAM-Benutzer, der die Änderung vorgenommen hat. Weitere Informationen finden Sie unter Änderungsverlauf.

Frage: Kann ich verhindern, dass Amazon EBS-Volumes gelöscht werden, wenn Instances beendet werden?

Instances in Ihrer Umgebung verwenden Amazon EBS zur Speicherung. Das Stamm-Volume wird jedoch gelöscht, wenn eine Instance durch Auto Scaling beendet wird. Wir empfehlen nicht, dass Sie Status- oder andere Daten auf Ihren Instances speichern. Bei Bedarf können Sie verhindern, dass Volumes gelöscht werden, indem Sie AWS CLI: verwenden, `$ aws ec2 modify-instance-attribute -b '/dev/sdc=<vol-id>:false` wie in der AWS CLI Referenz beschrieben.

Frage: Wie lösche ich persönliche Informationen aus meiner Elastic Beanstalk-Anwendung?

AWS Ressourcen, die Ihre Elastic Beanstalk Beanstalk-Anwendung verwendet, speichern möglicherweise persönliche Informationen. Wenn Sie eine Umgebung beenden, beendet Elastic Beanstalk die erstellten Ressourcen. Die Ressourcen, die Sie mithilfe von Konfigurationsdateien hinzugefügt haben, werden ebenfalls beendet. Wenn Sie jedoch AWS Ressourcen außerhalb Ihrer Elastic Beanstalk Beanstalk-Umgebung erstellt und sie mit Ihrer Anwendung verknüpft haben, müssen Sie möglicherweise manuell überprüfen, ob persönliche Informationen, die Ihre Anwendung möglicherweise gespeichert hat, nicht beibehalten werden. In diesem Entwicklerhandbuch erwähnen wir, wann immer wir über die Erstellung zusätzlicher Ressourcen sprechen, auch, wann Sie darüber nachdenken sollten, diese zu löschen.

Häufige Fehler

In diesem Thema werden häufig auftretende Fehlermeldungen im Zusammenhang mit der EB CLI-Verwendung sowie mögliche Lösungen aufgeführt. Falls Ihnen eine Fehlermeldung angezeigt wird, die hier nicht enthalten ist, teilen Sie uns dies bitte über die Feedback-Links mit.

FEHLER: Bei der Verarbeitung des Befehls git ist ein Fehler aufgetreten. Fehlercode: 128 Fehler: fatal: Kein gültiger Objektname HEAD

Ursache: Diese Fehlermeldung wird angezeigt, wenn Sie ein Git-Repository initialisiert, aber noch nicht übergeben haben. Wenn der Projektordner ein Git-Repository enthält, sucht die EB CLI nach einer HEAD-Revision.

Lösung: Fügen Sie die Dateien in Ihrem Projektordner zum Staging-Bereich hinzu und führen Sie den Commit aus:

```
~/my-app$ git add .
~/my-app$ git commit -m "First commit"
```

FEHLER: Dieser Zweig hat keine Standardumgebung. Sie müssen entweder eine Umgebung angeben, indem Sie „eb status my-env-name“ eingeben, oder eine Standardumgebung festlegen, indem Sie „eb use my-env-name“ eingeben.

Ursache: Wenn Sie in Git eine neue Verzweigung erstellen, wird dieser nicht standardmäßig einer Elastic Beanstalk-Umgebung zugewiesen.

Lösung: Führen Sie eb list aus, um eine Liste der verfügbaren Umgebungen aufzurufen. Führen Sie anschließend eb use ***env-name*** aus, um eine der verfügbaren Umgebungen zu verwenden.

FEHLER: Für Plattformen ab Version 2.0 ist eine Servicerolle erforderlich. Sie können eine mit der Option --service-role angeben

Ursache: Wenn Sie einen Umgebungsnamen mit eb create angeben (z. B. eb create my-env), erstellt die EB CLI keine Servicerolle für Sie. Ist die standardmäßige Servicerolle nicht vorhanden, wird die obige Fehlermeldung angezeigt.

Lösung: Führen Sie eb create ohne den Umgebungsnamen aus und folgen Sie den Anweisungen, um die standardmäßige Servicerolle zu erstellen.

Fehler bei der Bereitstellung

Ihr Elastic Beanstalk-Deployment reagiert möglicherweise mit einer 404-Antwort (wenn Ihre Anwendung nicht gestartet werden konnte) oder einer 500-Antwort (wenn Ihre Anwendung während der Laufzeit fehlschlägt). Um viele häufig auftretende Probleme zu beheben, können Sie die EB CLI verwenden, um den Status Ihrer Bereitstellung zu überprüfen, die zugehörigen Protokolle einzusehen, mit SSH auf Ihre EC2-Instance zuzugreifen oder die AWS Management Console-Seite für Ihre Anwendungsumgebung zu öffnen.

So setzen Sie die EB CLI zum Beheben von Bereitstellungsfehlern ein

1. Führen Sie eb status aus, um den Status der aktuellen Bereitstellung sowie den Zustand der EC2-Hosts anzuzeigen. Beispiel:

```
$ eb status --verbose

Environment details for: python_eb_app
  Application name: python_eb_app
  Region: us-west-2
  Deployed Version: app-150206_035343
  Environment ID: e-wa8u6rrmqy
  Platform: 64bit Amazon Linux 2014.09 v1.1.0 running Python 2.7
  Tier: WebServer-Standard-
  CNAME: python_eb_app.elasticbeanstalk.com
  Updated: 2015-02-06 12:00:08.557000+00:00
  Status: Ready
  Health: Green
  Running instances: 1
      i-8000528c: InService
```

> **Note**
>
> Mit `--verbose` erhalten Sie detaillierte Informationen über den Status der ausgeführten Instances. Andernfalls werden nur allgemeine Umgebungsinformationen von eb status ausgegeben.

2. Führen Sie eb health aus, um Zustandsinformationen zur Umgebung anzuzeigen:

```
$ eb health --refresh
```

```
 elasticBeanstalkExa-env                                    Degraded
 2016-03-28 23:13:20
WebServer
    Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        2        0        2        1        0        0        0

  instance-id   status     cause
    Overall     Degraded  Incorrect application version found on 3 out of 5
 instances. Expected version "Sample Application" (deployment 1).
  i-d581497d    Degraded  Incorrect application version "v2" (deployment 2).
 Expected version "Sample Application" (deployment 1).
  i-d481497c    Degraded  Incorrect application version "v2" (deployment 2).
 Expected version "Sample Application" (deployment 1).
  i-136e00c0    Severe    Instance ELB health has not been available for 5 minutes.
  i-126e00c1    Ok
  i-8b2cf575    Ok

  instance-id   r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75
 p50     p10
    Overall     646.7   100.0    0.0    0.0    0.0    0.003    0.002    0.001
 0.001   0.000
  i-dac3f859    167.5    1675      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-05013a81    161.2    1612      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-04013a80    0.0         -      -      -      -        -        -       -
 -       -
  i-3ab524a1    155.9    1559      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-bf300d3c    162.1    1621      0      0      0    0.003    0.002    0.001
 0.001   0.000

  instance-id   type       az   running     load 1  load 5      user%  nice%
 system%  idle%   iowait%
  i-d581497d    t2.micro   1a   25 mins       0.16     0.1        7.0    0.0
 1.7   91.0       0.1
  i-d481497c    t2.micro   1a   25 mins       0.14     0.1        7.2    0.0
 1.6   91.1       0.0
  i-136e00c0    t2.micro   1b   25 mins        0.0    0.01        0.0    0.0
 0.0   99.9       0.1
  i-126e00c1    t2.micro   1b   25 mins       0.03    0.08        6.9    0.0
 2.1   90.7       0.1
```

```
  i-8b2cf575    t2.micro   1c   1 hour        0.05    0.41        6.9    0.0
2.0   90.9       0.0

  instance-id   status     id   version              ago
      deployments
  i-d581497d    Deployed   2    v2                   9 mins
  i-d481497c    Deployed   2    v2                   7 mins
  i-136e00c0    Failed     2    v2                   5 mins
  i-126e00c1    Deployed   1    Sample Application   25 mins
  i-8b2cf575    Deployed   1    Sample Application   1 hour
```

Im obigen Beispiel wird eine Umgebung mit fünf Instances dargestellt, in der die Bereitstellung der Version "v2" auf der dritten Instance fehlerhaft verlaufen ist. Nach der fehlgeschlagenen Bereitstellung wird die erwartete Version auf die letzte erfolgreiche Version zurückgesetzt, in diesem Falle auf "Sample Application" aus der ersten Bereitstellung. Weitere Informationen finden Sie unter Verwenden der EB-CLI zur Überwachung des Umgebungszustands.

3. Führen Sie eb logs aus, um die Protokolle zur Anwendungsbereitstellung herunterzuladen und aufzurufen.

```
$ eb logs
```

4. Führen Sie eb ssh aus, um die Verbindung zur EC2-Instance, auf der die Anwendung ausgeführt wird, herzustellen und diese direkt zu überprüfen. Auf der Instance finden Sie die bereitgestellte Anwendung im Verzeichnis `/opt/python/current/app` und die Python-Umgebung in `/opt/python/run/venv/`.
5. Führen Sie eb console aus, um die Anwendungsumgebung in der AWS -Managementkonsole anzuzeigen. Über die Weboberfläche können Sie schnell zahlreiche Bereitstellungsaspekte untersuchen, darunter Konfiguration, Status, Ereignisse und Protokolle der Anwendung. Außerdem können Sie die aktuellen oder beendeten Anwendungsversionen herunterladen, die Sie auf dem Server bereitgestellt haben.

Elastic Beanstalk-Ressourcen

Die folgenden verwandten Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit diesem Service.

- Elastic Beanstalk API-Referenz Eine umfassende Beschreibung aller SOAP- und Query-Funktionen. APIs Zudem ist eine Liste aller SOAP-Datentypen enthalten.
- elastic-beanstalk-samples on GitHub — Ein GitHub Repository mit Elastic Beanstalk Beanstalk-Beispielkonfigurationsdateien (.ebextensions). Die `README.md` Datei des Repositorys enthält Links zu weiteren GitHub Repositorys mit Beispielanwendungen.
- Häufig gestellte technische Fragen zu Elastic Beanstalk – Die wichtigsten Fragen, die Entwickler zu diesem Produkt gestellt haben.
- AWS Elastic Beanstalk Versionshinweise — Einzelheiten zu neuen Funktionen, Updates und Fixes in den Versionen Service, Plattform, Konsole und EB CLI von Elastic Beanstalk.

- Kurse und Workshops — Links zu rollen- und Spezialkursen sowie zu Übungen zum Selbststudium, mit denen Sie Ihre Fähigkeiten verbessern und praktische Erfahrungen sammeln können. AWS
- AWS Developer Center — Erkunden Sie Tutorials, laden Sie Tools herunter und erfahren Sie mehr über Veranstaltungen für Entwickler. AWS
- AWS Entwicklertools — Links zu Entwicklertools SDKs, IDE-Toolkits und Befehlszeilentools für die Entwicklung und Verwaltung von AWS Anwendungen.
- Ressourcencenter für die ersten Schritte — Erfahren Sie AWS-Konto, wie Sie Ihre erste Anwendung einrichten, der AWS Community beitreten und sie starten.
- Praktische Tutorials — Folgen Sie den step-by-step Tutorials, um Ihre erste Anwendung zu starten. AWS
- AWS Whitepapers — Links zu einer umfassenden Liste von technischen AWS Whitepapers zu Themen wie Architektur, Sicherheit und Wirtschaft, die von Solutions Architects oder anderen technischen Experten verfasst wurden. AWS
- AWS Support Center — Die zentrale Anlaufstelle für die Erstellung und Verwaltung Ihrer Fälle. AWS Support Enthält auch Links zu anderen hilfreichen Ressourcen wie Foren, technischen FAQs Informationen, Servicestatus und AWS Trusted Advisor.

- Support— Die wichtigste Webseite mit Informationen über Support einen Support-Kanal mit schnellen Reaktionszeiten one-on-one, der Sie bei der Entwicklung und Ausführung von Anwendungen in der Cloud unterstützt.
- Kontakt – Zentraler Kontaktpunkt für Fragen zu AWS -Abrechnung, Konten, Ereignissen Missbrauch und anderen Problemen.
- AWS Nutzungsbedingungen der Website — Detaillierte Informationen zu unseren Urheberrechten und Marken, zu Ihrem Konto, Ihrer Lizenz und Ihrem Zugriff auf die Website sowie zu anderen Themen.

Beispielanwendungen

Nachfolgend finden Sie Links zu den herunterladbaren Beispielanwendungen, die unter Erfahren Sie, wie Sie mit Elastic Beanstalk loslegen können bereitgestellt werden.

> **Note**
>
> In einigen Beispielen werden Funktionen verwendet, die möglicherweise nach der von Ihnen verwendeten Plattformversion veröffentlicht wurden. Falls das Beispiel nicht ausgeführt werden kann, versuchen Sie, Ihre Plattform wie unter the section called “Unterstützte Plattformen” beschrieben auf die neueste Version zu aktualisieren.

- Docker – docker.zip
- Multicontainer Docker — 2.zip docker-multicontainer-v
- Vorkonfigurierter Docker (Glassfish) — 1.zip docker-glassfish-v
- Go – go.zip
- Corretto – corretto.zip
- Tomcat – tomcat.zip
- .NET Core unter Linux — .zip dotnet-core-linux
- .NET Core — dotnet-asp-windows .zip
- Node.js – nodejs.zip
- PHP – php.zip
- Python – python.zip

- Ruby – ruby.zip

AWS SDK für Java

Das AWS SDK für Java stellt eine Java-API bereit, mit der Sie Anwendungen erstellen können, die AWS Infrastrukturdienste nutzen. Mit dem AWS SDK für Java können Sie innerhalb von Minuten mit einem einzigen herunterladbaren Paket loslegen, das die AWS Java-Bibliothek, Codebeispiele und Dokumentation enthält.

Das AWS SDK für Java erfordert das J2SE Development Kit. Sie können die neueste Java-Software unter http://developers.sun.com/downloads/ herunterladen. Das SDK benötigt außerdem Apache Commons (Codec HTTPClient, und Logging) und Saxon-HE-Drittanbieterpakete, die im Drittanbieter-Verzeichnis des SDK enthalten sind.

Weitere Informationen finden Sie unter AWS SDK for Java. Weitere Informationen zum SDK, Beispielcode, Dokumentation, Tools und zusätzliche Ressourcen finden Sie im Java on AWS Developer Center.

AWS SDK für .NET

Mit dem AWS SDK für .NET können Sie schnell mit einem einzigen herunterladbaren Paket beginnen, das die AWS .NET-Bibliothek, Codebeispiele und Dokumentation enthält.

Weitere Informationen finden Sie unter AWS SDK for .NET. Die unterstützten .NET Framework- und Visual Studio-Versionen finden Sie im AWS SDK für .NET -Entwicklerhandbuch.

Weitere Informationen zum SDK, Beispielcode, Dokumentation, Tools und zusätzliche Ressourcen finden Sie unter.NET on AWS Developer Center.

AWS Toolkit for Visual Studio

Mit dem AWS Toolkit for Visual Studio Plug-in können Sie eine vorhandene.NET-Anwendung auf Elastic Beanstalk bereitstellen. Sie können Projekte auch mithilfe der AWS Vorlagen erstellen, die mit dem vorkonfiguriert sind. AWS SDK für .NET

Weitere Informationen zu Voraussetzungen und zur Installation finden Sie unter AWS Toolkit for Visual Studio. Informationen zum Erstellen Ihrer Elastic Beanstalk-Anwendung mit Visual Studio finden Sie unter Bereitstellen von.NET-Windows-Anwendungen mit Elastic Beanstalk.

AWS SDK für JavaScript in Node.js

Mit dem AWS SDK für JavaScript in Node.js können Sie mit einem einzigen herunterladbaren Paket, das die Bibliothek AWS Node.js, Codebeispiele und Dokumentation enthält, schnell loslegen.

Weitere Informationen finden Sie im AWS SDK für JavaScript in Node.js.

AWS SDK für PHP

Mit dem AWS SDK für PHP können Sie mit einem einzigen herunterladbaren Paket, das die AWS PHP-Bibliothek, Codebeispiele und Dokumentation enthält, schnell loslegen.

Weitere Informationen finden Sie unter AWS SDK for PHP. Weitere Informationen zum SDK, Beispielcode, Dokumentation, Tools und zusätzliche Ressourcen finden Sie im PHP on AWS Developer Center.

AWS SDK für Python (Boto)

Mit dem AWS SDK für Python (Boto) können Sie schnell mit einem einzigen herunterladbaren Paket loslegen, das die AWS Python-Bibliothek, Codebeispiele und Dokumentation enthält. Darüber hinaus können Sie Python-Anwendungen erstellen APIs , die die Komplexität der Codierung direkt für Webservice-Schnittstellen verringern.

Die all-in-one Bibliothek bietet entwicklerfreundliche Python-Funktionen APIs , die viele der untergeordneten Aufgaben im Zusammenhang mit der Programmierung für die AWS Cloud verbergen, darunter Authentifizierung, Wiederholungsversuche von Anfragen und Fehlerbehandlung. Das SDK enthält auch praktische Python-Beispiele, um mit den Bibliotheken Anwendungen zu erstellen.

Informationen zu Boto, Beispielcode, Dokumentation, Tools und zusätzliche Ressourcen finden Sie im Python on AWS Developer Center.

AWS SDK für Ruby

Mit der AWS Ruby-Bibliothek, den Codebeispielen und der Dokumentation können Sie zusätzlich Ruby-Anwendungen erstellen APIs , die das Programmieren direkt mit Webservice-Schnittstellen vereinfachen.

Die all-in-one Bibliothek ist entwicklerfreundlich APIs und verbirgt viele der untergeordneten Aufgaben im Zusammenhang mit der Programmierung für die AWS Cloud, einschließlich Authentifizierung, Wiederholungsversuche von Anfragen und Fehlerbehandlung. Das SDK enthält auch praktische Ruby-Beispiele, um mit den Bibliotheken Anwendungen zu erstellen.

Informationen über das SDK, Beispielcode, Dokumentation, Tools und zusätzliche Ressourcen finden Sie im Ruby on Developer Center. AWS

Archivierte Inhalte für das AWS Elastic Beanstalk Developer Guide

Zu Referenzzwecken enthält dieses Kapitel Konfigurationsinformationen zu einigen Legacy-Funktionen von Elastic Beanstalk.

Themen

- Benutzerdefinierte Elastic Beanstalk-Plattformen (nicht mehr verfügbar)

Benutzerdefinierte Elastic Beanstalk-Plattformen (nicht mehr verfügbar)

> **Note**
>
> Am 18. Juli 2022 hat Elastic Beanstalk den Status aller Plattformbranches, die auf Amazon Linux AMI (AL1) basieren, auf eingestellt gesetzt. Dies schließt auch benutzerdefinierte Plattformen ein. Elastic Beanstalk unterstützt keine benutzerdefinierte Plattformen. Weitere Informationen über Elastic Beanstalks Einstellung von Amazon Linux AMI finden Sie unter Häufig gestellte Fragen zur Einstellung von Plattformen.

Dieses Thema verbleibt in diesem Dokument als Referenz für alle Kunden, die die benutzerdefinierte Plattformfunktion von Elastic Beanstalk vor ihrer Einstellung verwendet haben. In der Vergangenheit unterstützten benutzerdefinierte Elastic Beanstalk-Plattformen die Erstellung eines AMI auf Basis von Amazon Linux AMI, RHEL 7, RHEL 6 oder Ubuntu 16.04. AMIs Diese Betriebssysteme werden von Elastic Beanstalk nicht mehr unterstützt. Um mehr über die nicht länger unterstützte Funktion der benutzerdefinierten Plattformen zu erfahren, erweitern Sie das folgende Thema.

Benutzerdefinierte Plattformen

Eine benutzerdefinierte Plattform ist mehr eine erweiterte Anpassung als ein benutzerdefiniertes Image. Eine benutzerdefinierte Plattform gestattet Ihnen, eine ganze Plattform von Grund auf neu zu entwickeln, das Betriebssystem anzupassen sowie zusätzliche Software und Skripts hinzuzufügen, die Elastic Beanstalk auf Plattform-Instances ausführt. Diese Flexibilität ermöglicht Ihnen den Aufbau einer Plattform für eine Anwendung, die eine Sprache oder andere Infrastruktur-

Software verwendet, für die Elastic Beanstalk keine verwaltete Plattform bietet. Vergleichen Sie dies mit benutzerdefinierten Abbildern, bei denen Sie ein Amazon Machine Image (AMI) für die Verwendung mit einer vorhandenen Elastic Beanstalk-Plattform ändern und Elastic Beanstalk weiterhin die Plattform-Skripts bereitstellt und den Software-Stack der Plattform steuert. Darüber hinaus verwenden Sie bei benutzerdefinierten Plattformen eine automatisierte, skriptgesteuerte Methode, Ihre Anpassung zu erstellen und zu verwalten, während Sie bei benutzerdefinierten Abbildern die Änderungen manuell über eine laufende Instance ausführen.

Um eine benutzerdefinierte Plattform zu erstellen, erzeugen Sie ein AMI aus einem der unterstützten Betriebssysteme – Ubuntu, RHEL oder Amazon Linux (die genauen Versionsnummern finden Sie im Eintrag `flavor` unter Platform.yaml-Dateiformat) – und fügen weitere Anpassungen hinzu. Sie erstellen Ihre eigene Elastic Beanstalk-Plattform mit Packer, einem Open-Source-Tool zur Erstellung von Maschinenimages für viele Plattformen, auch AMIs für die Verwendung mit Amazon Elastic Compute Cloud (Amazon). EC2 Eine Elastic Beanstalk-Plattform umfasst ein AMI, das zur Ausführung der anwendungsunterstützenden Software konfiguriert ist, sowie Metadaten, die benutzerdefinierte Konfigurationsoptionen und Standardeinstellungen der Konfigurationsoptionen enthalten können.

Elastic Beanstalk verwaltet Packer als separate integrierte Plattform und Sie brauchen sich nicht um die Konfiguration und Versionen von Packer zu kümmern.

Sie erstellen eine Plattform, indem Sie eine Packer-Vorlage sowie die von der Vorlage aufgerufenen Skripts und Dateien für die AMI-Erstellung in Elastic Beanstalk bereitstellen. Diese Komponenten werden mit einer Plattformdefinitionsdatei, in der die Vorlage und die Metadaten spezifiziert werden, in ein ZIP-Archiv verpackt, das als Plattformdefinitionsarchiv bezeichnet wird.

Wenn Sie eine benutzerdefinierte Plattform erstellen, starten Sie eine Umgebung mit einer einzelnen Instance ohne eine Elastic IP, die Packer ausführt. Packer startet dann eine weitere Instance, um ein Image zu erstellen. Sie können diese Umgebung für mehrere Plattformen und verschiedene Versionen jeder Plattform einsetzen.

> **Note**
>
> Benutzerdefinierte Plattformen sind regionsspezifisch. AWS Falls Sie Elastic Beanstalk in mehreren Regionen nutzen, müssen Sie die Plattformen in jeder Region separat erstellen. In bestimmten Situationen werden vom Packer gestartete Instances nicht gelöscht und müssen manuell beendet werden. Informationen zum manuellen Bereinigen dieser Instances finden Sie unter Bereinigen von Instances durch Packer.

Die Benutzer in Ihrem Konto können die benutzerdefinierten Plattformen verwenden, indem sie während der Umgebungserstellung einen Plattform-ARN angeben. Diese ARNs werden von dem eb platform create Befehl zurückgegeben, mit dem Sie die benutzerdefinierte Plattform erstellt haben.

Jedes Mal, wenn Sie eine benutzerdefinierte Plattform erstellen, generiert Elastic Beanstalk eine neue Plattformversion. Benutzer können den Namen einer Plattform angeben, damit nur die neueste Version der Plattform aufgerufen wird, oder eine Versionsnummer spezifizieren, um eine bestimmte Version zu erhalten.

Beispielsweise stellen Sie die benutzerdefinierte Plattform mit dem ARN **`MyCustomPlatformARN`** und der neuesten Version 3.0 mit folgendem EB CLI-Befehl bereit:

```
eb create -p MyCustomPlatformARN
```

Für die Bereitstellung von Version 2.1 sieht der EB CLI-Befehl wie folgt aus:

```
eb create -p MyCustomPlatformARN --version 2.1
```

Sie können eine benutzerdefinierte Plattformversion mit Tags markieren, wenn Sie sie erstellen, und Tags von vorhandenen benutzerdefinierten Plattformversionen bearbeiten. Details hierzu finden Sie unter Markieren von benutzerdefinierten Plattformversionen.

Erstellen einer benutzerdefinierten Plattform

Damit Sie eine benutzerdefinierte Plattform erstellen können, muss der Anwendungsstamm eine Plattformdefinitionsdatei enthalten, `platform.yaml`. Diese definiert den Builder-Typ, mit dem die benutzerdefinierte Plattform erstellt wird. Das Format dieser Datei ist in Platform.yaml-Dateiformat beschrieben. Sie können die benutzerdefinierte Plattform von Grund auf neu erstellen oder eines der benutzerdefinierten Plattformbeispiele als Ausgangspunkt verwenden.

Verwenden eines benutzerdefinierten Plattformbeispiels

Als Alternative zum Erstellen einer eigenen benutzerdefinierten Plattform können Sie eines der Plattformdefinitionsarchiv-Beispiele für den Bootstrap der benutzerdefinierten Plattform nutzen. Die einzigen Elemente, die Sie in den Beispielen konfigurieren müssen, bevor Sie sie verwenden können, sind ein Quell-AMI und eine Region.

> **Note**
>
> Verwenden Sie kein nicht abgeändertes Beispiel für eine benutzerdefinierte Plattform in der Produktion. Das Ziel der Beispiele besteht darin, einige verfügbare Funktionen für eine benutzerdefinierte Plattform zu veranschaulichen, sie sind jedoch nicht für den Einsatz in der Produktion gehärtet worden.

NodePlatform_Ubuntu.zip

Diese benutzerdefinierte Plattform basiert auf Ubuntu 16.04 und unterstützt Node.js 4.4.4. In den folgenden Beispielen wird diese benutzerdefinierte Plattform verwendet.

NodePlatform_RHEL.zip

Diese benutzerdefinierte Plattform basiert auf RHEL 7.2 und unterstützt Node.js 4.4.4.

NodePlatform_ AmazonLinux .zip

Diese benutzerdefinierte Plattform basiert auf Amazon Linux 2016.09.1 und unterstützt Node.js 4.4.4.

TomcatPlatform_Ubuntu.zip

Diese benutzerdefinierte Plattform basiert auf Ubuntu 16.04 und unterstützt Tomcat 7/Java 8.

CustomPlatform_ NodeSampleApp .zip

In diesem Node.js-Beispiel wird eine statische Webseite mithilfe von express und ejs angezeigt.

CustomPlatform_ .zip TomcatSampleApp

In diesem Tomcat-Beispiel wird eine statische Webseite bei der Bereitstellung angezeigt.

Laden Sie das Plattformdefinitionsarchiv-Beispiel herunter: `NodePlatform_Ubuntu.zip`. Diese Datei enthält eine Plattformdefinitionsdatei, eine Packer-Vorlage, Skripts, die während der Image-Erstellung von Packer ausgeführt werden, sowie Skripts und Konfigurationsdateien, die von Packer im Rahmen der Plattformerstellung auf die Builder-Instance kopiert werden.

Example NodePlatform_Ubuntu.zip

```
|-- builder                Contains files used by Packer to create the custom platform
```

```
|-- custom_platform.json    Packer template
|-- platform.yaml           Platform definition file
|-- ReadMe.txt              Briefly describes the sample
```

Die Plattformdefinitionsdatei, `platform.yaml`, teilt Elastic Beanstalk den Namen der Packer-Vorlage mit, `custom_platform.json`.

```
version: "1.0"

provisioner:
  type: packer
  template: custom_platform.json
  flavor: ubuntu1604
```

Die Packer-Vorlage teilt Packer mit, wie das AMIs für die Plattform erstellt wird, wobei ein Ubuntu-AMI als Basis für das Plattform-Image für HVM-Instance-Typen verwendet wird. Der Abschnitt `provisioners` weist Packer an, alle Dateien im Archiv-Ordner `builder` auf die Instance zu kopieren und das Skript `builder.sh` auf der Instance auszuführen. Nach Abschluss des Skripts erstellt Packer ein Image von der geänderten Instance.

Elastic Beanstalk erstellt drei Umgebungsvariablen, die zum Taggen AMIs in Packer verwendet werden können:

AWS_EB_PLATFORM_ARN

Der ARN der benutzerdefinierten Plattform.

AWS_EB_PLATTFORMNAME

Der Name der benutzerdefinierten Plattform.

AWS_EB_PLATTFORMVERSION

Die Version der benutzerdefinierten Plattform.

Die Beispieldatei `custom_platform.json` nutzt diese Variablen, um die folgenden in den Skripts verwendeten Werte zu definieren:

- `platform_name`, wird festgelegt von der Datei `platform.yaml`.
- `platform_version`, wird festgelegt von der Datei `platform.yaml`.

- `platform_arn`, wird festgelegt vom Build-Hauptskript `builder.sh`, das sich am Ende der Beispieldatei `custom_platform.json` befindet.

Die Datei `custom_platform.json` enthält zwei Eigenschaften, für die Sie Werte angeben müssen: `source_ami` und `region`. Einzelheiten zur Auswahl der richtigen AMI- und Regionswerte finden Sie unter Packer-Vorlage im eb-custom-platforms-samples GitHub Repository aktualisieren.

Example custom_platform.json

```
{
  "variables": {
    "platform_name": "{{env `AWS_EB_PLATFORM_NAME`}}",
    "platform_version": "{{env `AWS_EB_PLATFORM_VERSION`}}",
    "platform_arn": "{{env `AWS_EB_PLATFORM_ARN`}}"
  },
  "builders": [
    {
      ...
      "region": "",
      "source_ami": "",
      ...
    }
  ],
  "provisioners": [
    {...},
    {
      "type": "shell",
      "execute_command": "chmod +x {{ .Path }}; {{ .Vars }} sudo {{ .Path }}",
      "scripts": [
        "builder/builder.sh"
      ]
    }
  ]
}
```

Die Skripts und andere Dateien, die Sie in das Plattformdefinitionsarchiv einbinden, sind je nach den Änderungen, die Sie an der Instance vornehmen möchten, sehr unterschiedlich. Das Plattformbeispiel enthält die folgenden Skripts:

- `00-sync-apt.sh` – Auskommentiert: `apt -y update`. Wir haben den Befehl auskommentiert, weil er den Benutzer zu einer Eingabe auffordert, die die automatische Paketaktualisierung

unterbricht. Dies könnte ein Ubuntu-Problem sein. Als bewährte Methode empfehlen wir jedoch weiterhin, `apt -y update` auszuführen. Aus diesem Grund haben wir den Befehl zur Referenz im Beispiel-Skript beibehalten.

- `01-install-nginx.sh` – Installiert nginx.
- `02-setup-platform.sh` – Installiert `wget`, `tree` und `git`. Kopiert Hooks sowie Protokollierungskonfigurationen auf die Instance und erstellt die folgenden Verzeichnisse:
  - `/etc/SampleNodePlatform` – Gibt an, wohin die Container-Konfigurationsdatei während der Bereitstellung hochgeladen wird.
  - `/opt/elasticbeanstalk/deploy/appsource/` – Gibt an, wohin der Anwendungsquellcode während der Bereitstellung vom `00-unzip.sh`-Skript hochgeladen wird (weitere Informationen zu diesem Skript finden Sie im Abschnitt Plattform-Skripttools für Ihre Elastic Beanstalk Beanstalk-Umgebungen).
  - `/var/app/staging/` – Gibt an, wo der Anwendungsquellcode während der Bereitstellung verarbeitet wird.
  - `/var/app/current/` – Gibt an, wo der Anwendungsquellcode nach der Verarbeitung ausgeführt wird.
  - `/var/log/nginx/healthd/` – Gibt an, wohin die Protokolle vom Agent für erweiterte Integritätsberichte geschrieben werden.
  - `/var/nodejs` – Gibt an, wohin die Node.js-Dateien während der Bereitstellung hochgeladen werden.

Verwenden Sie die EB CLI, um Ihre erste benutzerdefinierte Plattform mit dem Plattformdefinitionsarchiv-Beispiel zu erstellen.

So erstellen Sie eine benutzerdefinierte Plattform

1. Installieren Sie die EB CLI.
2. Erstellen Sie ein Verzeichnis, in welches das benutzerdefinierte Plattformbeispiel extrahiert wird.

   ```
   ~$ mkdir ~/custom-platform
   ```

3. Extrahieren Sie `NodePlatform_Ubuntu.zip` in das Verzeichnis und wechseln Sie dann zu diesem Verzeichnis.

   ```
   ~$ cd ~/custom-platform
   ~/custom-platform$ unzip ~/NodePlatform_Ubuntu.zip
   ```

```
~/custom-platform$ cd NodePlatform_Ubuntu
```

4. Bearbeiten Sie die Datei `custom_platform.json` und geben Sie Werte für die Eigenschaften `region` und `source_ami` an. Weitere Informationen finden Sie unter Aktualisieren der Packer-Vorlage.
5. Führen Sie eb platform init aus und folgen Sie den Anweisungen, um ein Plattform-Repository zu initialisieren.

   Sie können eb platform zu ebp verkürzen.

   > **Note**
   >
   > Windows PowerShell verwendet ebp als Befehlsalias. Wenn Sie die EB-CLI in Windows ausführen PowerShell, verwenden Sie die Langform dieses Befehls:eb platform.

   ```
   ~/custom-platform$ eb platform init
   ```

   Mit diesem Befehl wird zudem das Verzeichnis `.elasticbeanstalk` im aktuellen Verzeichnis erstellt, dann wird die Konfigurationsdatei `config.yml` zum Verzeichnis hinzugefügt. Diese Datei wird von Elastic Beanstalk zum Erstellen der benutzerdefinierten Plattform benötigt, daher darf sie weder geändert noch gelöscht werden.

   Standardmäßig verwendet eb platform init den Namen des aktuellen Ordners als Bezeichnung für die benutzerdefinierte Plattform, in diesem Beispiel also `custom-platform`.
6. Führen Sie eb platform create aus, um die Packer-Umgebung zu starten und den ARN der benutzerdefinierten Plattform abzurufen. Diesen Wert benötigen Sie später zum Erstellen der Umgebung für die benutzerdefinierte Plattform.

   ```
   ~/custom-platform$ eb platform create
   ...
   ```

   Standardmäßig erstellt Elastic Beanstalk das Instance-Profil `aws-elasticbeanstalk-custom-platform-ec2-role` für benutzerdefinierte Plattformen. Wenn Sie stattdessen ein vorhandenes Instance-Profil nutzen möchten, fügen Sie die Option `-ip` *`INSTANCE_PROFILE`* zum Befehl eb platform create hinzu.

> **Note**
>
> Falls Sie das Instance-Standardprofil `aws-elasticbeanstalk-ec2-role` von Elastic Beanstalk nutzen, kann Packer die benutzerdefinierte Plattform nicht erstellen.

In der EB CLI wird die Ereignisausgabe der Packer-Umgebung angezeigt, bis die Erstellung abgeschlossen ist. Sie können die Ereignisanzeige verlassen, indem Sie Strg+C drücken.

7. Mit dem Befehl eb platform logs können Sie die Protokolle auf Fehler überprüfen.

```
~/custom-platform$ eb platform logs
...
```

8. Mit eb platform events können Sie den Prozess später prüfen.

```
~/custom-platform$ eb platform events
...
```

9. Überprüfen Sie den Status der Plattform mit eb platform status.

```
~/custom-platform$ eb platform status
...
```

Nach Abschluss des Vorgangs steht Ihnen eine Plattform zur Verfügung, auf der Sie eine Elastic Beanstalk-Umgebung starten können.

Sie können die benutzerdefinierte Plattform einsetzen, um eine neue Umgebung mit der Konsole zu erstellen. Siehe Der Assistent zum Erstellen einer neuen Umgebung.

So starten Sie eine Umgebung auf der benutzerdefinierten Plattform

1. Erstellen Sie ein Verzeichnis für die Anwendung.

```
~$ mkdir custom-platform-app
~$ cd ~/custom-platform-app
```

2. Initialisieren Sie ein Anwendungs-Repository.

```
~/custom-platform-app$ eb init
...
```

3. Laden Sie die Beispielanwendung (NodeSampleApp.zip) herunter.
4. Extrahieren Sie die Beispielanwendung.

```
~/custom-platform-app$ unzip ~/NodeSampleApp.zip
```

5. Ausführeneb create -p *CUSTOM-PLATFORM-ARN*, wobei der von einem eb platform create Befehl zurückgegebene ARN angezeigt *CUSTOM-PLATFORM-ARN* wird, um eine Umgebung zu starten, auf der Ihre benutzerdefinierte Plattform ausgeführt wird.

```
~/custom-platform-app$ eb create -p CUSTOM-PLATFORM-ARN
...
```

Inhalte des Plattformdefinitionsarchivs

Ein Plattformdefinitionsarchiv ist die Plattformentsprechung vom Quell-Bundle der Anwendung. Das Plattformdefinitionsarchiv ist eine ZIP-Datei und enthält eine Plattformdefinitionsdatei, eine Packer-Vorlage sowie die von der Packer-Vorlage verwendeten Skripts und Dateien für die Plattformerstellung.

> **Note**
>
> Wenn Sie die EB CLI zum Erstellen der benutzerdefinierten Plattform verwenden, generiert die EB CLI ein Plattformdefinitionsarchiv aus den Dateien und Ordnern im Plattform-Repository, daher muss das Archiv nicht manuell angelegt werden.

Die Datei Plattformdefinitionsdatei weist das YAML-Format auf, muss den Namen `platform.yaml` haben und im Plattformdefinitionsarchiv-Stamm sein. Unter Erstellen einer benutzerdefinierten Plattform finden Sie eine Liste der erforderlichen und optionalen Schlüssel, die in einer Plattformdefinitionsdatei unterstützt werden.

Die Packer-Vorlage muss nicht nach einer bestimmten Konvention benannt werden, jedoch muss dieser Dateiname mit der Provisioner-Vorlage, die in der Plattformdefinitionsdatei angegeben ist,

übereinstimmen. Anleitungen zum Erstellen von Packer-Vorlagen finden Sie in der offiziellen Packer-Dokumentation.

Bei den anderen Dateien im Plattformdefinitionsarchiv handelt es sich um Skripts und Dateien, die von der Vorlage vor der AMI-Erstellung zur Instance-Anpassung verwendet werden.

Benutzerdefinierte Plattform-Hooks

Elastic Beanstalk verwendet eine standardisierte Verzeichnisstruktur für Hooks auf benutzerdefinierten Plattformen. Dies sind Skripts, die bei Lebenszyklusereignissen und als Reaktion auf Verwaltungsvorgänge ausgeführt werden, z. B. wenn Instances in der Umgebung gestartet werden oder wenn ein Benutzer eine Bereitstellung initiiert oder die Funktion für den Neustart des Anwendungsservers nutzt.

Platzieren Sie Skripts, die Hooks auslösen sollen, in einem der Unterordner des Ordners `/opt/elasticbeanstalk/hooks/`.

> **Warning**
>
> Die Verwendung benutzerdefinierter Plattform-Hooks auf verwalteten Plattformen wird nicht unterstützt. Benutzerdefinierte Plattform-Hooks sind für benutzerdefinierte Plattformen bestimmt. Auf von Elastic Beanstalk verwalteten Plattformen funktionieren sie möglicherweise anders oder weisen Probleme auf und das Verhalten kann sich je nach Plattform unterscheiden. Auf Amazon Linux AMI-Plattformen (vor Amazon Linux 2) funktionieren sie möglicherweise immer noch auf nützliche Weise. Verwenden Sie sie jedoch mit Vorsicht. Benutzerdefinierte Plattform-Hooks sind eine Legacy-Funktion auf Amazon Linux AMI-Plattformen. Auf Amazon Linux 2-Plattformen werden benutzerdefinierte Plattform-Hooks im Ordner `/opt/elasticbeanstalk/hooks/` vollständig eingestellt. Elastic Beanstalk liest oder führt sie nicht aus. Amazon Linux 2-Plattformen unterstützen eine neue Art von Plattform-Hooks, die speziell zur Erweiterung von verwalteten Elastic Beanstalk-Plattformen entwickelt wurden. Sie können benutzerdefinierte Skripte und Programme direkt zu einem Hook-Verzeichnis in Ihrem Anwendungs-Quell-Bundle hinzufügen. Elastic Beanstalk führt sie während verschiedener Instance-Bereitstellungsphasen aus. Weitere Informationen erhalten Sie, indem Sie den Abschnitt Plattform-Hooks unter the section called “Erweitern von Linux-Plattformen” erweitern.

Hooks sind in den folgenden Ordnern organisiert:

- `appdeploy` – Skripts, die während der Anwendungsbereitstellung ausgeführt werden. Wenn neue Instances gestartet werden oder ein Client eine neue Versionsbereitstellung initiiert, führt Elastic Beanstalk eine Anwendungsbereitstellung aus.
- `configdeploy` – Skripts, die im Rahmen von clientseitigen Konfigurationsaktualisierungen, welche sich auf die Softwarekonfiguration der Instance auswirken (z. B. durch Festlegen von Umgebungseigenschaften oder Aktivieren der Protokollrotation an Amazon S3), ausgeführt werden.
- `restartappserver` – Skripts, die bei einem clientseitigen Neustart des Anwendungsservers ausgeführt werden.
- `preinit` – Skripts, die während des Instance-Bootstrapping ausgeführt werden.
- `postinit` – Skripts, die nach dem Instance-Bootstrapping ausgeführt werden.

Die Ordner `appdeploy`, `configdeploy` und `restartappserver` enthalten die Unterordner `pre`, `enact` und `post`. In den einzelnen Vorgangsphasen werden alle Skripts im Ordner `pre` in alphabetischer Reihenfolge ausgeführt, dann folgen diejenigen im Ordner `enact` und anschließend diejenigen im Ordner `post`.

Beim Starten einer Instance führt Elastic Beanstalk `preinit`, `appdeploy` und `postinit` in dieser Reihenfolge aus. Bei nachfolgenden Bereitstellungen auf ausgeführten Instances führt Elastic Beanstalk die `appdeploy`-Hooks aus. Die `configdeploy`-Hooks werden ausgeführt, wenn ein Benutzer die Instance-Softwarekonfigurationseinstellungen aktualisiert. Die `restartappserver`-Hooks werden nur ausgeführt, wenn ein Benutzer den Neustart des Anwendungsservers initiiert.

Falls die Skripts auf Fehler treffen, werden sie mit einem Status ungleich null beendet und schreiben den fehlgeschlagenen Vorgang in die Datei `stderr`. Die in die Datei `stderr` geschriebene Nachricht wird im Ereignis angezeigt, das im Falle eines fehlgeschlagenen Vorgangs ausgegeben wird. Zudem werden diese Informationen von Elastic Beanstalk in der Protokolldatei `/var/log/eb-activity.log` erfasst. Wenn der Vorgang nicht als fehlgeschlagen gelten soll, muss 0 (Null) zurückgegeben werden. Nachrichten, die in die Datei `stderr` oder `stdout` geschrieben werden, erscheinen in den Bereitstellungsprotokollen, werden aber nur bei einem fehlgeschlagenen Vorgang im Ereignis-Stream angegeben.

Bereinigen von Instances durch Packer

Unter bestimmten Umständen, z. B. wenn der Packer-Erstellungsprozess vorzeitig abgebrochen wird, werden von Packer gestartete Instances nicht bereinigt. Diese Instances sind nicht Teil der Elastic

Beanstalk Beanstalk-Umgebung und können nur mithilfe des EC2 Amazon-Service angezeigt und beendet werden.

So bereinigen Sie diese Instances manuell

1. Öffnen Sie die EC2 Amazon-Konsole.
2. Stellen Sie sicher, dass Sie sich in derselben AWS Region befinden, in der Sie die Instance mit Packer erstellt haben.
3. Wählen Sie unter Ressourcen die Option *N* Running Instances aus, *N* was die Anzahl der laufenden Instances angibt.
4. Klicken Sie in das Abfragetextfeld.
5. Wählen Sie das Tag Name aus.
6. Geben Sie packer ein.

   Die Abfrage sollte wie folgt aussehen: tag:Name: packer
7. Wählen Sie die Instances aus, die mit der Abfrage übereinstimmen.
8. Wenn der Wert für Instance State (Instance-Status) mit running (wird ausgeführt) angegeben wird, wählen Sie Actions (Aktionen), Instance State (Instance-Status), Stop (Anhalten) und dann Actions (Aktionen), Instance State (Instance-Status), Terminate (Beenden) aus.

Platform.yaml-Dateiformat

Die `platform.yaml`-Datei hat folgendes Format:

```
version: "version-number"

provisioner:
   type: provisioner-type
   template: provisioner-template
   flavor: provisioner-flavor

metadata:
   maintainer: metadata-maintainer
   description: metadata-description
   operating_system_name: metadata-operating_system_name
   operating_system_version: metadata-operating_system_version
   programming_language_name: metadata-programming_language_name
   programming_language_version: metadata-programming_language_version
   framework_name: metadata-framework_name
```

```
   framework_version: metadata-framework_version

option_definitions:
   - namespace: option-def-namespace
     option_name: option-def-option_name
     description: option-def-description
     default_value: option-def-default_value

option_settings:
   - namespace: "option-setting-namespace"
     option_name: "option-setting-option_name"
     value: "option-setting-value"
```

Ersetzen Sie die Platzhalter mit folgenden Werten:

version-number

Erforderlich Die Version der YAML-Definition. Der Wert muss **1.0** sein.

provisioner-type

Erforderlich Der Typ des Builder, der für die Erstellung der benutzerdefinierten Plattform verwendet wurde. Der Wert muss **packer** sein.

provisioner-template

Erforderlich Die JSON-Datei mit den Einstellungen für*provisioner-type*.

provisioner-flavor

Optional. Das Basis-Betriebssystem für die AMI. Eine der beiden folgenden Komponenten:

amazon (Standard)

Amazon Linux. Wenn nicht angegeben, wird die aktuelle Version von Amazon Linux verwendet, wenn die Plattform erstellt wird.

Amazon Linux 2 ist keine unterstützte Betriebssystemvariante.

ubuntu1604

Ubuntu 16.04 LTS

rhel7

RHEL 7

rhel6

RHEL 6

metadata-maintainer

Optional. Kontaktinformationen für den Eigentümer der Plattform (100 Zeichen).

metadata-description

Optional. Beschreibung der Plattform (2000 Zeichen).

metadata-operating_system_name

Optional. Name der Plattform im Betriebssystem (50 Zeichen). Dieser Wert ist verfügbar, wenn die Ausgabe für die ListPlatformVersionsAPI gefiltert wird.

metadata-operating_system_version

Optional. Version der Plattform im Betriebssystem (20 Zeichen).

metadata-programming_language_name

Optional. Programmiersprache, die von der Plattform unterstützt wird (50 Zeichen)

metadata-programming_language_version

Optional. Version der Sprache im Betriebssystem (20 Zeichen).

metadata-framework_name

Optional. Der Name des Web-Framework, das von der Plattform verwendet wird (50 Zeichen).

metadata-framework_version

Optional. Version des Web-Framework im Betriebssystem (20 Zeichen).

option-def-namespace

Optional. Ein Namespace unter `aws:elasticbeanstalk:container:custom` (100 Zeichen)

option-def-option_name

Optional. Der Name der Option (100 Zeichen). Sie können bis zu 50 benutzerdefinierte Konfigurationsoptionen festlegen, die die Plattform Benutzern bereitstellt.

option-def-description

Optional. Beschreibung der Option (1024 Zeichen).

option-def-default_value

Optional. Der Standardwert wird verwendet, wenn der Benutzer keinen angibt.

Das folgende Beispiel erstellt die Option **NPM_START**.

```
options_definitions:
 -  namespace: "aws:elasticbeanstalk:container:custom:application"
    option_name: "NPM_START"
    description: "Default application startup command"
    default_value: "node application.js"
```

option-setting-namespace

Optional. Namespace der Option.

option-setting-option_name

Optional. Name der Option. Sie können bis zu 50 Optionen von Elastic Beanstalk angeben.

option-setting-value

Optional. Der Wert wird verwendet, wenn der Benutzer keinen angibt.

Das folgende Beispiel erstellt die Option **TEST**.

```
option_settings:
 - namespace: "aws:elasticbeanstalk:application:environment"
   option_name: "TEST"
   value: "This is a test"
```

Markieren von benutzerdefinierten Plattformversionen

Sie können Tags auf Ihre AWS Elastic Beanstalk benutzerdefinierten Plattformversionen anwenden. Tags sind Schlüssel-Wert-Paare, die Ressourcen zugeordnet AWS sind. Weitere Informationen zum Elastic Beanstalk-Ressourcen-Tagging, zu Anwendungsfälle, Einschränkungen für Tag-Schlüssel und -Werte sowie zu unterstützten Ressourcentypen finden Sie unter Markieren von Elastic Beanstalk-Anwendungsressourcen.

Sie können Tags angeben, wenn Sie eine benutzerdefinierte Plattformversion erstellen. In einer vorhandenen benutzerdefinierten Plattformversion können Sie Tags hinzufügen oder entfernen

sowie die Werte von vorhandenen Tags aktualisieren. Sie können jeder benutzerdefinierten Plattformversion bis zu 50 Tags hinzufügen.

Hinzufügen von Tags beim Erstellen einer benutzerdefinierten Plattformversion

Wenn Sie mit der EB CLI Ihre benutzerdefinierte Plattform erstellen, verwenden Sie die `--tags`-Option mit dem Befehl eb platform create, um Tags hinzuzufügen.

```
~/workspace/my-app$ eb platform create --tags mytag1=value1,mytag2=value2
```

Fügen Sie mit dem AWS CLI oder anderen API-basierten Clients Tags hinzu, indem Sie den `--tags` Parameter im Befehl verwenden. create-platform-version

```
$ aws elasticbeanstalk create-platform-version \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
      --platform-name my-platform --platform-version 1.0.0 --platform-definition-bundle
 S3Bucket=amzn-s3-demo-bucket,S3Key=sample.zip
```

Verwalten von Tags einer vorhandenen benutzerdefinierten Plattformversion

In einer vorhandenen benutzerdefinierten Elastic Beanstalk-Plattformversion können Sie Tags hinzufügen, aktualisieren und löschen.

Wenn Sie Ihre benutzerdefinierte Plattformversion mit der EB CLI aktualisieren, verwenden Sie den Befehl eb tags zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags.

Beispiel: Der folgende Befehl listet die Tags in einer benutzerdefinierten Plattformversion auf.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-
account-id:platform/my-platform/1.0.0"
```

Mit dem folgenden Befehl wird das Tag `mytag1` aktualisiert und das Tag `mytag2` gelöscht.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-
platform/1.0.0"
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter `eb tags`.

Verwenden Sie bei den AWS CLI oder anderen API-basierten Clients den list-tags-for-resource Befehl, um die Tags einer benutzerdefinierten Plattformversion aufzulisten.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-platform/1.0.0"
```

Verwenden Sie den Befehl update-tags-for-resource zum Hinzufügen, Aktualisieren und Löschen von Tags in einer benutzerdefinierten Plattformversion.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-
platform/1.0.0"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter `--tags-to-add` des Befehls update-tags-for-resource an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

> **Note**
>
> Um einige der EB-CLI und AWS CLI Befehle mit einer benutzerdefinierten Plattformversion von Elastic Beanstalk zu verwenden, benötigen Sie den ARN der benutzerdefinierten Plattformversion. Verwenden Sie den folgenden Befehl, um den ARN abzurufen.
>
> ```
> $ aws elasticbeanstalk list-platform-versions
> ```
>
> Verwenden Sie die `--filters`-Option zum Filtern der Leistung Ihrer benutzerdefinierten Plattform.

Dokumentverlauf

In der folgenden Tabelle werden die wichtigen Änderungen am AWS Elastic Beanstalk Entwicklerhandbuch seit April 2024 beschrieben.

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinienaktualisierungen für `AWSElasticBeanstalkWorkerTier`	Aktualisierte Berechtigungen in AWS verwalteten Richtlinien für KI-gestützte Umgebungsanalysen.	11. März 2026
AWS verwaltete Richtlinienaktualisierungen für `AWSElasticBeanstalkWebTier`	Aktualisierte Berechtigungen in AWS verwalteten Richtlinien für KI-gestützte Umgebungsanalysen.	11. März 2026
AWS verwaltete Richtlinienaktualisierungen für `AWSElasticBeanstalkMulticontainerDocker`	Aktualisierte Berechtigungen in AWS verwalteten Richtlinien für KI-gestützte Umgebungsanalysen.	11. März 2026
Neues Thema: KI-gestützte Umgebungsanalyse	Elastic Beanstalk fügt KI-gestützte Analysen hinzu, um mithilfe von Amazon Bedrock Grundursachen zu identifizieren und Lösungen für Umweltprobleme bereitzustellen.	1. Dezember 2025
AWS verwaltete Richtlinienaktualisierungen für `AdministratorAccess-AWSElasticBeanstalk`	Aktualisierte Berechtigungen in AWS verwalteten Richtlinien.	12. November 2025

Neues Thema: Konfiguration von Elastic Beanstalk Load Balancern mit zwei Stacks	Elastic Beanstalk unterstützt Ihre Umgebung, um beide IPv4 IPv6 Protokolle zu bedienen, indem es die Dual-Stack-Konfigurationsoption für Application und Network Load Balancer bereitstellt.	9. September 2025
Überarbeitetes Thema: Was ist? AWS Elastic Beanstalk	Neues Architekturdiagramm und kurze Einführung zur besseren Übersicht.	5. Juni 2025
Überarbeitetes Thema: EB CLI einrichten	Kompakte und zielgerichtete Installationsanweisungen.	5. Juni 2025
Überarbeitetes Thema: PHP-Schnellstart	Ein Schritt zum Aktualisieren und erneuten Bereitstellen der Anwendung wurde hinzugefügt und einige Anweisungen wurden verbessert.	5. Juni 2025
Überarbeitetes Thema: Tutorial „Erste Schritte“	Das Tutorial „Erste Schritte“ wurde für mehr Übersichtlichkeit und schnelleres Onboarding optimiert.	5. Juni 2025
Inhalt neu organisiert	Die Inhaltsstruktur wurde neu organisiert, um wichtige Informationen hervorzuheben und die Auffindbarkeit zu verbessern.	5. Juni 2025
Neues Thema: Verwaltung von EC2-Sicherheitsgruppen	Elastic Beanstalk bietet Unterstützung für die Opt-Out-Umgebung aus der standardmäßigen EC2-Sicherheitsgruppe.	30. April 2025

Neues Thema: Migration von IIS-Anwendungen zu Elastic Beanstalk	Elastic Beanstalk fügt eine Funktion hinzu, zu der Sie Ihre Windows IIS-Anwen dungen migrieren können. AWS Elastic Beanstalk	18. April 2025
Neues Thema: Elastic Beanstalk mit AWS Secrets Manager und AWS Systems Manager Parameter Store verwenden	Elastic Beanstalk bietet Unterstützung für AWS Secrets Manager Referenzg eheimnisse und AWS Systems Manager Parameter Store-Parameter mit Umgebungs variablen.	31. März 2025
AWS verwaltete Richtlini enaktualisierungen für`AWSElasticBeanstal kManagedUpdatesCus tomerRolePolicy` .	Aktualisierte Berechtigungen in AWS verwalteten Richtlini en.	27. Februar 2025
Startkonfigurationen sind zugunsten von Startvorlagen veraltet	Dieses Thema wurde überarbeitet, um zu erläutern , wie Elastic Beanstalk Beanstalk-Umgebungen davon betroffen sind, dass Amazon EC2 Auto Scaling Startkonf igurationen zugunsten von Startvorlagen nicht mehr unterstützt. Wir haben auch andere Inhalte im Zusammenh ang mit Startkonfigurationen und Startvorlagen überarbei tet.	23. Januar 2025

Neues Thema: Verwenden von EC2 Fast Launch mit Windows-Plattformzweigen	Die Versionen der Elastic Beanstalk Windows-Plattform enthalten Basis-AMIs mit aktiviertem EC2 Fast Launch.	22. Januar 2025
Neues Thema: Strategie für die Zuweisung von Spot-Instances	Elastic Beanstalk bietet Unterstützung für die Konfiguration der Spot Allocation Strategy während der Umgebungserstellung.	15. Januar 2025
`AdministratorAccess-AWSElasticBeanstalk` AWS Verwaltete Richtlinie	Die Berechtigungen in der AWS verwalteten Richtlinie wurden aktualisiert.	11. Dezember 2024
Empfehlungen für Graviton Arm64-First-Wave-Umgebungen im Kapitel Archivierte Inhalte	Die EB-Empfehlungen für Graviton-Arm64-Umgebungen der ersten Welle wurden in das Kapitel Archivierter Inhalt verschoben	5. Oktober 2024
Startvorlagen	Neues Thema: Vorlagen starten. In diesem Thema wird erklärt, wie Amazon EC2 Auto Scaling Startkonfigurationen schrittweise zugunsten von Startvorlagen auslaufen lässt.	1. Oktober 2024
QuickStartfor Python	Neues Thema: QuickStart für Python	24. September 2024
Erforderliche Amazon S3 S3-Bucket-Berechtigungen für restriktive VPC-Endpunktrichtlinien	Neues Thema: Erforderliche Amazon S3 S3-Bucket-Berechtigungen für restriktive VPC-Endpunktrichtlinien	18. September 2024

QuickStart für Java auf Tomcat	Neues Thema: QuickStart für Java-JSP-Anwendungen, die auf Tomcat laufen	12. September 2024
QuickStart für Java	Neues Thema: QuickStart für Java	12. September 2024
Verwenden des Amazon EC2 Systems Manager für private Docker-Repositorys	Thema wieder hergestellt: Amazon EC2 Systems Manager für private Docker-Repositorys verwenden. Ersetzt das Thema Verwenden der privaten Repositorys AWS Secrets Manager für Docker, bis die Elastic Beanstalk Beanstalk/SSM-Integration unterstützt wird.	8. September 2024
EB CLI 2.6 (eingestellt) zum Kapitel Archivierter Inhalt	EB CLI 2.6 (eingestellt) wurde in das Kapitel Archivierter Inhalt verschoben	15. August 2024
EB API (eingestellt) in das Kapitel Archivierter Inhalt	Die EB-API (eingestellt) wurde in das Kapitel Archivierter Inhalt verschoben	15. August 2024
Bereitstellung von Elastic Beanstalk Beanstalk-Anwendungen aus Docker-Containern	Aktualisiert und neu organisiert: Bereitstellung von Elastic Beanstalk Beanstalk-Anwendungen aus Docker-Containern	15. August 2024
Archivierter Inhalt	Neues Kapitel: Archivierter Inhalt	15. August 2024
QuickStart für Windows ASP.NET	Neues Thema: QuickStart für Windows ASP.NET	5. Juli 2024

QuickStart für.NET Core unter Windows	Neues Thema: QuickStart für.NET Core unter Windows	28. Juni 2024
QuickStart für Docker	Neues Thema: QuickStart für Docker	19. Juni 2024
Verhinderung des umgebungsübergreifenden Amazon S3 S3-Bucket-Zugriffs	Neues Thema: Verhinderung des umgebungsübergreifenden Zugriffs auf Amazon S3 S3-Buckets	12. Juni 2024
QuickStart für.NET Core unter Linux	Neues Thema: QuickStart für.NET Core unter Windows	28. Mai 2024
QuickStart für PHP	Neues Thema: QuickStart für PHP	10. Mai 2024
QuickStart für Node.js	Neues Thema: QuickStart für Node.js	5. Mai 2024
QuickStart für Go	Neues Thema: QuickStart für Go	5. Mai 2024
Zeitplan für die Veröffentlichung der Elastic Beanstalk-Plattform	Neues Thema hinzugefügt, das einen Zeitplan für enthält. Kommende Veröffentlichungen in den Plattformbranchen Zeitplan für die Ausmusterung von Plattformvarianten Und Bisherige eingestellte Plattformzweige zu diesem Thema verschoben.	1. Mai 2024
`AWSElasticBeanstalkRoleCore` AWS Verwaltete Richtlinie	Die Berechtigungen in der AWS verwalteten Richtlinie wurden aktualisiert.	30. April 2024

AWSElasticBeanstalkManagedUpdatesServiceRolePolicy AWS Verwaltete Richtlinie	Die Berechtigungen in der AWS verwalteten Richtlinie wurden aktualisiert.	30. April 2024
AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy AWS Verwaltete Richtlinie	Die Berechtigungen in der AWS verwalteten Richtlinie wurden aktualisiert.	30. April 2024
AWSElasticBeanstalkMaintenance AWS Verwaltete Richtlinie	Die Berechtigungen in der AWS verwalteten Richtlinie wurden aktualisiert.	30. April 2024
AWSElasticBeanstalkInternalMaintenanceRolePolicy AWS Verwaltete Richtlinie	Die Berechtigungen in der AWS verwalteten Richtlinie wurden aktualisiert.	30. April 2024

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.