



Benutzer-Leitfaden

Amazon ElastiCache



API-Version 2015-02-02

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

Amazon ElastiCache: Benutzer-Leitfaden

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

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. 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 ElastiCache?	1
Serverless-Caching	1
Knotenbasierte Cluster	2
Zugehörige Services	2
Funktionsweise	4
Cache- und Caching-Engines	4
Auswählen aus den Bereitstellungsoptionen	10
ElastiCache Ressourcen	17
AWS Regionen und Verfügbarkeitszonen	19
Anwendungsfälle	21
Erste Schritte mit ElastiCache	33
Einrichten ElastiCache	33
Melden Sie sich an für ein AWS-Konto	33
Erstellen eines Benutzers mit Administratorzugriff	34
Erteilen programmgesteuerten Zugriffs	35
Berechtigungen einrichten	38
Richten Sie ein EC2	39
Gewähren des Netzwerkzugriffs	39
Richten Sie den Befehlszeilenzugriff ein	40
Erstellen Sie einen serverlosen Valkey-Cache	41
Lesen und Schreiben von Daten	43
Bereinigen	44
Nächste Schritte	46
Erstellen Sie einen serverlosen Redis OSS-Cache	46
Lesen und Schreiben von Daten	47
Bereinigen	49
Nächste Schritte	50
Erstellen Sie einen serverlosen Memcached-Cache	50
Lesen und Schreiben von Daten	52
Bereinigen	57
Nächste Schritte	58
Tutorials: Erste Schritte mit Python und ElastiCache	58
Python und ElastiCache	59
Tutorial: Lambda für den Zugriff ElastiCache in einer VPC konfigurieren	77

Schritt 1: Einen serverlosen Cache erstellen ElastiCache .	78
Schritt 2: Erstellen Sie eine Lambda-Funktion für ElastiCache	81
Schritt 3: Testen Sie die Lambda-Funktion mit ElastiCache	84
Schritt 4: Aufräumen (optional)	85
Einen knotenbasierten Cluster erstellen und verwalten ElastiCache	87
Komponenten und Funktionen	87
Knoten	88
ElastiCache Scherben	89
ElastiCache Cluster	89
ElastiCache Replikation	92
ElastiCache Endpunkte	94
Parametergruppen	95
ElastiCache Sicherheit	95
Subnetzgruppen	96
ElastiCache Backups	96
Ereignisse	97
ElastiCache Terminologie	98
Tutorial: So erstellen Sie einen knotenbasierten Cluster ElastiCache	100
Einen knotenbasierten ElastiCache Cluster für Valkey erstellen	100
Einen knotenbasierten ElastiCache Cluster für Redis OSS erstellen	122
Löschen eines Clusters	144
Weitere Tutorials und Videos	146
Videos	147
Knoten verwalten in ElastiCache	152
ElastiCache Knotenstatus anzeigen	153
Valkey- oder Redis-OSS-Knoten und -Shards	159
Mit Knoten verbinden	161
Unterstützte Knotentypen	166
Knoten neu starten	182
Knoten ersetzen (Valkey und Redis OSS)	187
Knoten ersetzen (Memcached)	195
Reservierte Knoten	197
Migration von Knoten der vorherigen Generation	213
Verwaltung von Clustern in ElastiCache	217
Auswahl eines Netzwerktyps in ElastiCache	219
Automatische Erkennung (Memcached)	224

Datenzuweisung ElastiCache	268
Einen Cluster vorbereiten in ElastiCache	275
Einen Cluster für Valkey oder Redis OSS erstellen	286
Einen Cluster für Memcached erstellen	296
Details eines ElastiCache Clusters anzeigen	300
Einen ElastiCache Cluster ändern	314
Hinzufügen von Knoten zu einem ElastiCache Cluster	320
Knoten aus einem ElastiCache Cluster entfernen	331
Abbrechen ausstehender Operationen zum Hinzufügen oder Löschen von Knoten in ElastiCache	340
Löschen eines Clusters in ElastiCache	341
Zugreifen auf Ihren ElastiCache Cluster oder Ihre Replikationsgruppe	344
Verbindungsendpunkte finden in ElastiCache	353
Scherben rein ElastiCache	369
Vergleich knotenbasierter Valkey-, Memcached- und Redis OSS-Cluster	375
Online-Migration für Valkey oder Redis OSS	380
Übersicht	381
Schritte zur Migration	382
Quelle und Ziel für die Migration vorbereiten	382
Testen der Datenmigration	384
Migration starten	385
Verifizieren des Fortschritts der Datenmigration	386
Datenmigration abschließen	387
Durchführen einer Online-Datenmigration mithilfe der Konsole	388
Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache	389
Überlegungen zur Availability Zone bei Memcached	390
Lokalisieren Ihrer Knoten	393
Unterstützte Regionen und Endpunkte	393
Verwenden von lokalen Zonen mit ElastiCache	398
Outposts verwenden mit ElastiCache	400
Arbeiten mit ElastiCache	405
Snapshot und Wiederherstellung	405
Beschränkungen	406
Auswirkungen von Backups knotenbasierter Cluster auf die Leistung	407
Planen automatischer Backups	409
Erstellen manueller Backups	411

Erstellen einer endgültigen Sicherung	418
Beschreiben von Sicherungen	421
Kopieren eines Backups	423
Exportieren einer Sicherung	425
Wiederherstellen aus einem Backup	433
Löschen einer Sicherung	434
Markieren von Sicherungen	436
Tutorial: Seeding eines knotenbasierten Clusters mit einem Backup	437
Motorversionen und Aufrüstung in ElastiCache	447
Upgrade von Engine-Versionen	447
Erweiterter Support	452
Versionsverwaltung für ElastiCache	456
Valkey 8.2	463
Valkey 8.1	463
Valkey 8.0	464
ElastiCache Version 7.2.6 für Valkey	464
Unterstützte Redis OSS-Versionen	465
Zeitplan für das Ende der Lebensdauer der Redis OSS-Versionen	480
Unterstützte Memcached-Versionen	481
Wichtige Unterschiede zwischen dem Verhalten und der Kompatibilität der Engine- Versionen mit Valkey	486
Wesentliche Unterschiede zwischen dem Verhalten und der Kompatibilität der Engine- Versionen mit Redis OSS	486
Überlegungen zum Upgrade bei der Arbeit mit knotenbasierten Clustern	491
Bewährte Methoden und Caching-Strategien	493
Allgemeine Best Practices	494
Bewährte Methoden für die Verwendung von Read Replicas	495
Unterstützte und eingeschränkte Valkey-, Memcached- und Redis OSS-Befehle	499
Konfiguration und Grenzwerte für Valkey und Redis OSS	538
IPv6 Kundenbeispiele für Valkey, Memcached und Redis OSS	542
Bewährte Methoden für Kunden (Valkey und Redis OSS)	543
Bewährte Methoden für Kunden (Memcached)	567
TLS-fähige ElastiCache Dual-Stack-Cluster	571
Verwaltung von reserviertem Speicher für Valkey und Redis OSS	575
Bewährte Methoden bei der Arbeit mit nodenbasierten Valkey- und Redis OSS-Clustern	582
Caching-Strategien für Memcached	588

Verwaltung Ihres knotenbasierten Clusters in ElastiCache	594
Auto Scaling Valkey- und Redis OSS-Cluster	594
Ändern des Clustermodus	643
AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher	647
Hohe Verfügbarkeit mit Replikationsgruppen	677
Verwaltung der ElastiCache Cluster-Wartung	768
Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen	770
Automatisches Verbinden einer EC2 Instanz und eines ElastiCache Caches	891
Automatische EC2 Konnektivität	892
Anzeigen verbundener Rechenressourcen	899
Skalierung ElastiCache	900
ElastiCache Serverlose Skalierung	900
Festlegen von Skalierungslimits zur Kostenkontrolle	900
Vorskalierung mit Serverless ElastiCache	900
Skalierungsgrenzen mithilfe der Konsole festlegen und AWS CLI	902
Skalierung knotenbasierter Cluster	903
Erste Schritte mit Bloom-Filtern	993
Bloom-Datentyp	994
Größenbeschränkung für Blüten	995
ACLs	995
Metriken im Zusammenhang mit Bloom	996
Bloom-Filter-Befehle	996
Erste Schritte mit Watch in Serverless	997
Limit ansehen	999
Unterstützte Befehle im Zusammenhang mit Watch	999
Erste Schritte mit Vector Search	1000
Überblick über die Vektorsuche	1000
Funktionen und Grenzen der Vektorsuche	1005
Auswahl der geeigneten Konfiguration	1008
Befehle für die Vektorsuche	1009
Erste Schritte mit JSON für Valkey und Redis OSS	1022
Überblick über den JSON-Datentyp	1023
JSON-Befehle	1035
Verschlagworten Sie Ihre Ressourcen ElastiCache	1079
Überwachung von Kosten mit Tags	1091
Verwaltung von Stichwörtern mithilfe der AWS CLI	1093

Verwaltung von Tags mithilfe der ElastiCache API	1098
Amazon ElastiCache Well-Architected Lens	1101
Säule „Betriebliche Exzellenz“	1102
Säule der Sicherheit	1111
Säule der Zuverlässigkeit	1118
Säule der Leistungseffizienz	1125
Säule der Kostenoptimierung	1136
Problembehebung in ElastiCache	1143
Verbindungsprobleme	1144
Valkey- oder Redis OSS-Client-Fehler	1145
Fehlerbehebung bei hoher Latenz in Serverless ElastiCache	1145
Behebung von Drosselungsproblemen in Serverless ElastiCache	1148
Anhaltende Verbindungsprobleme	1149
Verwandte Themen	1171
Sicherheit	1172
Datenschutz	1173
Datensicherheit bei Amazon ElastiCache	1174
Richtlinie für den Datenverkehr zwischen Netzwerken	1257
Amazon VPCs und ElastiCache Sicherheit	1257
ElastiCache API- und Schnittstellen-VPC-Endpunkte (AWS PrivateLink)	1284
Subnetze und Subnetzgruppen	1289
Identitäts- und Zugriffsverwaltung	1298
Zielgruppe	1298
Authentifizierung mit Identitäten	1299
Verwalten des Zugriffs mit Richtlinien	1300
So ElastiCache arbeitet Amazon mit IAM	1302
Beispiele für identitätsbasierte Richtlinien	1308
Fehlerbehebung	1311
Zugriffskontrolle	1313
Übersicht über die Verwaltung von Zugriffsberechtigungen	1314
Compliance-Validierung	1367
Weitere Informationen	1368
Ausfallsicherheit	1368
Minimieren von Ausfällen	1369
Sicherheit der Infrastruktur	1374
Service-Updates	1374

Verwaltung von Service-Updates	1375
Sicherheitslücken wurden behoben	1380
Protokollierung und Überwachung	1383
Metriken und Ereignisse für Valkey und Redis OSS serverless	1383
Serverless-Metriken	1384
Serverless-Ereignisse	1394
Metriken und Ereignisse für knotenbasierte Valkey- und Redis OSS-Cluster	1411
Knotenbasierte Cluster-Metriken	1411
Knotenbasierte Cluster-Ereignisse (Valkey und Redis OSS)	1411
Metriken und Ereignisse für Memcached	1420
Metriken für serverlose Memcached	1420
Serverlose Memcached-Ereignisse	1424
Protokollierung Amazon ElastiCache Amazon-API-Aufrufen mit AWS CloudTrail	1437
ElastiCache Amazon-Informationen in CloudTrail	1437
ElastiCache Amazon-Protokolldateieinträge verstehen	1438
Amazon-SNS-Ereignisüberwachung	1442
Verwaltung von ElastiCache Amazon SNS SNS-Benachrichtigungen	1442
ElastiCache Ereignisse anzeigen	1448
Ereignisbenachrichtigungen und Amazon SNS	1452
Protokollzustellung	1461
Inhalt eines langsamen Protokolleintrags	1462
Inhalte eines Engine-Protokolleintrags	1462
Berechtigungen zum Konfigurieren der Protokollierung	1462
Spezifikationen des Protokolltyps und des Protokollformats	1463
ElastiCache Ziele protokollieren	1464
Angaben der Protokollzustellung mithilfe der Konsole	1467
Angaben der Protokollzustellung mit dem AWS CLI	1468
Überwachung der Nutzung	1474
Metriken auf Host-Ebene	1474
Metriken für Valkey und Redis OSS	1478
Metriken für Memcached	1499
Welche Metriken sollte ich überwachen?	1505
Auswählen von Metrikstatistiken und -zeiträumen	1510
Überwachung von CloudWatch Cluster- und Knotenmetriken	1510
Kontingente	1514
Referenz	1517

Die ElastiCache API verwenden	1517
Verwenden der Abfrage-API	1517
Verfügbare Bibliotheken	1521
Fehlerbehebung bei Anwendungen	1521
Richten Sie das AWS CLI für ein ElastiCache	1523
Voraussetzungen	1523
Aufrufen der Befehlszeilen-Tools	1525
Einrichten der Tools	1525
Bereitstellung von Anmeldeinformationen für die Tools	1526
Umgebungsvariablen	1527
Fehlermeldungen	1528
Benachrichtigungen	1530
Allgemeine ElastiCache Benachrichtigungen	1530
ElastiCache für Memcached-Benachrichtigungen	1531
ElastiCache für Redis OSS-spezifische Benachrichtigungen	1531
ElastiCache Geschichte der Dokumentation	1532
AWS Glossar	1575
.....	mdlxxvi

Was ist Amazon ElastiCache?

Willkommen im ElastiCache Amazon-Benutzerhandbuch. Amazon ElastiCache ist ein Webservice, der es einfach macht, einen verteilten In-Memory-Datenspeicher oder eine Cache-Umgebung in der Cloud einzurichten, zu verwalten und zu skalieren. Es handelt sich um eine leistungsstarke, skalierbare und kostengünstige Caching-Lösung. Gleichzeitig trägt es dazu bei, die mit der Bereitstellung und Verwaltung einer verteilten Cache-Umgebung verbundene Komplexität zu beseitigen.

Sie können Amazon ElastiCache in zwei Formaten betreiben. Sie können mit einem serverlosen Cache beginnen oder einen knotenbasierten Cluster erstellen.

Note

Amazon ElastiCache arbeitet mit den OSS-Engines Valkey, Memcached und Redis. Wenn Sie sich nicht sicher sind, welche Engine Sie verwenden möchten, beachten Sie [Vergleich knotenbasierter Valkey-, Memcached- und Redis OSS-Cluster](#) in diesem Handbuch.

Serverless-Caching

ElastiCache bietet serverloses Caching, was das Hinzufügen und Betreiben eines Caches für Ihre Anwendung vereinfacht. ElastiCache Serverless ermöglicht es Ihnen, in weniger als einer Minute einen hochverfügbaren Cache zu erstellen, und macht die Bereitstellung von Instanzen oder die Konfiguration von Knoten oder Clustern überflüssig. Entwickler können einen serverlosen Cache erstellen, indem sie den Cache-Namen über die ElastiCache Konsole, das SDK oder die CLI angeben.

ElastiCache Serverless macht auch die Planung und Verwaltung der Caching-Kapazität überflüssig. ElastiCache überwacht ständig den Arbeitsspeicher, die Rechenleistung und die Netzwerkbandbreite des Caches, die von Ihrer Anwendung verwendet werden, und skaliert, um den Anforderungen Ihrer Anwendung gerecht zu werden. ElastiCache bietet Entwicklern eine einfache Endpunkterfahrung, indem die zugrunde liegende Cache-Infrastruktur und das Cluster-Design abstrahiert werden. ElastiCache verwaltet die Hardwarebereitstellung, die Überwachung, den Austausch von Knoten und das Patchen von Software automatisch und transparent, sodass Sie sich auf die Anwendungsentwicklung konzentrieren können, anstatt den Cache zu betreiben.

ElastiCache Serverless ist mit Valkey 7.2, Memcached 1.6.21 und höher sowie Redis OSS 7.1 und höher kompatibel.

Einen knotenbasierten Cluster erstellen

Wenn Sie eine genaue Kontrolle über Ihren ElastiCache Cluster benötigen, können Sie wählen, ob Sie einen knotenbasierten Valkey-, Memcached- oder Redis-OSS-Cluster erstellen möchten. ElastiCache ermöglicht es Ihnen, einen knotenbasierten Cluster zu erstellen, indem Sie den Knotentyp, die Anzahl der Knoten und die Platzierung der Knoten in den Availability Zones für Ihren Cluster auswählen. AWS Da es ElastiCache sich um einen vollständig verwalteten Dienst handelt, verwaltet er automatisch die Hardwarebereitstellung, die Überwachung, den Austausch von Knoten und das Software-Patching für Ihren Cluster.

Die Erstellung eines knotenbasierten Clusters bietet mehr Flexibilität und Kontrolle über Ihre Cluster. Sie können beispielsweise je nach Ihren Anforderungen wählen, ob Sie einen Cluster mit Single-AZ- oder Multi-AZ-Verfügbarkeit betreiben möchten. Sie können auch wählen, ob Sie Valkey, Memcached oder Redis OSS im Clustermodus ausführen möchten, der horizontale Skalierung ermöglicht, oder ohne Clustermodus, um nur vertikal zu skalieren. Bei der Erstellung eines knotenbasierten Clusters sind Sie dafür verantwortlich, den Typ und die Anzahl der Knoten richtig auszuwählen, um sicherzustellen, dass Ihr Cache über die für Ihre Anwendung erforderliche Kapazität verfügt. Sie können auch wählen, wann neue Software-Patches auf Ihren Valkey- oder Redis OSS-Cluster angewendet werden sollen.

Bei der Erstellung eines knotenbasierten Clusters können Sie aus mehreren unterstützten Versionen von Valkey, Memcached und Redis OSS wählen. Weitere Informationen zu unterstützten Engine-Versionen finden Sie unter [Motorversionen und Aufrüstung in ElastiCache](#)

Zugehörige Services

[Speicher-DB](#)

Bei der Entscheidung, ob Sie MemoryDB ElastiCache oder MemoryDB verwenden möchten, sollten Sie die folgenden Vergleiche berücksichtigen:

- ElastiCache ist ein Dienst, der häufig verwendet wird, um Daten aus anderen Datenbanken und Datenspeichern mithilfe von Valkey, Memcached oder Redis OSS zwischenspeichern. Sie sollten beim Zwischenspeichern von Workloads in Betracht ziehen ElastiCache , bei denen Sie

den Datenzugriff mit Ihrer vorhandenen Primärdatenbank oder Ihrem vorhandenen Datenspeicher beschleunigen möchten (Lese- und Schreibleistung im Mikrosekundenbereich). Sie sollten auch Anwendungsfälle in Betracht ziehen ElastiCache , in denen Sie Valkey- oder Redis-OSS-Datenstrukturen verwenden und auf Daten zugreifen APIs möchten, die in einer Primärdatenbank oder einem Datenspeicher gespeichert sind.

- ElastiCache kann Ihnen auch dabei helfen, Datenbankkosten zu sparen, indem häufig aufgerufene Daten in einem Cache gespeichert werden. Wenn Ihre Anwendung hohe Anforderungen an den Lesedurchsatz stellt, können Sie eine hohe Skalierbarkeit, hohe Leistung und geringere Datenspeicherkosten erzielen, indem Sie Ihre zugrunde liegende Datenbank verwenden ElastiCache, anstatt sie zu skalieren.
- MemoryDB ist eine robuste In-Memory-Datenbank für Workloads, die eine ultraschnelle Primärdatenbank erfordern. Sie ist mit Valkey und Redis OSS kompatibel. Sie sollten die Verwendung von MemoryDB in Betracht ziehen, wenn Ihr Workload eine dauerhafte Datenbank erfordert, die eine ultraschnelle Leistung bietet (Leselatenz im Mikrosekundenbereich und Schreiblatenz im einstelligen Millisekundenbereich). MemoryDB eignet sich möglicherweise auch gut für Ihren Anwendungsfall, wenn Sie eine Anwendung mit Valkey- oder Redis OSS-Datenstrukturen und APIs mit einer primären, dauerhaften Datenbank erstellen möchten. Schließlich sollten Sie erwägen, MemoryDB zu verwenden, um Ihre Anwendungsarchitektur zu vereinfachen und die Kosten zu senken, indem Sie die Verwendung einer Datenbank durch einen Cache ersetzen, um Dauerhaftigkeit und Leistung zu gewährleisten.

[Amazon Relational Database Service](#)

ElastiCache kann Ihnen helfen, Datenbankkosten zu sparen, indem Daten, auf die häufig zugegriffen wird, in einem Cache gespeichert werden. Wenn Ihre Anwendung hohe Anforderungen an den Lesedurchsatz stellt, können Sie eine hohe Skalierbarkeit, hohe Leistung und geringere Datenspeicherkosten erzielen, indem Sie Ihre zugrunde liegende Datenbank verwenden ElastiCache, anstatt sie zu skalieren.

Weitere Hintergrundinformationen zum entsprechenden Service Amazon Relational Database Service finden Sie unter [Amazon](#) RDS

ElastiCache kann Ihnen helfen, Datenbankkosten zu sparen, indem Daten, auf die häufig zugegriffen wird, in einem Cache gespeichert werden. Wenn Ihre Anwendung hohe Anforderungen an den Lesedurchsatz stellt, können Sie eine hohe Skalierbarkeit, hohe Leistung und geringere Datenspeicherkosten erzielen, indem Sie Ihre zugrunde liegende Datenbank verwenden ElastiCache, anstatt sie zu skalieren.

Wie ElastiCache funktioniert

Hier finden Sie einen Überblick über die wichtigsten Komponenten eines ElastiCache Deployments.

Cache- und Caching-Engines

Ein Cache ist ein In-Memory-Datenspeicher, den Sie zum Speichern zwischengespeicherter Daten verwenden können. In der Regel speichert Ihre Anwendung häufig aufgerufene Daten in einem Cache, um die Antwortzeiten zu optimieren. ElastiCache bietet zwei Bereitstellungsoptionen: serverlose Caches und knotenbasierte Cluster. Siehe [Auswählen aus den Bereitstellungsoptionen](#).

Note

Amazon ElastiCache arbeitet mit den OSS-Engines Valkey, Memcached und Redis. Wenn Sie sich nicht sicher sind, welche Engine Sie verwenden möchten, beachten Sie [Vergleich knotenbasierter Valkey-, Memcached- und Redis OSS-Cluster](#) in diesem Handbuch.

Themen

- [Wie ElastiCache funktioniert](#)
- [Preisdimensionen](#)
- [ElastiCache Backups](#)

Wie ElastiCache funktioniert

ElastiCache Serverless

ElastiCache Mit Serverless können Sie einen Cache erstellen, ohne sich Gedanken über Kapazitätsplanung, Hardwaremanagement oder Clusterdesign machen zu müssen. Sie geben einfach einen Namen für Ihren Cache an und Sie erhalten einen einzigen Endpunkt, den Sie in Ihrem Valkey-, Memcached- und Redis OSS-Client konfigurieren können, um mit dem Zugriff auf Ihren Cache zu beginnen.

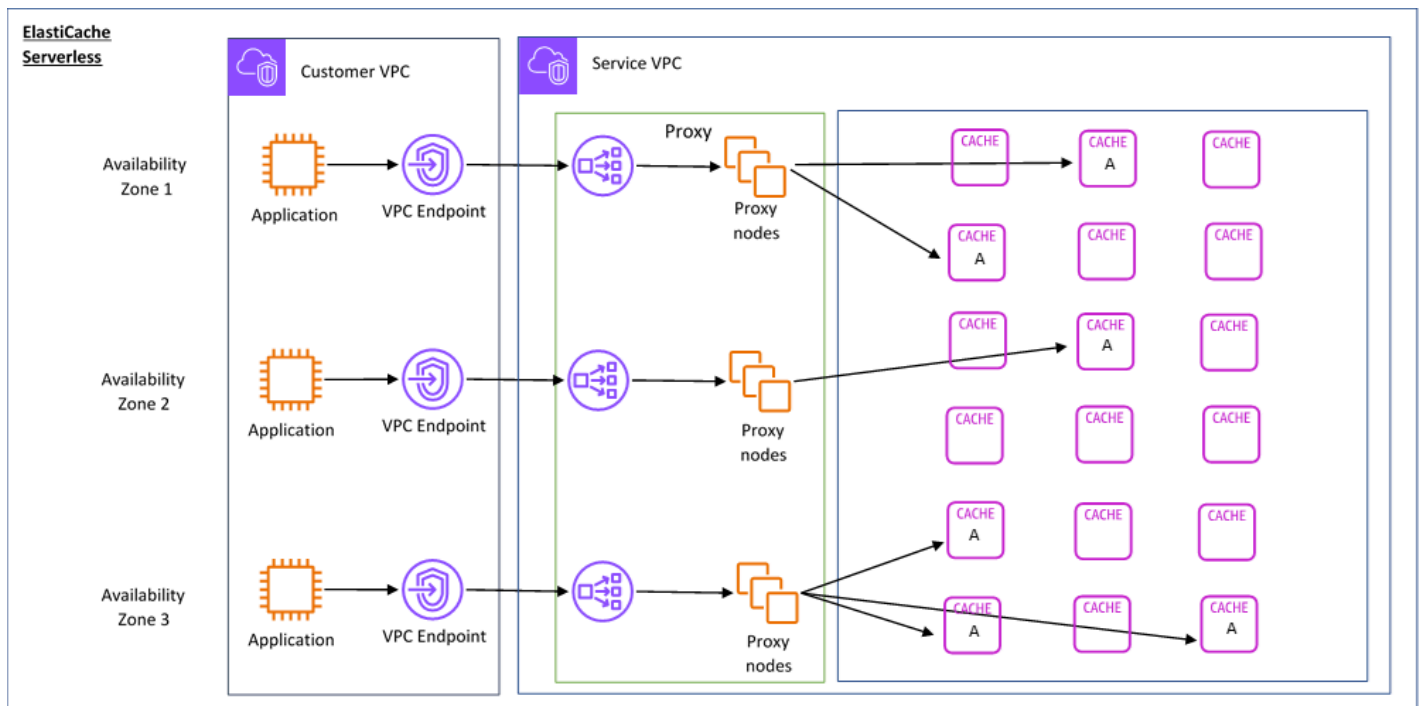
Note

- ElastiCache Serverless führt Valkey, Memcached oder Redis OSS im Clustermodus aus und ist nur mit Clients kompatibel, die TLS unterstützen.

Wichtigste Vorteile

- **Keine Kapazitätsplanung:** ElastiCache Serverless macht die Kapazitätsplanung überflüssig. ElastiCache Serverless überwacht kontinuierlich die Speicher-, Rechen- und Netzwerkbandbreitennutzung Ihres Caches und skaliert sowohl vertikal als auch horizontal. Dadurch kann ein Cache-Knoten an Größe zunehmen und gleichzeitig ein Aufskalierungsvorgang initiiert werden, um sicherzustellen, dass der Cache jederzeit skaliert werden kann, um Ihre Anwendungsanforderungen zu erfüllen.
- **Pay-per-use:** Bei ElastiCache Serverless zahlen Sie für die im Cache gespeicherten Daten und die Rechenleistung, die von Ihrem Workload genutzt wird. Siehe [Preisdimensionen](#).
- **Hochverfügbarkeit:** ElastiCache Serverless repliziert Ihre Daten automatisch über mehrere Availability Zones (AZ) hinweg, um eine hohe Verfügbarkeit zu gewährleisten. Es überwacht automatisch die zugrunde liegenden Cache-Knoten und ersetzt sie bei Ausfällen. Es bietet ein SLA mit einer Verfügbarkeit von 99,99 % für jeden Cache.
- **Automatische Software-Upgrades:** ElastiCache Serverless aktualisiert Ihren Cache automatisch auf die neueste Minor- und Patch-Softwareversion, ohne dass die Verfügbarkeit Ihrer Anwendung beeinträchtigt wird. Sobald eine neue Hauptversion verfügbar ist, ElastiCache erhalten Sie eine Benachrichtigung.
- **Sicherheit:** Serverless verschlüsselt Daten während der Übertragung und im Ruhezustand. Sie können einen vom Service verwalteten Schlüssel oder einen eigenen kundenverwalteten Schlüssel verwenden, um Daten im Ruhezustand zu verschlüsseln.

Das folgende Diagramm zeigt, wie ElastiCache Serverless funktioniert.



Wenn Sie einen neuen serverlosen Cache erstellen, ElastiCache wird ein Virtual Private Cloud (VPC) -Endpunkt in den Subnetzen Ihrer Wahl in Ihrer VPC erstellt. Ihre Anwendung kann über diese VPC-Endpunkte eine Verbindung mit dem Cache herstellen.

Mit ElastiCache Serverless erhalten Sie einen einzigen DNS-Endpunkt, mit dem Ihre Anwendung eine Verbindung herstellt. Wenn Sie eine neue Verbindung zum Endpunkt anfordern, verarbeitet ElastiCache Serverless alle Cache-Verbindungen über eine Proxyschicht. Die Proxy-Schicht trägt zu einer weniger komplexen Client-Konfiguration bei, da der Client die Cluster-Topologie nicht erneut ermitteln muss, wenn Änderungen am zugrunde liegenden Cluster vorgenommen werden. Die Proxy-Schicht besteht aus einer Reihe von Proxy-Knoten, die Verbindungen mithilfe eines Network Load Balancer verarbeiten.

Wenn Ihre Anwendung eine neue Cache-Verbindung erstellt, wird die Anfrage vom Network Load Balancer an einen Proxy-Knoten gesendet. Führt Ihre Anwendung Cache-Befehle aus, dann führt der Proxy-Knoten, der mit Ihrer Anwendung verbunden ist, die Anfragen auf einem Cache-Knoten in Ihrem Cache aus. Die Proxyschicht abstrahiert die Clustertopologie und die Knoten von Ihrem Client. Dies ermöglicht ElastiCache einen intelligenten Lastenausgleich, die Skalierung und das Hinzufügen neuer Cache-Knoten, den Austausch von Cache-Knoten bei deren Ausfall und die Aktualisierung der Software auf den Cache-Knoten, ohne dass die Verfügbarkeit Ihrer Anwendung beeinträchtigt wird oder Verbindungen zurückgesetzt werden müssen.

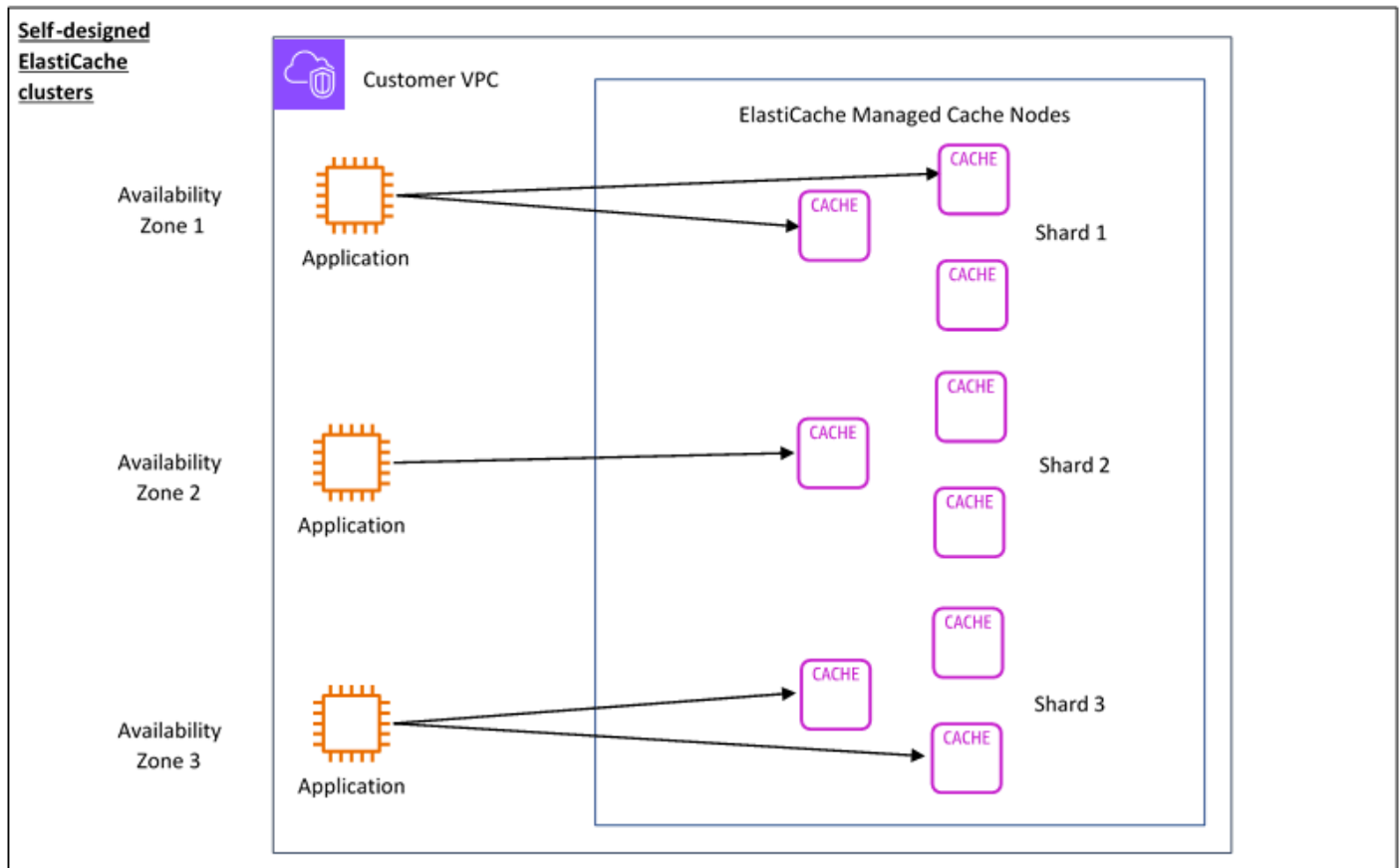
Knotenbasierte Cluster

Sie können einen knotenbasierten ElastiCache Cluster erstellen, indem Sie eine Cache-Knotenfamilie, Größe und Anzahl der Knoten für Ihren Cluster auswählen. Durch die Erstellung eines knotenbasierten Clusters haben Sie eine genauere Kontrolle und können die Anzahl der Shards in Ihrem Cache und die Anzahl der Knoten (primär und repliziert) in jedem Shard auswählen. Sie können wählen, ob Sie Valkey oder Redis OSS im Clustermodus betreiben möchten, indem Sie einen Cluster mit mehreren Shards erstellen, oder im Nicht-Cluster-Modus mit einem einzigen Shard.

Wichtigste Vorteile

- Erstellen Sie einen knotenbasierten Cluster: Mit können Sie einen knotenbasierten Cluster erstellen und auswählen ElastiCache, wo Sie Ihre Cache-Knoten platzieren möchten. Wenn Sie beispielsweise über eine Anwendung verfügen, bei der es um einen Kompromiss zwischen hoher Verfügbarkeit und niedriger Latenz geht, können Sie sich dafür entscheiden, Ihre Cache-Knoten in einer einzigen AZ bereitzustellen. Alternativ können Sie einen knotenbasierten Cluster mit Knoten über mehrere Knoten erstellen, um eine hohe Verfügbarkeit zu erreichen. AZs
- Präzise Steuerung: Wenn Sie einen knotenbasierten Cluster erstellen, haben Sie mehr Kontrolle über die Feinabstimmung der Einstellungen in Ihrem Cache. Beispielsweise können Sie [Valkey- und Redis OSS-Parameter](#) oder verwenden, um die Cache-Engine [Memcached-spezifische Parameter](#) zu konfigurieren.
- Vertikales und horizontales Skalieren: Sie können Ihren Cluster wahlweise manuell skalieren, indem Sie die Größe des Cache-Knotens bei Bedarf erhöhen oder verringern. Sie können auch horizontal skalieren, indem Sie neue Shards oder weitere Replikate zu Ihren Shards hinzufügen. Sie können die Auto-Scaling-Funktion auch verwenden, um die Skalierung auf der Grundlage eines Zeitplans oder eine Skalierung auf der Grundlage von Messwerten wie der CPU- und Speicherauslastung im Cache zu konfigurieren.

Das folgende Diagramm zeigt, wie knotenbasierte Cluster funktionieren ElastiCache .



Preisdimensionen

Sie können die Bereitstellung ElastiCache in zwei Bereitstellungsoptionen durchführen. Bei der Bereitstellung von ElastiCache Serverless zahlen Sie für die Nutzung der in GB-Stunden gespeicherten Daten und für die ElastiCache Datenverarbeitung in Processing Units (ECPU). Bei der Erstellung eines knotenbasierten Clusters zahlen Sie pro Stunde für die Nutzung des Cache-Knotens. Weitere Informationen zu Preisen finden Sie [hier](#).

Datenspeicher

Sie zahlen für in ElastiCache Serverless gespeicherte Daten, die in Gigabyte-Stunden (GB-Stunden) abgerechnet werden. ElastiCache Serverless überwacht kontinuierlich die in Ihrem Cache gespeicherten Daten, wobei mehrere Samplings pro Minute durchgeführt werden, und berechnet einen Stundendurchschnitt, um die Datenspeichernutzung des Caches in GB-Stunden zu ermitteln. Jeder ElastiCache serverlose Cache wird für mindestens 1 GB an gespeicherten Daten gemessen.

ElastiCache Verarbeitungseinheiten (ECPUs)

Sie zahlen für die Anfragen, die Ihre Anwendung auf ElastiCache Serverless ausführt, in ElastiCache Processing Units (ECPUs), einer Einheit, die sowohl vCPU-Zeit als auch übertragene Daten beinhaltet.

- Einfache Lese- und Schreibvorgänge erfordern 1 ECPU für jedes Kilobyte (KB) übertragener Daten. Beispielsweise verbraucht ein GET-Befehl, der bis zu 1 KB Daten überträgt, 1 ECPU. Eine SET-Anfrage, die 3,2 KB an Daten überträgt, verbraucht 3,2 KB. ECPUs
- Bei Valkey und Redis OSS werden Befehle, die mehr vCPU-Zeit verbrauchen und mehr Daten übertragen, auf der ECPUs Grundlage der höheren der beiden Dimensionen verbraucht. Wenn Ihre Anwendung beispielsweise den Befehl HMGET verwendet, die dreifache vCPU-Zeit wie ein einfacher SET/GET Befehl verbraucht und 3,2 KB an Daten überträgt, verbraucht sie 3,2 ECPU. Wenn sie jedoch nur 2 KB an Daten überträgt, verbraucht sie 3 KB. ECPUs
- Mit Valkey und Redis OSS verbrauchen Befehle, die zusätzliche vCPU-Zeit benötigen, proportional mehr. ECPUs Wenn Ihre Anwendung beispielsweise den Befehl Valkey oder Redis OSS [HMGET verwendet und als einfacher SET/GET Befehl](#) das Dreifache der vCPU-Zeit verbraucht, verbraucht sie 3. ECPUs
- Mit Memcached verbrauchen Befehle, die auf mehrere Elemente angewendet werden, proportional mehr. ECPUs Wenn Ihre Anwendung beispielsweise ein Multiget für 3 Elemente ausführt, verbraucht sie 3 Objekte. ECPUs
- Bei Memcached werden Befehle, die auf mehr Elemente angewendet und mehr Daten übertragen, auf der ECPUs Grundlage der höheren der beiden Dimensionen verbraucht. Wenn Ihre Anwendung beispielsweise den Befehl GET verwendet, 3 Elemente abrufen und 3,2 KB Daten überträgt, verbraucht sie 3,2 ECPU. Wenn nur 2 KB an Daten übertragen werden, werden alternativ 3 KB verbraucht. ECPUs

ElastiCache Serverless gibt eine neue Metrik mit dem Namen `awsElastiCacheProcessingUnits`, die Ihnen hilft, zu verstehen, wie viel Ihr ECPUs Workload verbraucht.

Knotenstunden

Sie können einen knotenbasierten Cluster erstellen, indem Sie die EC2 Knotenfamilie, Größe, Anzahl der Knoten und die Platzierung in den Availability Zones auswählen. Bei der Erstellung eines knotenbasierten Clusters zahlen Sie pro Stunde für jeden Cache-Knoten.

ElastiCache Backups

Ein Backup ist eine point-in-time Kopie eines serverlosen Caches oder eines knotenbasierten Valkey- oder Redis OSS-Clusters. ElastiCache ermöglicht es Ihnen, jederzeit eine Sicherungskopie Ihrer Daten zu erstellen oder automatische Backups einzurichten. Backups können zur Wiederherstellung eines bestehenden Caches oder zum Seeding eines neuen Caches verwendet werden. Backups umfassen alle Daten in einem Cache sowie zusätzlich einige Metadaten. Weitere Informationen finden Sie unter [Snapshot und Wiederherstellung](#).

Auswählen aus den Bereitstellungsoptionen

Amazon ElastiCache bietet zwei Bereitstellungsoptionen:

- Serverless-Caching
- Knotenbasierte Cluster

Eine Liste der unterstützten Befehle für beide finden Sie unter. [Unterstützte und eingeschränkte Valkey-, Memcached- und Redis OSS-Befehle](#)

Serverless-Caching

Amazon ElastiCache Serverless vereinfacht die Cache-Erstellung und lässt sich sofort skalieren, um die anspruchsvollsten Anwendungen der Kunden zu unterstützen. Mit ElastiCache Serverless können Sie in weniger als einer Minute einen hochverfügbaren und skalierbaren Cache erstellen, sodass Sie keine Cluster-Kapazität bereitstellen, planen und verwalten müssen. ElastiCache Serverless speichert Daten automatisch redundant in drei Availability Zones und bietet ein Service Level Agreement (SLA) mit einer Verfügbarkeit von 99,99%. Backups von nodenbasierten Valkey- oder Redis OSS-Clustern können in einer serverlosen Konfiguration wiederhergestellt werden.

Knotenbasierte Cluster

Wenn Sie eine detaillierte Kontrolle über Ihren Valkey-, Memcached- oder Redis OSS-Cluster benötigen, können Sie einen knotenbasierten Cluster mit erstellen. ElastiCache Sie wählen den Knotentyp, die Anzahl der Knoten und die Platzierung der Knoten in den Availability Zones für Ihren Cluster. AWS Da es ElastiCache sich um einen vollständig verwalteten Service handelt, hilft er bei der Verwaltung der Hardwarebereitstellung, der Überwachung, des Austauschs von Knoten und des Software-Patches für Ihren Cluster. Knotenbasierte Cluster können so konzipiert werden, dass sie eine Verfügbarkeit von bis zu 99,99% bieten. Backups aus serverlosen Valkey- oder Redis OSS-Caches können in einem knotenbasierten Cluster wiederhergestellt werden.

Auswählen aus den Bereitstellungsoptionen

Wählen Sie Serverless-Caching in folgenden Fällen:

- Sie erstellen einen Cache für Workloads, die entweder neu oder schwer vorhersehbar sind.
- Sie haben unvorhersehbaren Anwendungsverkehr.
- Sie möchten den einfachsten Einstieg in die Arbeit mit einem Cache.

Erstellen Sie Ihren eigenen knotenbasierten Cluster, wenn:

- Sie verwenden bereits ElastiCache Serverless und möchten eine genauere Kontrolle über den Knotentyp, auf dem Valkey, Memcached oder Redis OSS ausgeführt wird, die Anzahl der Knoten und die Platzierung dieser Knoten.
- Sie erwarten, dass Ihr Anwendungsdatenverkehr relativ vorhersehbar ist, und Sie möchten eine genaue Kontrolle über Leistung, Verfügbarkeit und Kosten.
- Sie können Ihre Kapazitätsanforderungen vorhersagen, um Kosten zu kontrollieren.

Vergleich von serverlosem Caching und knotenbasierten Clustern

Feature	Serverless-Caching	Knotenbasierte Cluster
Cache-Einrichtung	Erstellen Sie in weniger als einer Minute einen Cache mit nur einem Namen	Bietet eine detaillierte Kontrolle über das Cluster-Design. Der Benutzer kann den Knotentyp, die Anzahl der Knoten und die Platzierung in den Availability Zones wählen AWS
Unterstützte Version ElastiCache	Valkey 7.2 und höher, Redis OSS Version 7.1 und höher, Memcached 1.6.21 und höher	Valkey 7.2 und höher, Redis OSS Version 4.0 und höher, Memcached 1.4 und höher
Cluster-Modus (Valkey und Redis OSS)	Betreibt Motoren nur <code>incluster mode enabled</code> . Clients müssen die Verbindung <code>cluster mode enabled</code>	Kann so konfiguriert werden, dass der Clustermodus aktiviert oder der Clustermodus deaktiviert ist.

Feature	Serverless-Caching	Knotenbasierte Cluster
	zu ElastiCache Serverless unterstützen.	
Skalierung	Skaliert Engines automatisch sowohl vertikal als auch horizontal ohne Kapazitätsmanagement.	Ermöglicht die Kontrolle über die Skalierung und erfordert gleichzeitig eine Überwachung, um sicherzustellen, dass die aktuelle Kapazität dem Bedarf angemessen entspricht. Für Valkey und Redis OSS können Sie wählen, ob Sie vertikal skalieren möchten, indem Sie die Größe des Cache-Knotens bei Bedarf erhöhen oder verringern. Sie können auch horizontal skalieren, indem Sie neue Shards oder weitere Replikate zu Ihren Shards hinzufügen. Diese Funktion ist für Memcached nicht verfügbar. Mit der Auto-Scaling-Funktion können Sie die Skalierung auch auf der Grundlage eines Zeitplans oder anhand von Messwerten wie der CPU- und Speicherauslastung im Cache konfigurieren.

Feature	Serverless-Caching	Knotenbasierte Cluster
Client-Verbindung	Clients stellen eine Verbindung zu einem einzelnen Endpunkt her. Dadurch kann die zugrunde liegende Cache-Knotentopologie (Skalierung, Austausch und Upgrades) geändert werden, ohne dass die Verbindung zum Client unterbrochen wird.	Clients stellen eine Verbindung zu jedem einzelnen Cache-Knoten her. Wenn ein Knoten ersetzt wird, erkennt der Client die Clustertopologie erneut und stellt die Verbindungen wieder her.
Konfigurierbarkeit	Keine detaillierte Konfiguration verfügbar. Kunden können grundlegende Einstellungen konfigurieren, darunter Subnetze, die auf den Cache zugreifen können, ob automatische Backups aktiviert oder deaktiviert werden, und maximale Cache-Nutzungsbeschränkungen.	Knotenbasierte Cluster bieten detaillierte Konfigurationsoptionen. Kunden können Parametergruppen für eine detaillierte Steuerung verwenden. Eine Tabelle dieser Parameterwerte nach Knotentyp finden Sie unter Motorspezifische Parameter .
Multi-AZ	Daten werden asynchron über mehrere Availability Zones repliziert, um eine höhere Verfügbarkeit und eine verbesserte Leselatenz zu erreichen.	Bietet eine Option zum Erstellen des Clusters in einer einzelnen Availability Zone oder in mehreren Availability Zones (AZs). Bietet bei Verwendung von Valkey oder Redis OSS Multi-AZ-Cluster mit Daten, die asynchron über mehrere Availability Zones repliziert werden, um eine höhere Verfügbarkeit und eine verbesserte Leselatenz zu erzielen.

Feature	Serverless-Caching	Knotenbasierte Cluster
Verschlüsselung im Ruhezustand	Immer aktiviert. Kunden können einen Von AWS verwalteter Schlüssel oder einen vom Kunden verwalteten Schlüssel in verwenden AWS KMS.	Option zum Aktivieren oder Deaktivieren der Verschlüsselung im Ruhezustand. Wenn diese Option aktiviert ist, können Kunden einen Von AWS verwalteter Schlüssel oder einen vom Kunden verwalteten Schlüssel in verwenden AWS KMS.
Verschlüsselung bei der Übertragung (TLS)	Immer aktiviert. Clients müssen TLS-Konnektivität unterstützen.	Option zum Aktivieren oder Deaktivieren.
Sicherungen	Unterstützt automatische und manuelle Backups von Caches ohne Leistungs einbußen. Valkey- und Redis OSS-Backups sind untereinander kompatibel und können in einem ElastiCache serverlosen Cache oder einem knotenbasierten Cluster wiederhergestellt werden.	Unterstützt automatische und manuelle Backups für Valkey und Redis OSS. Bei Clustern kann es je nach verfügbarem reservierten Speicher zu Leistungseinbußen kommen. Weitere Informationen finden Sie unter Verwaltung von reserviertem Speicher für Valkey und Redis OSS. Valkey- und Redis OSS-Backups sind untereinander kompatibel und können in einem ElastiCache serverlosen Cache oder einem knotenbasierten Cluster wiederhergestellt werden.

Feature	Serverless-Caching	Knotenbasierte Cluster
Überwachen	Support Metriken auf Cache-Ebene, einschließlich Cache-Trefferquote, Cache-Fehlrate, Datengröße und ECPUs verbrauchter Daten. ElastiCache Serverless sendet Ereignisse und verwendet dabei den EventBridge Zeitpunkt, an dem signifikante Ereignisse in Ihrem Cache auftreten. Sie können wählen, ob Sie ElastiCache Ereignisse mithilfe von Amazon EventBridge überwachen, aufnehmen, transformieren und darauf reagieren möchten. Weitere Informationen finden Sie unter Serverless-Cache-Ereignisse.	Knotenbasierte ElastiCache Cluster geben Metriken auf jeder Knotenebene aus, darunter sowohl Metriken auf Host-Ebene als auch Cache-Metriken. Knotenbasierte Cluster senden SNS-Benachrichtigungen für wichtige Ereignisse aus. Siehe Metriken für Memcached und Metriken für Valkey und Redis OSS.
Verfügbarkeit	Service Level Agreement (SLA) mit einer Verfügbarkeit von 99,99%	Knotenbasierte Cluster können so konfiguriert werden, dass sie je nach Konfiguration ein Service Level Agreement (SLA) mit einer Verfügbarkeit von bis zu 99,99% erreichen.

Feature	Serverless-Caching	Knotenbasierte Cluster
Software-Upgrades und Patches	Führt ein automatisches Upgrade der Cache-Software auf die neueste Minor- und Patch-Version durch, ohne dass sich dies auf die Anwendung auswirkt. Kunden erhalten eine Benachrichtigung über Hauptversions-Upgrades, und Kunden können jederzeit auf die neueste Hauptversion aktualisieren.	Knotenbasierte Cluster bieten kundenseitig aktivierten Self-Service für kleinere und patchende Versions-Upgrades sowie für Hauptversions-Upgrades. Verwaltete Updates werden automatisch während der vom Kunden definierten Wartungsfenster installiert. Kunden können auch wählen, ob sie bei Bedarf ein Upgrade auf eine Nebenversion oder eine Patch-Version durchführen möchten.
Globaler Datenspeicher	Nicht unterstützt	Unterstützt Global Data Store, der eine regionsübergreifende Replikation mit Schreibvorgängen in einer Region und Lesevorgängen in mehreren Regionen ermöglicht

Feature	Serverless-Caching	Knotenbasierte Cluster
Datenklassifizierung	Nicht unterstützt	Bei Clustern, die mithilfe von Knoten der R6GD-Familie erstellt werden, werden die Daten zwischen Arbeitsspeicher und lokalem SSD-Speicher (Solid State Drives) aufgeteilt. Data Tiering bietet eine preisgünstige Option für Valkey- und Redis-OSS-Workloads, da neben der Speicherung von Daten im Arbeitsspeicher auch kostengünstigere Solid-State-Laufwerke (SSDs) in jedem Clusterknoten verwendet werden.
Preismodell	Pay-per-use, basierend auf Daten, die in GB-Stunden gespeichert sind, und Anfragen in Processing Units (ECPU). ElastiCache Weitere Informationen zu Preisen finden Sie hier .	Pay-per-hour, basierend auf der Nutzung des Cache-Knotens. Weitere Informationen zu Preisen finden Sie hier .

Verwandte Themen:

- [Einen knotenbasierten Cluster erstellen und verwalten ElastiCache](#)

ElastiCache Amazon-Ressourcen für Erstnutzer

Wir empfehlen Erstbenutzern, mit der Lektüre der folgenden Abschnitte zu beginnen und bei Bedarf darauf zurückzugreifen.

- Leistungsmerkmale und Preise — Die [Produktdetailseite](#) bietet einen allgemeinen Produktüberblick über ElastiCache die wichtigsten Dienstleistungen und die Preisgestaltung.
- ElastiCache Videos — [ElastiCache Videos](#) In diesem Bereich finden Sie Videos, die Ihnen Amazon vorstellen ElastiCache. Die Videos behandeln allgemeine Anwendungsfälle ElastiCache und zeigen, wie Sie ElastiCache damit die Latenz reduzieren und den Durchsatz Ihrer Anwendungen verbessern können.
- Erste Schritte — Dieser [Erste Schritte mit Amazon ElastiCache](#) Abschnitt enthält Informationen zum Erstellen eines Caches. Außerdem erfahren Sie, wie Sie den Zugriff auf den Cache autorisieren, eine Verbindung zu einem Cache-Knoten herstellen und den Cache löschen.
- Skalierbare Leistung — Das ElastiCache Whitepaper [Performance at Scale with Amazon befasst sich mit](#) Caching-Strategien, die dazu beitragen, dass Ihre Anwendung im großen Maßstab eine gute Leistung erbringt.

Wenn Sie die vorhergehenden Abschnitte abgeschlossen haben, lesen die folgenden:

- [Auswahl der Knotengröße](#)

Knoten müssen so groß dimensioniert sein, dass sie alle Daten speichern können, die in den Cache aufgenommen werden sollen. Dabei möchten Sie auch nicht für mehr Cache bezahlen, als Sie benötigen. In diesem Thema finden Sie Informationen zur Auswahl der optimalen Knotengröße.

- [ElastiCache Best Practices und Caching-Strategien](#)

Identifizieren und Beheben von Problemen, die sich auf die Effizienz Ihres Clusters auswirken können.

Wenn Sie das AWS Command Line Interface(AWS CLI) verwenden möchten, können Sie die folgenden Dokumente verwenden, um Ihnen den Einstieg zu erleichtern:

- [AWS Command Line Interface Dokumentation](#)

Dieser Abschnitt enthält Informationen zum Herunterladen von AWS CLI, zum AWS CLI Betrieb auf Ihrem System und zum Eingeben Ihrer AWS Anmeldeinformationen.

- [AWS CLI Dokumentation für ElastiCache](#)

Dieses separate Dokument behandelt alle ElastiCache Befehle AWS CLI für, einschließlich Syntax und Beispielen.

Sie können Anwendungsprogramme schreiben, um die ElastiCache API mit einer Vielzahl gängiger Programmiersprachen zu verwenden. Nachfolgend finden Sie einige Ressourcen:

- [Tools für Amazon Web Services](#)

Amazon Web Services bietet eine Reihe von Software Development Kits (SDKs) mit Unterstützung für ElastiCache. Sie können für die ElastiCache Verwendung von Java, .NET, PHP, Ruby und anderen Sprachen programmieren. Diese SDKs können Ihre Anwendungsentwicklung erheblich vereinfachen, indem sie Ihre Anfragen formatieren ElastiCache, Antworten analysieren und Wiederholungslogik und Fehlerbehandlung bereitstellen.

- [Die ElastiCache API verwenden](#)

Wenn Sie das nicht verwenden möchten AWS SDKs, können Sie ElastiCache direkt über die Abfrage-API mit ihnen interagieren. In diesem Abschnitt finden Sie auch Tipps zur Fehlerbehebung sowie Informationen zur Erstellung und Authentifizierung von Anfragen und der Behandlung von Antworten.

- [ElastiCache Amazon-API-Referenz](#)

Dieses separate Dokument behandelt alle ElastiCache API-Operationen, einschließlich Syntax und Beispielen.

AWS Regionen und Verfügbarkeitszonen

Amazon Cloud Computing-Ressourcen sind in hochverfügbaren Rechenzentren in verschiedenen Regionen der Welt untergebracht (zum Beispiel in Nordamerika, Europa oder Asien). Jeder Rechenzentrumsstandort wird als AWS Region bezeichnet.

Jede AWS Region enthält mehrere unterschiedliche Standorte, die als Availability Zones oder bezeichnet AZs werden. Jede Availability Zone ist so aufgebaut, dass sie von Fehlern in anderen Availability Zones nicht betroffen ist. Jede ist so konzipiert, dass sie kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben AWS Region bietet. Indem Instances in separaten Availability Zones gestartet werden, können Sie Ihre Anwendungen vor den Fehlern eines einzelnen Standorts schützen. Weitere Informationen finden Sie unter [Auswählen von Regionen und Availability Zones](#).

Sie können Ihren Cluster in mehreren Availability Zones erstellen, eine Option, die als Multi-AZ-Bereitstellung bezeichnet wird. Wenn Sie diese Option wählen, stellt Amazon automatisch eine sekundäre Standby-Knoten-Instance in einer anderen Availability Zone bereit und verwaltet sie.

Ihre Primärknoten-Instance wird asynchron über Availability Zones auf die sekundäre Instance repliziert. Dieser Ansatz trägt dazu bei, Datenredundanz und Failover-Unterstützung bereitzustellen, Blockierungen zu vermeiden I/O und Latenzspitzen bei Systemsicherungen zu minimieren.

Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in Valkey und Redis OSS mit ElastiCache Multi-AZ](#).

Häufige ElastiCache Anwendungsfälle und wie ElastiCache Sie helfen können

Ob es sich um die neuesten Nachrichten, eine Top-10-Rangliste, einen Produktkatalog oder den Verkauf von Eintrittskarten für eine Veranstaltung handelt - Geschwindigkeit ist das A und O. Der Erfolg Ihrer Website und Ihres Unternehmens hängt stark von der Geschwindigkeit ab, mit der Sie Inhalte bereitstellen.

In dem Artikel "[Für ungeduldige Internetnutzer ist ein Augenzwinkern einfach zu lang, um zu warten](#)" stellte die New York Times fest, dass die Nutzer einen Unterschied von 250 Millisekunden (1/4 Sekunde) zwischen konkurrierenden Websites registrieren können. Die Nutzer neigen dazu, die langsamere Seite zugunsten der schnelleren Seite zu verlassen. Tests bei Amazon, die in dem Artikel [Wie die Ladezeit von Webseiten mit dem Verlust von Besuchern zusammenhängt](#) zitiert werden, ergaben, dass für jede Erhöhung der Ladezeit um 100 ms (1/10 Sekunde) die Verkaufszahlen um 1 Prozent sinken.

Wenn jemand Daten wünscht, können Sie diese Daten viel schneller bereitstellen, wenn sie im Cache gespeichert sind. Das gilt sowohl für eine Webseite als auch für einen Bericht, der als Grundlage für Geschäftsentscheidungen dient. Kann es sich Ihr Unternehmen leisten, Ihre Webseiten nicht zwischenspeichern, um sie so mit der kürzestmöglichen Latenz bereitzustellen?

Es mag intuitiv einleuchtend erscheinen, dass Sie die am häufigsten nachgefragten Artikel in den Cache stellen wollen. Aber warum sollten Sie nicht auch weniger häufig angefragte Elemente zwischenspeichern? Selbst eine optimierte Datenbankabfrage oder ein Remote-API-Aufruf ist deutlich langsamer als das Abrufen eines flachen Schlüssels aus einem In-Memory-Cache. Eine merkliche Verlangsamung führt dazu, dass Kunden abwandern.

Die folgenden Beispiele veranschaulichen einige Möglichkeiten, wie die Verwendung die Gesamtleistung Ihrer Anwendung verbessern ElastiCache kann.

Themen

- [In-Memory-Datastore](#)
- [Bestenlisten für Spiele](#)
- [Nachrichtenübermittlung \(Pub/Sub\)](#)
- [Empfehlungsdaten \(Hashes\)](#)
- [Semantisches Caching für generative KI-Anwendungen](#)
- [ElastiCache Kundenreferenzen](#)

In-Memory-Datastore

Der primäre Zweck eines Hauptspeicher-basierten Key-Value Store besteht darin, superschnellen (mit einer Latenz von unter einer Millisekunde) und kostengünstigen Zugriff auf Kopien von Daten bereitzustellen. Die meisten Datenspeicher haben Bereiche mit Daten, auf die häufig zugegriffen wird, die aber selten aktualisiert werden. Außerdem wird das Abfragen einer Datenbank immer langsamer und kostspieliger sein als das Auffinden eines Schlüssels in einem Schlüsselwertpaar-Cache. Manche Datenbankabfragen sind besonders aufwendig in der Durchführung. Ein Beispiel dafür sind Abfragen, die Joins über mehrere Tabellen oder Abfragen mit intensiven Berechnungen beinhalten. Durch die Zwischenspeicherung solcher Abfrageergebnisse zahlen Sie den Preis für die Abfrage nur einmal. Dann können Sie die Daten schnell und mehrfach abrufen, ohne die Abfrage erneut ausführen zu müssen.

Was sollte ich zwischenspeichern?

Bei der Entscheidung, welche Daten zwischengespeichert werden, sind folgende Faktoren zu berücksichtigen:

Geschwindigkeit und Kosten – Es ist immer langsamer und teurer, Daten aus einer Datenbank abzurufen als aus einem Cache. Manche Datenbankabfragen sind grundsätzlich langsamer und kostspieliger als andere. Beispielsweise sind Abfragen, bei denen mehrere Tabellen miteinander verknüpft werden, viel langsamer und teurer als einfache Abfragen, die nur eine Tabelle betreffen. Wenn die interessanten Daten eine langsame und teure Abfrage erfordern, sind sie ein Kandidat für das Caching. Wenn der Abruf der Daten eine relativ schnelle und einfache Abfrage erfordert, kann er je nach anderen Faktoren immer noch ein Kandidat für die Zwischenspeicherung sein.

Daten und Zugriffsmuster – Um zu bestimmen, was zwischengespeichert werden soll, müssen auch die Daten selbst und ihre Zugriffsmuster verstanden werden. So ist es beispielsweise nicht sinnvoll, Daten, die sich schnell ändern oder auf die nur selten zugegriffen wird, im Cache zu speichern. Damit die Zwischenspeicherung einen echten Nutzen bringt, sollten die Daten relativ statisch sein und häufig abgerufen werden. Ein Beispiel wäre ein persönliches Profil auf einer Social-Media-Website. Andererseits sollten Sie keine Daten zwischenspeichern, wenn das Zwischenspeichern keine Geschwindigkeits- oder Preisvorteile bringt. So ist es beispielsweise nicht sinnvoll, Webseiten, die Suchergebnisse liefern, in den Cache zu stellen, da die Abfragen und Ergebnisse in der Regel einzigartig sind.

Unvergänglichkeit – Per Definition sind zwischengespeicherte Daten veraltete Daten. Selbst wenn es unter bestimmten Umständen nicht veraltet ist, sollte es immer als veraltet betrachtet und behandelt

werden. Um festzustellen, ob Ihre Daten für die Zwischenspeicherung geeignet sind, müssen Sie die Toleranz Ihrer Anwendung gegenüber veralteten Daten ermitteln.

Es kann sein, dass Ihre Anwendung veraltete Daten in dem einen Kontext tolerieren kann, nicht jedoch in einem anderen. Nehmen wir zum Beispiel an, dass Ihre Website einen öffentlich gehandelten Aktienkurs anbietet. Ihre Kunden könnten eine gewisse Unbeständigkeit akzeptieren, wenn sie darauf hingewiesen werden, dass sich die Preise um n Minuten verzögern können. Aber wenn Sie diesen Aktienkurs einem Broker für einen Kauf oder Verkauf zur Verfügung stellen, benötigen Sie Echtzeitdaten.

Erwägen Sie die Zwischenspeicherung Ihrer Daten, wenn Folgendes zutrifft:

- Der Abruf Ihrer Daten ist im Vergleich zum Abruf aus dem Cache langsam oder teuer.
- Benutzer greifen häufig auf Ihre Daten zu.
- Ihre Daten bleiben relativ konstant, oder wenn sie sich schnell ändern, ist die Unbeständigkeit kein großes Problem.

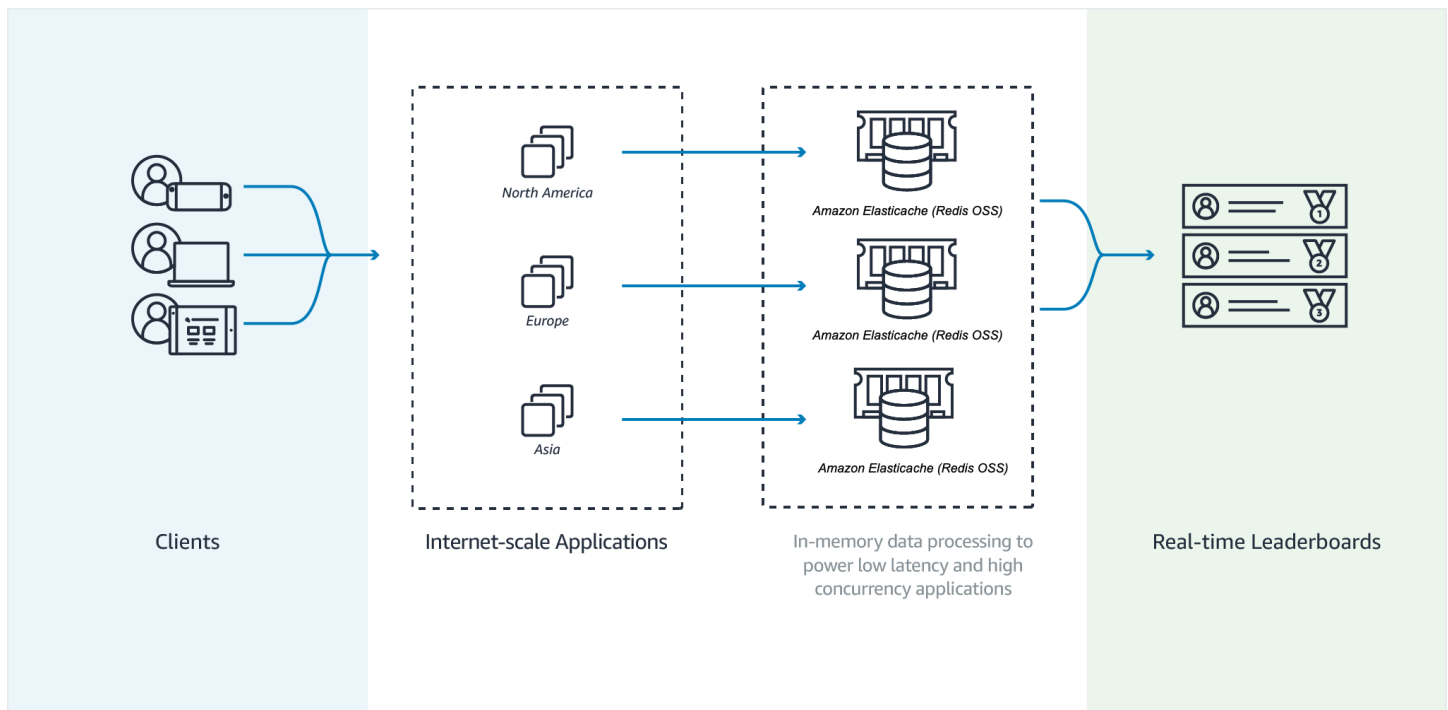
Weitere Informationen finden Sie unter [Caching-Strategien für Memcached](#).

Bestenlisten für Spiele

Mit sortierten Sets von Valkey oder Redis OSS können Sie die Rechenkomplexität von Bestenlisten von Ihrer Anwendung auf Ihren Cluster verlagern.

Bestenlisten, wie z. B. die 10 besten Ergebnisse eines Spiels, sind rechenintensiv. Dies gilt vor allem bei einer großen Anzahl gleichzeitiger Spieler und sich ständig ändernden Spielständen. Die sortierten Sätze von Valkey und Redis OSS garantieren sowohl Einzigartigkeit als auch Reihenfolge der Elemente. Bei sortierten Sätzen wird jedes Mal, wenn ein neues Element zur sortierten Menge hinzugefügt wird, dieses in Echtzeit neu eingestuft. Es wird dann in der numerisch richtigen Reihenfolge in die Menge eingefügt.

In der folgenden Abbildung können Sie sehen, wie eine ElastiCache Gaming-Bestenliste funktioniert.



Example Valkey oder Redis OSS-Bestenliste

In diesem Beispiel werden vier Spieler und ihre Ergebnisse mithilfe von ZADD in eine sortierte Liste eingegeben. Der Befehl ZREVRANGEBYSCORE listet die Spieler in absteigender Reihenfolge entsprechend ihrem Ergebnis auf. Als Nächstes wird ZADD verwendet, um das Ergebnis von June zu aktualisieren, indem der bestehende Eintrag überschrieben wird. Zum Schluss werden die Spieler von ZREVRANGEBYSCORE nach ihrer Punktzahl aufgelistet, von hoch bis niedrig. Die Liste zeigt, dass June in der Rangliste aufgestiegen ist.

```
ZADD leaderboard 132 Robert
ZADD leaderboard 231 Sandra
ZADD leaderboard 32 June
ZADD leaderboard 381 Adam

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
2) Sandra
3) Robert
4) June

ZADD leaderboard 232 June

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
```

- 2) June
- 3) Sandra
- 4) Robert

Mit dem folgenden Befehl erfährt June, auf welchem Platz sie unter allen Spielern steht. Da die Rangliste auf der Basis von Nullen erstellt wird, ergibt ZREVRANK eine 1 für June, die an zweiter Stelle steht.

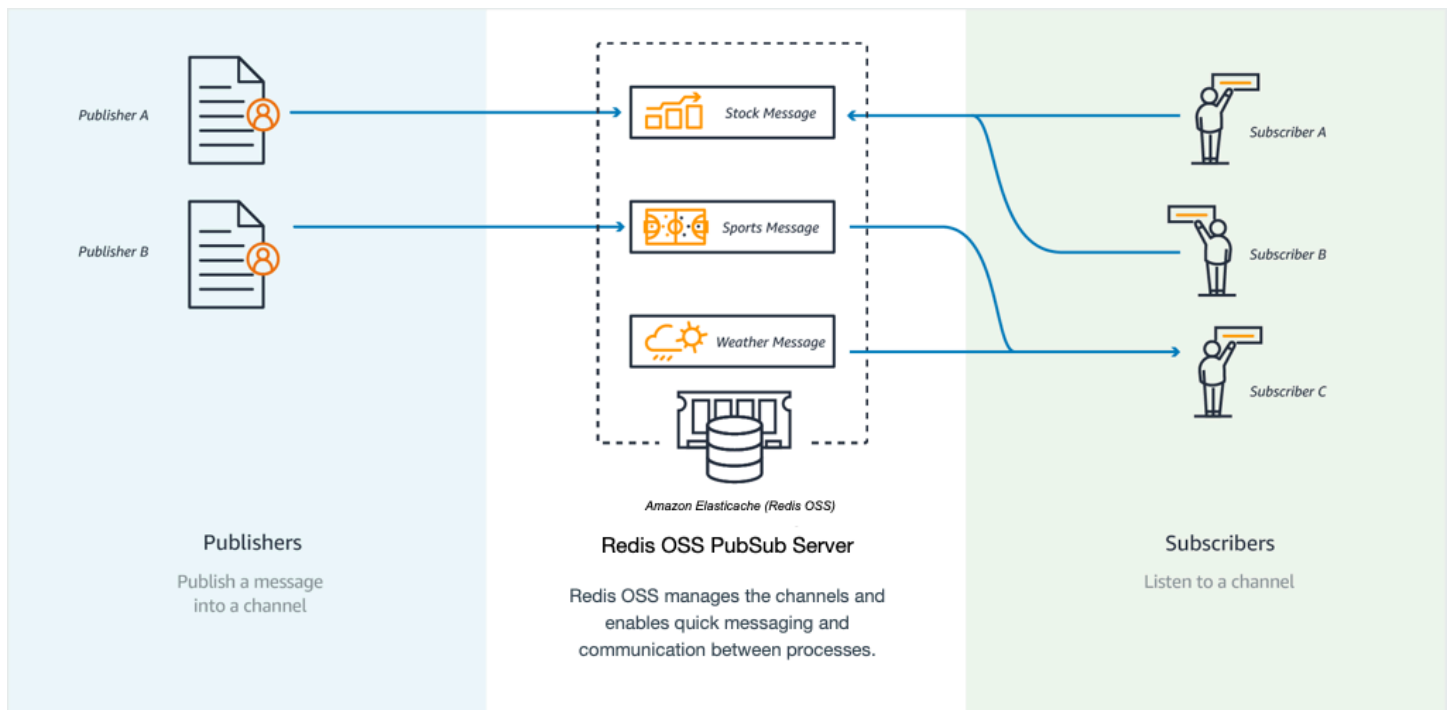
```
ZREVRANK leaderboard June  
1
```

Weitere Informationen finden Sie in der [Valkey-Dokumentation](#) zu sortierten Sets.

Nachrichtenübermittlung (Pub/Sub)

Wenn Sie eine E-Mail-Nachricht versenden, senden Sie sie an mindestens einen angegebenen Empfänger. Im Pub/Sub-Paradigma von Valkey und Redis OSS senden Sie eine Nachricht an einen bestimmten Kanal, ohne zu wissen, wer, wenn überhaupt, sie empfängt. Die Nachricht wird nur an die Personen gesendet, die den Kanal abonniert haben. Angenommen, Sie haben etwa den news.sports.golf-Channel abonniert. Sie und alle anderen, die den Kanal news.sports.golf abonniert haben, erhalten alle auf news.sports.golf veröffentlichten Nachrichten.

Die Pub/Sub-Funktionalität hat nichts mit einem Schlüsselraum zu tun. Daher gibt es auf keinem Level Beeinträchtigungen. In der folgenden Abbildung finden Sie eine Abbildung des ElastiCache Messagings mit Valkey und Redis OSS.



Abonnieren

Um Nachrichten in einem Kanal zu erhalten, abonnieren Sie den Kanal. Sie können einen einzelnen Channel, mehrere spezielle Channels oder alle Channels, die einem Muster entsprechen, abonnieren. Um ein Abonnement zu kündigen, melden Sie sich von dem Kanal ab, den Sie abonniert haben. Wenn Sie sich mit Hilfe eines Musterabgleichs angemeldet haben, können Sie sich auch mit demselben Muster wieder abmelden, das Sie zuvor verwendet haben.

Example– Abonnement eines einzelnen Channels

Um einen einzelnen Channel zu abonnieren, verwenden Sie den SUBSCRIBE-Befehl unter Angabe des Channels, den Sie abonnieren möchten. Im folgenden Beispiel abonniert ein Client den Channel `news.sports.golf`.

```
SUBSCRIBE news.sports.golf
```

Nach einer Weile kündigt der Client das Abonnement für den Channel mithilfe des UNSUBSCRIBE-Befehls unter Angabe des Channels, dessen Abonnement gekündigt werden soll.

```
UNSUBSCRIBE news.sports.golf
```


Example– Abonnement mehrerer ausgewählter Channels

Um mehrere bestimmte Channels zu abonnieren, listen Sie die Channels mit dem SUBSCRIBE-Befehl auf. Im folgenden Beispiel abonniert ein Client die Channels news.sports.golf, news.sports.soccer und news.sports.skiing.

```
SUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

Um ein Abonnement für einen bestimmten Kanal zu kündigen, verwenden Sie den Befehl UNSUBSCRIBE und geben Sie den Kanal an, von dem Sie sich abmelden möchten.

```
UNSUBSCRIBE news.sports.golf
```

Um Abonnements für mehrere Kanäle zu kündigen, verwenden Sie den Befehl UNSUBSCRIBE und geben Sie die Kanäle an, die Sie abbestellen möchten.

```
UNSUBSCRIBE news.sports.golf news.sports.soccer
```

Um alle Abonnements zu kündigen, verwenden Sie UNSUBSCRIBE und geben Sie jeden Kanal an. Oder verwenden Sie UNSUBSCRIBE ohne einen Kanal anzugeben.

```
UNSUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

oder

```
UNSUBSCRIBE
```

Example– Abonnements mithilfe von Musterabgleich

Clients können alle Channels abonnieren, die einem Muster entsprechen, indem sie den PSUBSCRIBE-Befehl verwenden.

Im folgenden Beispiel abonniert ein Client alle Sport-Channels. Sie listen nicht alle Sportkanäle einzeln auf, wie Sie es bei SUBSCRIBE tun. Mit dem PSUBSCRIBE-Befehl verwenden Sie stattdessen den Mustervergleich.

```
PSUBSCRIBE news.sports.*
```

Example Kündigen von Abonnements

Um Abonnements für diese Channels zu kündigen, verwenden Sie den PUNSUBSCRIBE-Befehl.

```
PUNSUBSCRIBE news.sports.*
```

Important

- Die Channel-Zeichenfolgen, die an einen [P]SUBSCRIBE-Befehl bzw. an den [P]UNSUBSCRIBE-Befehl gesendet werden, müssen übereinstimmen. PSUBSCRIBE zu news.* und PUNSUBSCRIBE von news.sports.* oder UNSUBSCRIBE von news.sports.golf ist nicht möglich.
- PSUBSCRIBE und PUNSUBSCRIBE sind nicht für ElastiCache Serverless verfügbar.

Veröffentlichen

Um eine Nachricht an alle Abonnenten eines Kanals zu senden, verwenden Sie den PUBLISH-Befehl, wobei Sie den Kanal und die Nachricht angeben. Im folgenden Beispiel wird diese Nachricht veröffentlicht: "Es ist Samstag und sonnig. Ich fahre zu den Links." auf den news.sports.golf-Kanal.

```
PUBLISH news.sports.golf "It's Saturday and sunny. I'm headed to the links."
```

Ein Client kann nicht auf einem Kanal veröffentlichen, den er abonniert hat.

Weitere Informationen finden Sie unter [Pub/Sub](#) in der Valkey-Dokumentation.

Empfehlungsdaten (Hashes)

Die Verwendung von INCR oder DECR in Valkey oder Redis OSS macht das Kompilieren von Empfehlungen einfach. Immer dann, wenn Benutzer ein Produkt "mögen", erhöhen Sie einen item:productID:like-Zähler. Immer dann, wenn Benutzer ein Produkt ein "dismögen", erhöhen Sie einen item:productID:dislike-Zähler. Mithilfe von Hashes können Sie auch eine Liste aller Personen führen, denen ein Produkt gefallen oder nicht gefallen hat.

Example– Likes und Dislikes

```
INCR item:38923:likes  
HSET item:38923:ratings Susan 1  
INCR item:38923:dislikes
```

HSET item:38923:ratings Tommy -1

Semantisches Caching für generative KI-Anwendungen

Der Betrieb generativer KI-Anwendungen in großem Maßstab kann aufgrund der Kosten und der Latenz, die mit Inferenzaufrufen für große Sprachmodelle verbunden sind, eine Herausforderung sein (). LLMs Sie können es ElastiCache für semantisches Caching in generativen KI-Anwendungen verwenden, wodurch Sie die Kosten und die Latenz von LLM-Inferenzaufrufen reduzieren können. Mit semantischem Caching können Sie eine zwischengespeicherte Antwort zurückgeben, indem Sie mithilfe eines vektorbasierten Abgleichs Ähnlichkeiten zwischen aktuellen und früheren Eingabeaufforderungen ermitteln. Wenn die Eingabeaufforderung eines Benutzers einer vorherigen Aufforderung semantisch ähnelt, wird eine zwischengespeicherte Antwort zurückgegeben, anstatt einen neuen LLM-Inferenzaufruf zu tätigen. Dadurch werden die Kosten für generative KI-Anwendungen gesenkt und schnellere Antworten bereitgestellt, die die Benutzererfahrung verbessern. Sie können steuern, welche Abfragen an den Cache weitergeleitet werden, indem Sie Ähnlichkeitsschwellenwerte für Eingabeaufforderungen konfigurieren und Tagfilter oder numerische Metadatenfilter anwenden.

Die Inline-Indexaktualisierungen in Echtzeit, die von Vector Search for bereitgestellt werden ElastiCache , stellen sicher, dass der Cache kontinuierlich aktualisiert wird, wenn Benutzeraufforderungen und LLM-Antworten eingehen. Diese Indizierung in Echtzeit ist entscheidend, um die Aktualität der zwischengespeicherten Ergebnisse und die Cache-Trefferquoten aufrechtzuerhalten, insbesondere bei hohem Datenverkehr. Darüber hinaus ElastiCache vereinfacht sie Operationen für das semantische Caching durch ausgereifte Cache-Primitive wie konfigurierbare Entfernungsstrategien pro Schlüssel TTLs, atomare Operationen und umfangreiche Datenstruktur- und Skriptunterstützung.

Speicher für generative KI-Assistenten und -Agenten

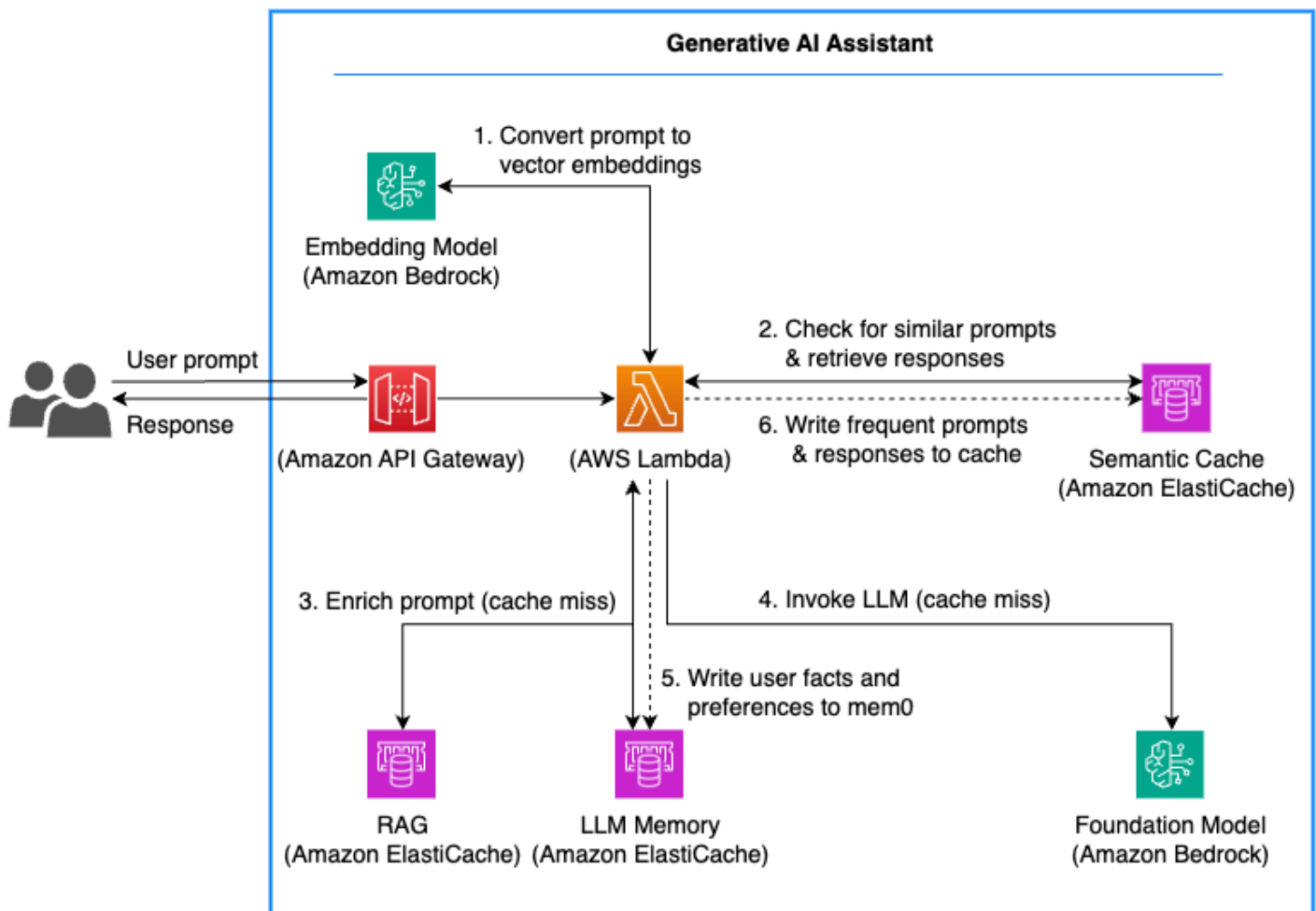
Sie können ElastiCache es verwenden, um personalisiertere, kontextsensivere Antworten zu geben, indem Sie Speichermechanismen implementieren, die sitzungsübergreifende Konversationsverläufe anzeigen. LLMs Das Konversationsgedächtnis ermöglicht generativen KI-Assistenten und -Agenten, vergangene Interaktionen zu speichern und zu nutzen, um Antworten zu personalisieren und die Relevanz zu verbessern. Das bloße Zusammenfassen aller vorherigen Interaktionen in der Aufforderung ist jedoch ineffektiv, da irrelevante zusätzliche Token die Kosten erhöhen, die Antwortqualität verschlechtern und das Risiko bergen, das Kontextfenster des LLM zu überschreiten. Stattdessen können Sie die Vektorsuche verwenden, um für jeden LLM-Aufruf nur die relevantesten Daten im Kontext abzurufen und bereitzustellen.

ElastiCache for Valkey bietet Integrationen mit Open-Source-Speicherschichten und bietet integrierte Konnektoren zum Speichern und Abrufen von Speichern für LLM-Anwendungen und -Agenten. Die Vektorsuche für ElastiCache ermöglicht schnelle Indexaktualisierungen, hält den Speicher auf dem neuesten Stand und macht neue Speicher sofort durchsuchbar. Die Vektorsuche mit niedriger Latenz ermöglicht eine schnelle Speichersuche, sodass sie im Online-Pfad jeder Anfrage implementiert werden kann, nicht nur bei Hintergrundaufgaben. Neben der Vektorsuche bietet ElastiCache for Valkey auch Caching-Primitive für Sitzungsstatus, Benutzereinstellungen und Feature-Flags und bietet so einen einzigen Dienst zum Speichern von kurzlebigen Sitzungszuständen und langfristigen „Erinnerungen“ in einem Datenspeicher.

Erweiterte Generierung (Retrieval Augmented Generation)

RAG ist der Prozess der Bereitstellung LLMs von up-to-date Informationen, um die Relevanz der Antworten zu verbessern. RAG reduziert Halluzinationen und verbessert die Genauigkeit der Fakten, indem es die Ergebnisse auf realen Datenquellen stützt. RAG-Anwendungen verwenden die Vektorsuche, um semantisch relevante Inhalte aus einer Wissensdatenbank abzurufen. Die von bereitgestellte Vektorsuche mit niedriger Latenz ElastiCache eignet sich für die Implementierung von RAG in Workloads mit großen Datensätzen mit Millionen von Vektoren und mehr. Darüber hinaus ElastiCache eignet sich die Unterstützung für Online-Vektorindex-Updates für Assistenten mit Upload-Workflows, die sicherstellen müssen, dass alle hochgeladenen Daten sofort durchsucht werden können. RAG stellt in KI-Systemen für Agenturen sicher, dass die Agenten up-to-date über Informationen verfügen, um präzise Maßnahmen ergreifen zu können. Die Vektorsuche mit niedriger Latenz ist auch für RAG in agentischen KI-Systemen von entscheidender Bedeutung, bei denen eine einzelne Abfrage mehrere LLM-Aufrufe auslösen und die Latenz der zugrunde liegenden Vektorsuche erhöhen kann.

Das folgende Diagramm zeigt eine Beispielarchitektur, mit ElastiCache der ein semantischer Cache, Speichermechanismen und RAG implementiert werden, um eine generative KI-Anwendung in der Produktion zu verbessern.



Semantische Suche

Die Vektorsuche ruft die relevantesten Text-, Sprach-, Bild- oder Videodaten auf der Grundlage ähnlicher Bedeutungen oder Merkmale ab. Diese Funktion ermöglicht Anwendungen für maschinelles Lernen, die auf der Ähnlichkeitssuche in verschiedenen Datenmodalitäten basieren, darunter Empfehlungsmaschinen, Anomalieerkennungs-, Personalisierungs- und Wissensmanagementsysteme. Empfehlungssysteme verwenden Vektordarstellungen, um komplexe Muster im Nutzerverhalten und in den Artikelmerkmalen zu erfassen, sodass sie die relevantesten Inhalte vorschlagen können. Die Vektorsuche eignet sich gut für diese Anwendungen, da sie fast in Echtzeit aktualisiert ElastiCache wird und die Latenz gering ist. Sie ermöglicht Ähnlichkeitsvergleiche, die sofortige, hochrelevante Empfehlungen auf der Grundlage von Benutzerinteraktionen in Echtzeit liefern.

ElastiCache Kundenreferenzen

Weitere Informationen darüber, wie Unternehmen wie Airbnb, PBS, Esri und andere Amazon nutzen, ElastiCache um ihr Geschäft durch ein verbessertes Kundenerlebnis auszubauen, finden Sie unter [So nutzen andere](#) Amazon. ElastiCache

Sie können sich auch die [Tutorial-Videos](#) für weitere Anwendungsfälle von ElastiCache Kunden ansehen.

Erste Schritte mit Amazon ElastiCache

Verwenden Sie das praktische Tutorial in diesem Abschnitt, um Ihnen den Einstieg zu erleichtern und mehr über die Verwendung ElastiCache zu erfahren.

Themen

- [Einrichten ElastiCache](#)
- [Erstellen Sie einen serverlosen Valkey-Cache](#)
- [Erstellen Sie einen serverlosen Redis OSS-Cache](#)
- [Einen serverlosen Memcached-Cache erstellen](#)
- [Tutorials: Erste Schritte mit Python und ElastiCache](#)
- [Tutorial: Lambda für den Zugriff ElastiCache in einer VPC konfigurieren](#)

Einrichten ElastiCache

Gehen Sie wie folgt vor, um den ElastiCache Webservice zu verwenden.

Themen

- [Melden Sie sich an für ein AWS-Konto](#)
- [Erstellen eines Benutzers mit Administratorzugriff](#)
- [Erteilen programmgesteuerten Zugriffs](#)
- [Richten Sie Ihre Berechtigungen ein \(nur ElastiCache für neue Benutzer\)](#)
- [Richten Sie ein EC2](#)
- [Gewähren des Netzwerkzugriffs aus einer Amazon-VPC-Sicherheitsgruppe für den Cache](#)
- [Laden Sie den Befehlszeilenzugriff herunter und richten Sie ihn ein](#)

Melden Sie sich 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-Kontos wird 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-Managementkonsole](#) als 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.

Erteilen programmgesteuerten Zugriffs

Benutzer benötigen programmatischen Zugriff, wenn sie mit AWS außerhalb des interagieren möchten.AWS-Managementkonsole Die Art und Weise, wie programmatischer Zugriff gewährt wird, hängt vom Benutzertyp ab, der zugreift.AWS

Um Benutzern programmgesteuerten Zugriff zu gewähren, wählen Sie eine der folgenden Optionen.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	(Empfohlen) Verwenden Sie Konsolenanmeldeinformationen als temporäre Anmeldeinformationen, um programmatische Anfragen an AWS CLI, AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zum AWS CLI finden Sie unter Anmeldung für AWS lokale Entwicklung im AWS Command Line Interface Benutzerhandbuch. • Weitere Informationen finden Sie unter Anmeldung für AWS lokale Entwicklung im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch AWS SDKs
Mitarbeiteridentität (Benutzer, die in IAM Identity Center verwaltet werden)	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI, AWS SDKs, oder AWS APIs zu signieren.	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zu den AWS CLI finden Sie unter Konfiguration der AWS CLI zur Verwendung AWS IAM Identity Center im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs, Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch AWS

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
		SDKs und im Tools-Referenzhandbuch.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI/AWS SDKs, oder zu signieren. AWS APIs	Folgen Sie den Anweisungen unter Verwenden temporärer Anmeldeinformationen mit AWS Ressourcen im IAM-Benutzerhandbuch.
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an das AWS CLI/AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu AWS CLI finden Sie unter Authentifizierung mithilfe von IAM-Benutzeranmeldinformationen im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldeinformationen im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch. • Weitere Informationen finden Sie unter Verwaltung von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch. AWS APIs

Verwandte Themen:

- [Was ist IAM](#) im IAM-Benutzerhandbuch.
- AWSAWS Allgemeine Referenz zu [Sicherheitsanmeldedaten](#).

Richten Sie Ihre Berechtigungen ein (nur ElastiCache für neue Benutzer)

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter [Erstellen eines Berechtigungssatzes](#) im AWS IAM Identity Center-Benutzerhandbuch.

- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter [Eine Rolle für einen externen Identitätsanbieter \(Verbund\) erstellen](#) im IAM-Benutzerhandbuch.

- IAM-Benutzer:

- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter [Eine Rolle für einen IAM-Benutzer erstellen](#) im IAM-Benutzerhandbuch.
- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter [Hinzufügen von Berechtigungen zu einem Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Amazon ElastiCache erstellt und verwendet servicebezogene Rollen, um in Ihrem Namen Ressourcen bereitzustellen und auf andere AWS Ressourcen und Services zuzugreifen. ElastiCache Um eine servicebezogene Rolle für Sie zu erstellen, verwenden Sie die AWS verwaltete Richtlinie mit dem Namen `AmazonElastiCacheFullAccess`. Diese Rolle besitzt die vordefinierte Berechtigung, die der Service benötigt, um eine Service-verknüpfte Rolle für Sie zu erstellen.

Sie können sich entscheiden, anstelle der Standardrichtlinie eine benutzerseitig verwaltete Richtlinie zu verwenden. Stellen Sie in diesem Fall sicher, dass Sie entweder über die Berechtigung zum Aufrufen von `iam:createServiceLinkedRole` verfügen oder die serviceverknüpfte ElastiCache-Rolle bereits erstellt wurde.

Weitere Informationen finden Sie hier:

- [Erstellen einer neuen Richtlinie](#)(IAM)

- [AWS verwaltete Richtlinien für Amazon ElastiCache](#)
- [Verwenden von serviceverknüpften Rollen für Amazon ElastiCache](#)

Richten Sie ein EC2

Sie müssen eine EC2 Instanz einrichten, von der aus Sie eine Verbindung zu Ihrem Cache herstellen.

- Wenn Sie noch keine EC2 Instance haben, erfahren Sie hier, wie Sie eine EC2 Instance einrichten: [Amazon EC2 Getting Started Guide](#).
- Ihre EC2 Instance muss sich in derselben VPC befinden und dieselben Sicherheitsgruppeneinstellungen wie Ihr Cache haben. Standardmäßig ElastiCache erstellt Amazon einen Cache in Ihrer Standard-VPC und verwendet die Standardsicherheitsgruppe. Um diesem Tutorial zu folgen, stellen Sie sicher, dass sich Ihre EC2 Instance in der Standard-VPC befindet und über die Standardsicherheitsgruppe verfügt.

Gewähren des Netzwerkzugriffs aus einer Amazon-VPC-Sicherheitsgruppe für den Cache

ElastiCache knotenbasierte Cluster verwenden Port 6379 für Valkey- und Redis OSS-Befehle, und ElastiCache Serverless verwendet sowohl Port 6379 als auch Port 6380. Um erfolgreich eine Verbindung herzustellen und Valkey- oder Redis OSS-Befehle von Ihrer EC2 Instance aus auszuführen, muss Ihre Sicherheitsgruppe bei Bedarf Zugriff auf diese Ports gewähren.

ElastiCache for Memcached verwendet die Ports 11211 und 11212, um Memcached-Befehle zu akzeptieren. Um erfolgreich eine Verbindung herzustellen und Memcached-Befehle von Ihrer EC2 Instance aus auszuführen, muss Ihre Sicherheitsgruppe den Zugriff auf diese Ports zulassen.

1. Melden Sie sich bei der an AWS Command Line Interface und öffnen Sie die [EC2 Amazon-Konsole](#).
2. Wählen Sie in der Navigationsleiste unter Network & Security die Option Security Groups aus.
3. Wählen Sie aus der Liste der Sicherheitsgruppen die Sicherheitsgruppe Ihrer Amazon VPC aus. Sofern Sie keine Sicherheitsgruppe zur ElastiCache Verwendung erstellt haben, wird diese Sicherheitsgruppe als Standard bezeichnet.
4. Klicken Sie auf die Registerkarte „Eingehend“ und führen Sie anschließend folgende Schritte aus.

- a. Wählen Sie Bearbeiten aus.
- b. Wählen Sie Regel hinzufügen aus.
- c. Wählen Sie in der Spalte „Typ“ die Option Benutzerdefinierte TCP-Regel aus.
- d. Wenn Sie Valkey oder Redis OSS verwenden, geben Sie in das Feld Portbereich den Text ein. 6379

Wenn Sie Memcached verwenden, geben Sie in das Feld Portbereich den Text ein. 11211

- e. Wählen Sie im Feld Quelle die Option Anywhere mit dem Portbereich (0.0.0.0/0) aus, sodass jede EC2 Amazon-Instance, die Sie in Ihrer Amazon VPC starten, eine Verbindung zu Ihrem Cache herstellen kann.
- f. Wenn Sie ElastiCache Serverless verwenden, fügen Sie eine weitere Regel hinzu, indem Sie Regel hinzufügen wählen.
- g. Wählen Sie in der Spalte Typ die Option Benutzerdefinierte TCP-Regel aus.
- h. Wenn Sie es ElastiCache für Redis OSS verwenden, geben Sie in das Feld Portbereich den Text ein. 6380

Wenn Sie ElastiCache für Memcached verwenden, geben Sie in das Feld Portbereich den Text ein. 11212

- i. Wählen Sie im Feld Quelle die Option Anywhere mit dem Portbereich (0.0.0.0/0) aus, sodass jede EC2 Amazon-Instance, die Sie in Ihrer Amazon VPC starten, eine Verbindung zu Ihrem Cache herstellen kann.
- j. Wählen Sie Speichern aus.

Laden Sie den Befehlszeilenzugriff herunter und richten Sie ihn ein

Laden Sie das Valkey-Cli-Hilfsprogramm herunter und installieren Sie es.

Wenn Sie ElastiCache für Valkey verwenden, finden Sie das Valkey-Cli-Hilfsprogramm möglicherweise nützlich. Wenn Sie ElastiCache für Redis OSS mit redis-cli verwenden, sollten Sie erwägen, zu valkey-cli zu wechseln, da es auch für Redis OSS funktioniert.

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her. Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2 Erste Schritte“](#).

2. Laden Sie das valkey-cli Utility herunter und installieren Sie es, indem Sie den entsprechenden Befehl für Ihr Setup ausführen.

Amazon Linux 2

```
sudo amazon-linux-extras install epel -y
sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel -y
wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
tar xvzf valkey-8.0.0.tar.gz
cd valkey-8.0.0
make BUILD_TLS=yes
```

Note

- Beim Installieren des redis6-Pakets wird redis6-cli mit standardmäßiger Verschlüsselungsunterstützung installiert.
- Bei der Installation von valkey-cli oder redis-cli ist es wichtig, Build-Unterstützung für TLS zu haben. ElastiCache Serverless ist nur zugänglich, wenn TLS aktiviert ist.
- Wenn der Cluster, mit dem Sie eine Verbindung herstellen, nicht verschlüsselt ist, benötigen Sie die Option Build_TLS=yes nicht.

Erstellen Sie einen serverlosen Valkey-Cache

In diesem Schritt erstellen Sie einen neuen Cache in Amazon ElastiCache.

AWS-Managementkonsole

So erstellen Sie mit der ElastiCache Konsole einen neuen Cache:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey-Caches aus.
3. Wählen Sie auf der rechten Seite der Konsole die Option Valkey-Cache erstellen
4. Geben Sie in den Cache-Einstellungen einen Namen ein. Sie können optional eine Beschreibung für den Cache eingeben.

5. Lassen Sie die Standardeinstellungen ausgewählt.
6. Klicken Sie auf Erstellen, um den Cache zu erstellen.
7. Sobald sich der Cache im Status „AKTIV“ befindet, können Sie mit dem Schreiben und Lesen von Daten in den Cache beginnen.

AWS CLI

Das folgende AWS CLI Beispiel erstellt einen neuen Cache mit create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine valkey
```

Windows

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name CacheName ^  
  --engine valkey
```

Beachten Sie, dass für das Feld „Status“ der Wert CREATING festgelegt ist.

Verwenden Sie den describe-serverless-caches Befehl, ElastiCache um zu überprüfen, ob die Erstellung des Caches abgeschlossen ist.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Fahren Sie nach dem Erstellen des neuen Caches mit [Daten in den Cache lesen und in den Cache schreiben](#) fort.

Daten in den Cache lesen und in den Cache schreiben

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon EC2 Getting Started Guide](#).

In diesem Abschnitt wird auch davon ausgegangen, dass Sie VPC-Zugriffs- und Sicherheitsgruppeneinstellungen für die EC2 Instance eingerichtet haben, von der aus Sie eine Verbindung zu Ihrem Cache herstellen, und Valkey-CLI auf Ihrer Instance eingerichtet haben. EC2 Weitere Informationen zu diesem Schritt finden Sie unter [Einrichten ElastiCache](#).

Wenn Sie über eine große oder globale Anwendung verfügen, können Sie zusätzlich zu den unten aufgeführten Schritten die Leseleistung erheblich steigern, indem Sie Replikate erstellen und aus ihnen lesen. Weitere Informationen zu diesem fortgeschritteneren Schritt finden Sie unter [Bewährte Methoden für die Verwendung von Read Replicas](#).

Finden Ihres Cache-Endpunkts

AWS-Managementkonsole

So finden Sie den Endpunkt Ihres Caches mithilfe der ElastiCache Konsole:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey-Caches aus.
3. Klicken Sie auf der rechten Seite der Konsole auf den Namen des Caches, den Sie gerade erstellt haben.
4. Suchen Sie in den Cache-Details den Cache-Endpunkt und kopieren Sie ihn.

AWS CLI

Das folgende AWS CLI Beispiel zeigt, wie Sie mit dem Befehl den Endpunkt für Ihren neuen Cache finden. `describe-serverless-caches` Nachdem Sie den Befehl ausgeführt haben, suchen Sie nach dem Feld „Endpunkt“.

Linux

```
aws elasticache describe-serverless-caches \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^  
--serverless-cache-name CacheName
```

Connect zu Ihrem Valkey Cache her (Linux)

Jetzt, da Sie den benötigten Endpunkt haben, können Sie sich bei Ihrer EC2 Instance anmelden und eine Verbindung zum Cache herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Der folgende Befehl stellt eine Verbindung mit einem Cache her. (Hinweis: Ersetzen Sie `cache-endpoint` durch den Endpunkt, den Sie im vorherigen Schritt abgerufen haben.)

```
src/valkey-cli -h cache-endpoint --tls -p 6379  
set a "hello"           // Set key "a" with a string value and no expiration  
OK  
get a                   // Get value for key "a"  
"hello"
```

Connect zu Ihrem Valkey Cache her (Windows)

Jetzt, da Sie den benötigten Endpunkt haben, können Sie sich bei Ihrer EC2 Instance anmelden und eine Verbindung zum Cache herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Mit dem folgenden Befehl stellen Sie eine Verbindung mit einem Cache her. Öffnen Sie die Befehlszeile, wechseln Sie zum Valkey- oder Redis-OSS-Verzeichnis und führen Sie den Befehl aus (Hinweis: Ersetzen Sie `Cache_Endpoint` durch den Endpunkt, den Sie im vorherigen Schritt abgerufen haben).

```
c:\Valkey>valkey-cli -h Valkey_Cluster_Endpoint --tls -p 6379  
set a "hello"           // Set key "a" with a string value and no expiration  
OK  
get a                   // Get value for key "a"  
"hello"
```

Sie können jetzt mit [\(Optional\) Bereinigen](#) fortfahren.

(Optional) Bereinigen

Wenn Sie den von Ihnen erstellten ElastiCache Amazon-Cache nicht mehr benötigen, können Sie ihn löschen. Mit diesem Schritt wird sichergestellt, dass Ihnen keine Ressourcen in Rechnung gestellt

werden, die Sie nicht nutzen. Sie können die ElastiCache Konsole, die oder die ElastiCache API verwenden AWS CLI, um Ihren Cache zu löschen.

AWS-Managementkonsole

So löschen Sie Ihren Cache mithilfe der Konsole:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey Caches aus.
3. Wählen Sie das Optionsfeld neben dem Cache, dem Sie löschen möchten.
4. Wählen Sie oben rechts Aktionen und dann Löschen aus.
5. Sie können sich optional dafür entscheiden, einen abschließenden Snapshot zu erstellen, bevor Sie Ihren Cache löschen.
6. Geben Sie auf dem Bestätigungsbildschirm Löschen den Cache-Namen erneut ein und wählen Sie die Option Löschen aus, um den Cluster zu löschen, oder Abbrechen, um den Cluster beizubehalten.

Sobald Ihr Cache in den Status DELETING wechselt, fallen für ihn keine Gebühren mehr an.

AWS CLI

Im folgenden AWS CLI Beispiel wird ein Cache mithilfe des Befehls gelöscht. delete-serverless-cache

Linux

```
aws elasticache delete-serverless-cache \  
--serverless-cache-name CacheName
```

Windows

```
aws elasticache delete-serverless-cache ^  
--serverless-cache-name CacheName
```

Beachten Sie, dass für das Feld Status der Wert DELETING festgelegt ist.

Sie können jetzt mit [Nächste Schritte](#) fortfahren.

Nächste Schritte

Weitere Informationen dazu ElastiCache finden Sie auf den folgenden Seiten:

- [Arbeiten mit ElastiCache](#)
- [Skalierung ElastiCache](#)
- [Protokollierung und Überwachung in Amazon ElastiCache](#)
- [ElastiCache Best Practices und Caching-Strategien](#)
- [Snapshot und Wiederherstellung](#)
- [Amazon SNS SNS-Überwachung von Ereignissen ElastiCache](#)

Erstellen Sie einen serverlosen Redis OSS-Cache

In diesem Schritt erstellen Sie einen neuen Cache in Amazon ElastiCache.

AWS-Managementkonsole

So erstellen Sie mit der ElastiCache Konsole einen neuen Cache:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Redis OSS-Caches aus.
3. Wählen Sie auf der rechten Seite der Konsole die Option Redis OSS-Cache erstellen
4. Geben Sie in den Cache-Einstellungen einen Namen ein. Sie können optional eine Beschreibung für den Cache eingeben.
5. Lassen Sie die Standardeinstellungen ausgewählt.
6. Klicken Sie auf Erstellen, um den Cache zu erstellen.
7. Sobald sich der Cache im Status „AKTIV“ befindet, können Sie mit dem Schreiben und Lesen von Daten in den Cache beginnen.

AWS CLI

Das folgende AWS CLI Beispiel erstellt einen neuen Cache mit create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine redis
```

Windows

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name CacheName ^  
  --engine redis
```

Beachten Sie, dass für das Feld „Status“ der Wert CREATING festgelegt ist.

Verwenden Sie den `describe-serverless-caches` Befehl, ElastiCache um zu überprüfen, ob die Erstellung des Caches abgeschlossen ist.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Fahren Sie nach dem Erstellen des neuen Caches mit [Daten in den Cache lesen und in den Cache schreiben](#) fort.

Daten in den Cache lesen und in den Cache schreiben

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon EC2 Getting Started Guide](#).

In diesem Abschnitt wird auch davon ausgegangen, dass Sie VPC-Zugriffs- und Sicherheitsgruppeneinstellungen für die EC2 Instance eingerichtet haben, von der aus Sie eine Verbindung zu Ihrem Cache herstellen, und Valkey-CLI auf Ihrer Instance eingerichtet haben. EC2 Weitere Informationen zu diesem Schritt finden Sie unter [Einrichten ElastiCache](#).

Finden Ihres Cache-Endpunkts

AWS-Managementkonsole

So finden Sie den Endpunkt Ihres Caches mithilfe der Konsole: ElastiCache

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey caches Redis OSS caches aus.
3. Klicken Sie auf der rechten Seite der Konsole auf den Namen des Caches, den Sie gerade erstellt haben.
4. Suchen Sie in den Cache-Details den Cache-Endpoint und kopieren Sie ihn.

AWS CLI

Das folgende AWS CLI Beispiel zeigt, wie Sie mit dem Befehl den Endpunkt für Ihren neuen Cache finden. `describe-serverless-caches` Nachdem Sie den Befehl ausgeführt haben, suchen Sie nach dem Feld „Endpunkt“.

Linux

```
aws elasticache describe-serverless-caches \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^
  --serverless-cache-name CacheName
```

Connect zu Ihrem Valkey- oder Redis OSS-Cache her (Linux)

Jetzt, da Sie den Endpunkt haben, den Sie benötigen, können Sie sich bei Ihrer EC2 Instance anmelden und eine Verbindung zum Cache herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Der folgende Befehl stellt eine Verbindung mit einem Cache her. (Hinweis: Ersetzen Sie `cache-endpoint` durch den Endpunkt, den Sie im vorherigen Schritt abgerufen haben.)

```
src/valkey-cli -h cache-endpoint --tls -p 6379
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
```

```
"hello"
```

Connect zu Ihrem Valkey- oder Redis OSS-Cache her (Windows)

Da Sie nun den benötigten Endpunkt haben, können Sie sich bei Ihrer EC2 Instance anmelden und eine Verbindung zum Cache herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Mit dem folgenden Befehl stellen Sie eine Verbindung mit einem Cache her. Öffnen Sie die Befehlszeile, wechseln Sie zum Valkey-Verzeichnis und führen Sie den Befehl aus (Hinweis: Ersetzen Sie `Cache_Endpoint` durch den Endpunkt, den Sie im vorherigen Schritt abgerufen haben).

```
c:\Redis>valkey-cli -h Redis_Cluster_Endpoint --tls -p 6379
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
```

Sie können jetzt mit [\(Optional\) Bereinigen](#) fortfahren.

(Optional) Bereinigen

Wenn Sie den von Ihnen erstellten ElastiCache Amazon-Cache nicht mehr benötigen, können Sie ihn löschen. Mit diesem Schritt wird sichergestellt, dass Ihnen keine Ressourcen in Rechnung gestellt werden, die Sie nicht nutzen. Sie können die ElastiCache Konsole, die oder die ElastiCache API verwenden AWS CLI, um Ihren Cache zu löschen.

AWS-Managementkonsole

So löschen Sie Ihren Cache mithilfe der Konsole:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey oder Redis OSS Caches aus.
3. Wählen Sie das Optionsfeld neben dem Cache, dem Sie löschen möchten.
4. Wählen Sie oben rechts Aktionen und dann Löschen aus.
5. Sie können sich optional dafür entscheiden, einen abschließenden Snapshot zu erstellen, bevor Sie Ihren Cache löschen.

6. Geben Sie auf dem Bestätigungsbildschirm Löschen den Cache-Namen erneut ein und wählen Sie die Option Löschen aus, um den Cluster zu löschen, oder Abbrechen, um den Cluster beizubehalten.

Sobald Ihr Cache in den Status DELETING wechselt, fallen für ihn keine Gebühren mehr an.

AWS CLI

Im folgenden AWS CLI Beispiel wird ein Cache mit dem Befehl gelöscht. delete-serverless-cache

Linux

```
aws elasticache delete-serverless-cache \  
--serverless-cache-name CacheName
```

Windows

```
aws elasticache delete-serverless-cache ^  
--serverless-cache-name CacheName
```

Beachten Sie, dass für das Feld Status der Wert DELETING festgelegt ist.

Sie können jetzt mit [Nächste Schritte](#) fortfahren.

Nächste Schritte

Weitere Informationen dazu ElastiCache finden Sie auf den folgenden Seiten:

- [Arbeiten mit ElastiCache](#)
- [Skalierung ElastiCache](#)
- [Protokollierung und Überwachung in Amazon ElastiCache](#)
- [ElastiCache Best Practices und Caching-Strategien](#)
- [Snapshot und Wiederherstellung](#)
- [Amazon SNS SNS-Überwachung von Ereignissen ElastiCache](#)

Einen serverlosen Memcached-Cache erstellen

AWS-Managementkonsole

So erstellen Sie mit der Konsole einen neuen serverlosen Memcached-Cache: ElastiCache

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Memcached-Caches.
3. Wählen Sie auf der rechten Seite der Konsole Memcached-Cache erstellen aus.
4. Geben Sie in den Cache-Einstellungen einen Namen ein. Sie können optional eine Beschreibung für den Cache eingeben.
5. Lassen Sie die Standardeinstellungen ausgewählt.
6. Klicken Sie auf Erstellen, um den Cache zu erstellen.
7. Sobald sich der Cache im Status „AKTIV“ befindet, können Sie mit dem Schreiben und Lesen von Daten in den Cache beginnen.

Um einen neuen Cache mit dem zu erstellen AWS CLI

Das folgende AWS CLI Beispiel erstellt einen neuen Cache mit create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine memcached
```

Windows

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name CacheName ^  
  --engine memcached
```

Beachten Sie, dass für das Feld „Status“ der Wert CREATING festgelegt ist.

Verwenden Sie den describe-serverless-caches Befehl, ElastiCache um zu überprüfen, ob die Erstellung des Caches abgeschlossen ist.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Fahren Sie nach dem Erstellen des neuen Caches mit [Daten in den Cache lesen und in den Cache schreiben](#) fort.

Daten in den Cache lesen und in den Cache schreiben

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon-Leitfaden „EC2 Erste Schritte“](#).

ElastiCache Erstellt standardmäßig einen Cache in Ihrer Standard-VPC. Stellen Sie sicher, dass Ihre EC2 Instance auch in der Standard-VPC erstellt wurde, damit sie eine Verbindung zum Cache herstellen kann.

Finden Ihres Cache-Endpunkts

AWS-Managementkonsole

So finden Sie den Endpunkt Ihres Caches mithilfe der ElastiCache Konsole:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Memcached-Caches.
3. Klicken Sie auf der rechten Seite der Konsole auf den Namen des Caches, den Sie gerade erstellt haben.
4. Suchen Sie in den Cache-Details den Cache-Endpunkt und kopieren Sie ihn.

AWS CLI

Das folgende AWS CLI Beispiel zeigt, wie Sie mit dem describe-serverless-caches Befehl den Endpunkt für Ihren neuen Cache ermitteln können. Nachdem Sie den Befehl ausgeführt haben, suchen Sie nach dem Feld „Endpunkt“.

Linux

```
aws elasticache describe-serverless-caches \
```

```
--serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^  
--serverless-cache-name CacheName
```

Herstellen einer Verbindung mithilfe von OpenSSL

Weitere Informationen zum Herstellen einer Verbindung mithilfe von OpenSSL finden Sie unter [ElastiCache Verschlüsselung bei der Übertragung \(TLS\)](#).

Herstellen einer Verbindung mithilfe des Memcached-Java-Clients

```
import java.security.KeyStore;  
import javax.net.ssl.SSLContext;  
import javax.net.ssl.TrustManagerFactory;  
import net.spy.memcached.AddrUtil;  
import net.spy.memcached.ConnectionFactoryBuilder;  
import net.spy.memcached.FailureMode;  
import net.spy.memcached.MemcachedClient;  
  
public class TLSDemo {  
    public static void main(String[] args) throws Exception {  
        ConnectionFactoryBuilder connectionFactoryBuilder = new  
ConnectionFactoryBuilder();  
        // Build SSLContext  
        TrustManagerFactory tmf =  
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());  
        tmf.init((KeyStore) null);  
        SSLContext sslContext = SSLContext.getInstance("TLS");  
        sslContext.init(null, tmf.getTrustManagers(), null);  
        // Create the client in TLS mode  
        connectionFactoryBuilder.setSSLContext(sslContext);  
        // Set Failure Mode to Retry  
        connectionFactoryBuilder.setFailureMode(FailureMode.Retry);  
        MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),  
AddrUtil.getAddresses("mycluster-fnjyzo.serverless.use1.cache.amazonaws.com:11211"));  
  
        // Store a data item for an hour.  
        client.set("theKey", 3600, "This is the data value");  
    }  
}
```

```
}
```

Herstellen einer Verbindung mithilfe des Memcached-PHP-Clients

```
<?php
$cluster_endpoint = "mycluster.serverless.use1.cache.amazonaws.com";
$server_port = 11211;

/* Initialize a persistent Memcached client in TLS mode */
$tls_client = new Memcached('persistent-id');
$tls_client->addServer($cluster_endpoint, $server_port);
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastError_message(), "\n";
    exit(1);
}
$tls_config = new MemcachedTLSTextContextConfig();
$tls_config->hostname = '*.serverless.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;
$tls_client->createAndSetTLSTextContext((array)$tls_config);

/* store the data for 60 seconds in the cluster */
$tls_client->set('key', 'value', 60);
?>
```

Herstellen einer Verbindung mithilfe des Memcached-Python-Clients (PyMemcache)

Siehe https://pymemcache.readthedocs.io/en/latest/getting_started.html

```
import ssl
from pymemcache.client.base import Client

context = ssl.create_default_context()
cluster_endpoint = <To be taken from the AWS CLI / console>
target_port = 11211
memcached_client = Client("{cluster_endpoint}", target_port, tls_context=context)
memcached_client.set("key", "value", expire=500, noreply=False)
assert self.memcached_client.get("key").decode() == "value"
```

Connect mit dem NodeJS/TS Memcached-Client her (Electrode-IO Memcache)

<https://github.com/electrode-io/Siehe> Memcache und <https://www.npmjs.com/package/Memcache-Client>

Installieren über npm `i memcache-client`

Erstellen Sie in der Anwendung wie folgt einen Memcached-TLS-Client:

```
var memcache = require("memcache-client");
const client = new memcache.MemcacheClient({server: "{cluster_endpoint}:11211", tls:
  {}});
client.set("key", "value");
```

Herstellen einer Verbindung mithilfe des Memcached-Rust-Clients (rust-memcache)

Weitere [Informationen finden Sie unter https://crates.io/crates/memcache](https://crates.io/crates/memcache) und <https://github.com/aisk/Rust-Memcache>.

```
// create connection with to memcached server node:
let client = memcache::connect("memcache+tls://{cluster_endpoint}:11211?
verify_mode=none").unwrap();

// set a string value
client.set("foo", "bar", 0).unwrap();
```

Herstellen einer Verbindung mithilfe des Memcached-Go-Clients (Gomemcache)

<https://github.com/bradfitz/Siehe Gomemcache>

```
c := New(net.JoinHostPort("{cluster_endpoint}", strconv.Itoa(port)))
c.DialContext = func(ctx context.Context, network, addr string) (net.Conn, error) {
var td tls.Dialer
td.Config = &tls.Config{}
return td.DialContext(ctx, network, addr)
}
foo := &Item{Key: "foo", Value: []byte("fooval"), Flags: 123}
err := c.Set(foo)
```

Herstellen einer Verbindung mithilfe des Memcached-Ruby-Clients (Dalli)

[Siehst du Dalli https://github.com/petergoldstein/](https://github.com/petergoldstein/)

```
require 'dalli'
ssl_context = OpenSSL::SSL::SSLContext.new
```

```
ssl_context.ssl_version = :SSLv23
ssl_context.verify_hostname = true
ssl_context.verify_mode = OpenSSL::SSL::VERIFY_PEER
client = Dalli::Client.new("<cluster_endpoint>:11211", :ssl_context => ssl_context);
client.get("abc")
```

Connect mit Memcached.NET-Client herstellen () EnyimMemcachedCore

Siehe <https://github.com/cnblogs/EnyimMemcachedCore>

```
"MemcachedClient": {
  "Servers": [
    {
      "Address": "{cluster_endpoint}",
      "Port": 11211
    }
  ],
  "UseSslStream": true
}
```

Sie können jetzt mit [\(Optional\) Bereinigen](#) fortfahren.

(Optional) Bereinigen

Mit dem AWS-Managementkonsole

Mit dem folgenden Verfahren wird ein einzelner Cache aus Ihrer Bereitstellung gelöscht. Wenn Sie mehrere Caches löschen möchten, wiederholen Sie das Verfahren für jeden Cache, den Sie löschen möchten. Sie brauchen nicht zu warten, bis ein Cache fertig gelöscht ist, bevor Sie den Vorgang zum Löschen eines anderen Caches starten.

So löschen Sie einen Cache

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Konsolen-Dashboard die Engine aus, die auf dem Cache läuft, den Sie löschen möchten. Es wird eine Liste mit allen Caches, in denen diese Engine ausgeführt wird, angezeigt.
3. Wählen Sie den Namen des Caches aus der Liste der Caches aus, um den zu löschenden Cache anzugeben.

Important

Sie können jeweils nur einen Cache aus der ElastiCache Konsole löschen. Werden mehrere Caches ausgewählt, wird der Löschvorgang deaktiviert.

4. Klicken Sie bei Actions auf Delete.
5. Wählen Sie auf dem Bestätigungsbildschirm Cache löschen die Option Löschen aus, um den Cache zu löschen, oder Abbrechen, wenn Sie den Cache luster beibehalten möchten.
6. Wenn Sie Löschen auswählen, ändert sich der Status des Caches in deleting.

Sobald Ihr Cache in den Status DELETING wechselt, fallen für ihn keine Gebühren mehr an.

Mit dem AWS CLI

Der folgende Code löscht den Cache my-cache.

```
aws elasticache delete-serverless-cache --serverless-cache-name my-cache
```

Die `delete-serverless-cache` CLI-Aktion löscht nur einen serverlosen Cache. Um mehrere Caches zu löschen, rufen Sie `delete-serverless-cache` jeden serverlosen Cache auf, den Sie löschen möchten. Sie brauchen nicht zu warten, bis ein Serverless-Cache fertig gelöscht ist, bevor Sie einen anderen löschen.

Für Linux, macOS oder Unix:

```
aws elasticache delete-serverless-cache \  
  --serverless-cache-name my-cache
```

Für Windows:

```
aws elasticache delete-serverless-cache ^  
  --serverless-cache-name my-cache
```

Weitere Informationen finden Sie im AWS CLI For-Thema. ElastiCache `delete-serverless-cache`

Sie können jetzt mit [Nächste Schritte](#) fortfahren.

Nächste Schritte

Weitere Informationen dazu finden ElastiCache Sie unter:

- [Arbeiten mit ElastiCache](#)
- [Skalierung ElastiCache](#)
- [Kontingente für ElastiCache](#)
- [ElastiCache Best Practices und Caching-Strategien](#)
- [ElastiCache Ereignisse anzeigen](#)

Tutorials: Erste Schritte mit Python und ElastiCache

Dieser Abschnitt enthält praktische Tutorials, die Ihnen helfen, mehr über Valkey und Redis OSS zu ElastiCache erfahren. Wir empfehlen Ihnen, eines der sprachspezifischen Tutorials durchzuarbeiten.

Note

AWS SDKs sind für eine Vielzahl von Sprachen verfügbar. Eine vollständige Liste finden Sie unter [Tools für Amazon Web Services](#).

Themen

- [Python und ElastiCache](#)

Python und ElastiCache

In diesem Tutorial verwenden Sie das AWS SDK for Python (Boto3), um einfache Programme für die folgenden ElastiCache Operationen zu schreiben:

- ElastiCache Für Redis OSS-Cluster erstellen (Clustermodus aktiviert und Clustermodus deaktiviert)
- Prüfen Sie, ob Benutzer oder Benutzergruppen existieren, andernfalls erstellen Sie sie. (Diese Funktion ist mit Valkey 7.2 und höher sowie mit Redis OSS 6.0 bis 7.1 verfügbar.)
- Verbinden mit ElastiCache
- Führe Operationen wie das Einstellen und Abrufen von Zeichenketten, das Lesen von und Schreiben in Streams sowie das Veröffentlichen und Abonnieren von Inhalten über den Kanal aus. Pub/Sub

Während Sie dieses Tutorial durcharbeiten, können Sie sich auf die Dokumentation zum AWS SDK für Python (Boto) beziehen. Der folgende Abschnitt ist spezifisch für ElastiCache: [ElastiCache Low-Level-Client](#)

Tutorial-Voraussetzungen

- Richten Sie einen AWS Zugriffsschlüssel ein, um den zu verwenden. AWS SDKs Weitere Informationen finden Sie unter [Einrichten ElastiCache](#).
- Python 3.0 oder später installieren. Weitere Informationen finden Sie unter <https://www.python.org/downloads>. Weiter Informationen finden Sie unter [Quickstart](#) in der Boto 3-Dokumentation.

Themen

- [Tutorial: ElastiCache Cluster und Benutzer erstellen](#)
- [Tutorial: Verbindung herstellen mit ElastiCache](#)
- [Verwendungsbeispiele](#)

Tutorial: ElastiCache Cluster und Benutzer erstellen

In den folgenden Beispielen werden das boto3 SDK ElastiCache für Redis OSS-Verwaltungsvorgänge (Cluster- oder Benutzererstellung) und redis-py-cluster redis-py/ für die Datenverarbeitung verwendet.

Themen

- [Erstellen Sie einen Cluster mit deaktiviertem Cluster-Modus](#)
- [Erstellen eines Clusters mit deaktiviertem Cluster-Modus mit TLS und RBAC](#)
- [Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus](#)
- [Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus mit TLS und RBAC](#)
- [Prüfen Sie, ob sie users/usergroup existieren, andernfalls erstellen Sie sie](#)

Erstellen Sie einen Cluster mit deaktiviertem Cluster-Modus

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen .py ein.
CreateClusterModeDisabledCluster

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticsearch')

def
    create_cluster_mode_disabled(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumCacheClusters=1,
    cluster',ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode disabled

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
    cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
    CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
    used.
    :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
    node) and maximum 6 (1 primary and 5 replicas).
    If not specified, cluster will be created with 1 primary and 1 replica.
```

```
:param ReplicationGroupDescription: Description for the cluster.
:param ReplicationGroupId: Name for the cluster
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return 'ReplicationGroupId parameter is required'

response = client.create_replication_group(
    AutomaticFailoverEnabled=True,
    CacheNodeType=CacheNodeType,
    Engine='valkey',
    EngineVersion=EngineVersion,
    NumCacheClusters=NumCacheClusters,
    ReplicationGroupDescription=ReplicationGroupDescription,
    ReplicationGroupId=ReplicationGroupId,
    SnapshotRetentionLimit=30,
)
return response

if __name__ == '__main__':

    # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large
    nodes, Valkey 8.0, one primary and two replicas
    elasticacheResponse = create_cluster_mode_disabled(
        #CacheNodeType='cache.m6g.large',
        EngineVersion='8.0',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104053'
    )

    logging.info(elasticacheResponse)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python CreateClusterModeDisabledCluster.py
```

Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Erstellen eines Clusters mit deaktivierten Cluster-Modus mit TLS und RBAC

Um die Sicherheit zu gewährleisten, können Sie Transport Layer Security (TLS) und rollenbasierte Zugriffssteuerung (RBAC) beim Erstellen eines Clusters mit deaktiviertem Cluster-Modus verwenden. Im Gegensatz zu Valkey oder Redis OSS AUTH, bei denen alle authentifizierten Clients vollen Zugriff auf die Replikationsgruppe haben, wenn ihr Token authentifiziert ist, können Sie mit RBAC den Clusterzugriff über Benutzergruppen steuern. Diese Benutzergruppen dienen dazu, den Zugriff auf Replikationsgruppen zu organisieren. Weitere Informationen finden Sie unter [Rollenbasierte Zugriffskontrolle \(RBAC\)](#).

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen RBAC.py ein.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
create_cluster_mode_disabled_rbac(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumCacheC
cluster',ReplicationGroupId=None, UserGroupIds=None,
SecurityGroupIds=None,CacheSubnetGroupName=None):
    """Creates an ElastiCache Cluster with cluster mode disabled and RBAC

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
used.
    :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
node) and maximum 6 (1 primary and 5 replicas).
    If not specified, cluster will be created with 1 primary and 1 replica.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Mandatory name for the cluster.
    :param UserGroupIds: The ID of the user group to be assigned to the cluster.
    :param SecurityGroupIds: List of security groups to be assigned. If not defined,
default will be used
```

```

:param CacheSubnetGroupName: subnet group where the cluster will be placed. If not
defined, default will be used.
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return {'Error': 'ReplicationGroupId parameter is required'}
elif not isinstance(UserGroupIds,(list)):
    return {'Error': 'UserGroupIds parameter is required and must be a list'}

params={'AutomaticFailoverEnabled': True,
        'CacheNodeType': CacheNodeType,
        'Engine': 'valkey',
        'EngineVersion': EngineVersion,
        'NumCacheClusters': NumCacheClusters,
        'ReplicationGroupDescription': ReplicationGroupDescription,
        'ReplicationGroupId': ReplicationGroupId,
        'SnapshotRetentionLimit': 30,
        'TransitEncryptionEnabled': True,
        'UserGroupIds':UserGroupIds
    }

# defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
if isinstance(SecurityGroupIds,(list)):
    params.update({'SecurityGroupIds':SecurityGroupIds})
if CacheSubnetGroupName:
    params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

response = client.create_replication_group(**params)
return response

if __name__ == '__main__':

    # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large
    nodes, Valkey 8.0, one primary and two replicas.
    # Assigns the existent user group "mygroup" for RBAC authentication

    response=create_cluster_mode_disabled_rbac(
        CacheNodeType='cache.m6g.large',
        EngineVersion='8.0',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104',
        UserGroupIds=[

```

```
        'mygroup'
    ],
    SecurityGroupIds=[
        'sg-7cc73803'
    ],
    CacheSubnetGroupName='default'
)

logging.info(response)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ClusterModeDisabledWithRBAC.py
```

Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen `ClusterModeEnabled.py` ein.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
    create_cluster_mode_enabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumNodeGroups=1,
    ReplicationGroupDescription='Sample cache with cluster mode
    enabled', ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode enabled

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
    cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
    CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
    used.
    :param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximum 90.
    If not specified, cluster will be created with 1 shard.
```

```

:param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
replica per shard will be created.
:param ReplicationGroupDescription: Description for the cluster.
:param ReplicationGroupId: Name for the cluster
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return 'ReplicationGroupId parameter is required'

response = client.create_replication_group(
    AutomaticFailoverEnabled=True,
    CacheNodeType=CacheNodeType,
    Engine='valkey',
    EngineVersion=EngineVersion,
    ReplicationGroupDescription=ReplicationGroupDescription,
    ReplicationGroupId=ReplicationGroupId,
    # Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
node (implicit) and 2 replicas (replicasPerNodeGroup)
    NumNodeGroups=NumNodeGroups,
    ReplicasPerNodeGroup=ReplicasPerNodeGroup,
    CacheParameterGroupName='default.valkey7.2.cluster.on'
)

return response

# Creates a cluster mode enabled
response = create_cluster_mode_enabled(
    CacheNodeType='cache.m6g.large',
    EngineVersion='6.0',
    ReplicationGroupDescription='Valkey cluster mode enabled with replicas',
    ReplicationGroupId='valkey20210',
    # Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
(implicit) and 2 replicas (replicasPerNodeGroup)
    NumNodeGroups=2,
    ReplicasPerNodeGroup=1,
)

logging.info(response)

```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ClusterModeEnabled.py
```

Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus mit TLS und RBAC

Um die Sicherheit zu gewährleisten, können Sie Transport Layer Security (TLS) und rollenbasierte Zugriffssteuerung (RBAC) beim Erstellen eines Clusters mit aktiviertem Cluster-Modus verwenden. Im Gegensatz zu Valkey oder Redis OSS AUTH, bei denen alle authentifizierten Clients vollen Zugriff auf die Replikationsgruppe haben, wenn ihr Token authentifiziert ist, können Sie mit RBAC den Clusterzugriff über Benutzergruppen steuern. Diese Benutzergruppen dienen dazu, den Zugriff auf Replikationsgruppen zu organisieren. Weitere Informationen finden Sie unter [Rollenbasierte Zugriffskontrolle \(RBAC\)](#).

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen RBAC.py ein.

ClusterModeEnabledWith

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
    create_cluster_mode_enabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumNodeGroups=1,
    ReplicationGroupDescription='Sample cache with cluster
    mode enabled', ReplicationGroupId=None, UserGroupIds=None,
    SecurityGroupIds=None, CacheSubnetGroupName=None, CacheParameterGroupName='default.valkey7.2.clu
    """Creates an ElastiCache Cluster with cluster mode enabled and RBAC

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
    cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
    CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
    used.
    :param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximum 90.
    If not specified, cluster will be created with 1 shard.
    :param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
    replica per shard will be created.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Name for the cluster.
```



```

:param CacheParameterGroupName: Parameter group to be used. Must be compatible with
the engine version and cluster mode enabled.
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return 'ReplicationGroupId parameter is required'
elif not isinstance(UserGroupIds,(list)):
    return {'Error': 'UserGroupIds parameter is required and must be a list'}

params={'AutomaticFailoverEnabled': True,
        'CacheNodeType': CacheNodeType,
        'Engine': 'valkey',
        'EngineVersion': EngineVersion,
        'ReplicationGroupDescription': ReplicationGroupDescription,
        'ReplicationGroupId': ReplicationGroupId,
        'SnapshotRetentionLimit': 30,
        'TransitEncryptionEnabled': True,
        'UserGroupIds':UserGroupIds,
        'NumNodeGroups': NumNodeGroups,
        'ReplicasPerNodeGroup': ReplicasPerNodeGroup,
        'CacheParameterGroupName': CacheParameterGroupName
    }

# defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
if isinstance(SecurityGroupIds,(list)):
    params.update({'SecurityGroupIds':SecurityGroupIds})
if CacheSubnetGroupName:
    params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

response = client.create_replication_group(**params)
return response

if __name__ == '__main__':
    # Creates a cluster mode enabled cluster
    response = create_cluster_mode_enabled(
        CacheNodeType='cache.m6g.large',
        EngineVersion='7.2',
        ReplicationGroupDescription='Valkey cluster mode enabled with replicas',
        ReplicationGroupId='valkey2021',
        # Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
        (implicit) and 2 replicas (replicasPerNodeGroup)
        NumNodeGroups=2,
        ReplicasPerNodeGroup=1,

```

```
UserGroupIds=[
    'mygroup'
],
SecurityGroupIds=[
    'sg-7cc73803'
],
CacheSubnetGroupName='default'

)

logging.info(response)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ClusterModeEnabledWithRBAC.py
```

Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Prüfen Sie, ob sie users/usergroup existieren, andernfalls erstellen Sie sie

Mithilfe von RBAC erstellen Sie Benutzer und weisen ihnen spezifische Berechtigungen zu, indem Sie eine Zugriffszeichenfolge verwenden, wie im Folgenden beschrieben. Sie weisen die Benutzer Benutzergruppen zu, denen eine bestimmte Rolle zugewiesen ist (Administratoren, Personalabteilung), die dann einer oder mehreren ElastiCache Redis OSS-Replikationsgruppen zugewiesen werden. Auf diese Weise können Sie Sicherheitsgrenzen zwischen Clients einrichten, die dieselbe Valkey- oder Redis OSS-Replikationsgruppe (n) verwenden, und verhindern, dass Clients gegenseitig auf die Daten zugreifen. Weitere Informationen finden Sie unter [Rollenbasierte Zugriffskontrolle \(RBAC\)](#).

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen `UserAndUserGroups.py` ein. Aktualisieren Sie den Mechanismus für die Bereitstellung von Anmeldeinformationen. In diesem Beispiel werden Anmeldeinformationen als ersetzbar angezeigt und ihnen wird ein nicht deklariertes Element zugewiesen. Vermeiden Sie die Hartkodierung von Anmeldeinformationen.

In diesem Beispiel wird eine Zugriffszeichenfolge mit den Berechtigungen für den Benutzer verwendet. Weitere Informationen zu Zugriffszeichenfolgen finden Sie unter [Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge](#).

```
import boto3
import logging
```

```
logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def check_user_exists(UserId):
    """Checks if UserId exists

    Returns True if UserId exists, otherwise False
    :param UserId: ElastiCache User ID
    :return: True|False
    """
    try:
        response = client.describe_users(
            UserId=UserId,
        )
        if response['Users'][0]['UserId'].lower() == UserId.lower():
            return True
    except Exception as e:
        if e.response['Error']['Code'] == 'UserNotFound':
            logging.info(e.response['Error'])
            return False
        else:
            raise

def check_group_exists(UserGroupId):
    """Checks if UserGroupId exists

    Returns True if Group ID exists, otherwise False
    :param UserGroupId: ElastiCache User ID
    :return: True|False
    """

    try:
        response = client.describe_user_groups(
            UserGroupId=UserGroupId
        )
        if response['UserGroups'][0]['UserGroupId'].lower() == UserGroupId.lower():
            return True
    except Exception as e:
        if e.response['Error']['Code'] == 'UserGroupNotFound':
            logging.info(e.response['Error'])
            return False
        else:
            raise
```

```
def create_user(UserId=None, Username=None, Password=None, AccessString=None):
    """Creates a new user

    Returns the ARN for the newly created user or the error message
    :param UserId: ElastiCache user ID. User IDs must be unique
    :param Username: ElastiCache user name. ElastiCache allows multiple users with the
    same name as long as the associated user ID is unique.
    :param Password: Password for user. Must have at least 16 chars.
    :param AccessString: Access string with the permissions for the user.
    :return: user ARN
    """
    try:
        response = client.create_user(
            UserId=UserId,
            Username=Username,
            Engine='Redis',
            Passwords=[Password],
            AccessString=AccessString,
            NoPasswordRequired=False
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])
        return e.response['Error']

def create_group(UserGroupId=None, UserIds=None):
    """Creates a new group.

    A default user is required (mandatory) and should be specified in the UserIds list

    Return: Group ARN
    :param UserIds: List with user IDs to be associated with the new group. A default
    user is required
    :param UserGroupId: The ID (name) for the group
    :return: Group ARN
    """
    try:
        response = client.create_user_group(
            UserGroupId=UserGroupId,
            Engine='Redis',
            UserIds=UserIds
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])
```

```
if __name__ == '__main__':

    groupName='mygroup2'
    userName = 'myuser2'
    userId=groupName+'-'+userName

    # Creates a new user if the user ID does not exist.
    for tmpUserId,tmpUserName in [ (userId,userName), (groupName+'-
default','default')]:
        if not check_user_exists(tmpUserId):
            response=create_user(UserId=tmpUserId,
UserName=EXAMPLE,Password=EXAMPLE,AccessString='on ~* +@all')
            logging.info(response)
            # assigns the new user ID to the user group
        if not check_group_exists(groupName):
            UserIds = [ userId , groupName+'-default']
            response=create_group(UserGroupId=groupName,UserIds=UserIds)
            logging.info(response)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python UserAndUserGroups.py
```

Tutorial: Verbindung herstellen mit ElastiCache

In den folgenden Beispielen wird der Valkey- oder Redis OSS-Client verwendet, um eine Verbindung herzustellen. ElastiCache

Themen

- [Herstellen einer Verbindung zu einem Cluster mit deaktiviertem Cluster-Modus](#)
- [Herstellen einer Verbindung zu einem Cluster mit aktiviertem Cluster-Modus](#)

Herstellen einer Verbindung zu einem Cluster mit deaktiviertem Cluster-Modus

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens ConnectClusterModeDisabled.py ein. Aktualisieren Sie den Mechanismus für die Bereitstellung von Anmeldeinformationen. In diesem Beispiel werden Anmeldeinformationen als ersetzbar angezeigt

und ihnen wird ein nicht deklariertes Element zugewiesen. Vermeiden Sie die Hartkodierung von Anmeldeinformationen.

```
from redis import Redis
import logging

logging.basicConfig(level=logging.INFO)
redis = Redis(host='primary.xxx.yyyyyy.zzz1.cache.amazonaws.com', port=6379,
              decode_responses=True, ssl=True, username=example, password=EXAMPLE)

if redis.ping():
    logging.info("Connected to Redis")
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ConnectClusterModeDisabled.py
```

Herstellen einer Verbindung zu einem Cluster mit aktiviertem Cluster-Modus

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen ConnectClusterModeEnabled.py ein.

```
from rediscluster import RedisCluster
import logging

logging.basicConfig(level=logging.INFO)
redis = RedisCluster(startup_nodes=[{"host":
    "xxx.yyy.clustercfg.zzz1.cache.amazonaws.com", "port": "6379"}],
                    decode_responses=True, skip_full_coverage_check=True)

if redis.ping():
    logging.info("Connected to Redis")
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ConnectClusterModeEnabled.py
```

Verwendungsbeispiele

In den folgenden Beispielen wird das Boto3-SDK für die Arbeit mit ElastiCache ElastiCache Redis OSS verwendet.

Themen

- [Zeichenfolgen festlegen und abrufen](#)
- [Festlegen und Abrufen eines Hash mit mehreren Elementen](#)
- [Von einem Kanal aus veröffentlichen \(schreiben\) und abonnieren \(lesen\) Pub/Sub](#)
- [Schreiben und Lesen aus einem Stream](#)

Zeichenfolgen festlegen und abrufen

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens `.py` ein.

SetAndGetStrings

```
import time
import logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s: %(message)s')

keyName='mykey'
currTime=time.ctime(time.time())

# Set the key 'mykey' with the current date and time as value.
# The Key will expire and removed from cache in 60 seconds.
redis.set(keyName, currTime, ex=60)

# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)

# Retrieve the key value and current TTL
keyValue=redis.get(keyName)
keyTTL=redis.ttl(keyName)

logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
    keyValue, keyTTL))
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python SetAndGetStrings.py
```

Festlegen und Abrufen eines Hash mit mehreren Elementen

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens `SetAndGetHash.py` ein.

```
import logging
```

```
import time

logging.basicConfig(level=logging.INFO,format='%(asctime)s: %(message)s')

keyName='mykey'
keyValues={'datetime': time.ctime(time.time()), 'epochtime': time.time()}

# Set the hash 'mykey' with the current date and time in human readable format
# (datetime field) and epoch number (epochtime field).
redis.hset(keyName, mapping=keyValues)

# Set the key to expire and removed from cache in 60 seconds.
redis.expire(keyName, 60)

# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)

# Retrieves all the fields and current TTL
keyValues=redis.hgetall(keyName)
keyTTL=redis.ttl(keyName)

logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
keyValues, keyTTL))
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python SetAndGetHash.py
```

Von einem Kanal aus veröffentlichen (schreiben) und abonnieren (lesen) Pub/Sub

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens PubAndSub.py ein.

```
import logging
import time

def handlerFunction(message):
    """Prints message got from PubSub channel to the log output

    Return None
    :param message: message to log
    """
    logging.info(message)

logging.basicConfig(level=logging.INFO)
```



```

redis = Redis(host="redis202104053.tihewd.ng.0001.use1.cache.amazonaws.com", port=6379,
    decode_responses=True)

# Creates the subscriber connection on "mychannel"
subscriber = redis.pubsub()
subscriber.subscribe(**{'mychannel': handlerFunction})

# Creates a new thread to watch for messages while the main process continues with its
    routines
thread = subscriber.run_in_thread(sleep_time=0.01)

# Creates publisher connection on "mychannel"
redis.publish('mychannel', 'My message')

# Publishes several messages. Subscriber thread will read and print on log.
while True:
    redis.publish('mychannel',time.ctime(time.time()))
    time.sleep(1)

```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python PubAndSub.py
```

Schreiben und Lesen aus einem Stream

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens ReadWriteStream.py ein.

```

from redis import Redis
import redis.exceptions as exceptions
import logging
import time
import threading

logging.basicConfig(level=logging.INFO)

def writeMessage(streamName):
    """Starts a loop writting the current time and thread name to 'streamName'

    :param streamName: Stream (key) name to write messages.
    """
    fieldsDict={'writerId':threading.currentThread().getName(),'myvalue':None}
    while True:

```

```

        fieldsDict['myvalue'] = time.ctime(time.time())
        redis.xadd(streamName,fieldsDict)
        time.sleep(1)

def readMessage(groupName=None,streamName=None):
    """Starts a loop reading from 'streamName'
    Multiple threads will read from the same stream consumer group. Consumer group is
    used to coordinate data distribution.
    Once a thread acknowledges the message, it won't be provided again. If message
    wasn't acknowledged, it can be served to another thread.

    :param groupName: stream group where multiple threads will read.
    :param streamName: Stream (key) name where messages will be read.
    """

    readerID=threading.currentThread().getName()
    while True:
        try:
            # Check if the stream has any message
            if redis.xlen(streamName)>0:
                # Check if the messages are new (not acknowledged) or not (already
                processed)
                streamData=redis.xreadgroup(groupName,readerID,
{streamName:'>'},count=1)
                if len(streamData) > 0:
                    msgId,message = streamData[0][1][0]
                    logging.info("{}: Got {} from ID
{}".format(readerID,message,msgId))
                    #Do some processing here. If the message has been processed
                    successfully, acknowledge it and (optional) delete the message.
                    redis.xack(streamName,groupName,msgId)
                    logging.info("Stream message ID {} read and processed successfully
by {}".format(msgId,readerID))
                    redis.xdel(streamName,msgId)
                else:
                    pass
            except:
                raise

            time.sleep(0.5)

# Creates the stream 'mystream' and consumer group 'myworkergroup' where multiple
threads will write/read.
try:

```

```
    redis.xgroup_create('mystream','myworkergroup',mkstream=True)
except exceptions.ResponseError as e:
    logging.info("Consumer group already exists. Will continue despite the error:
    {}".format(e))
except:
    raise

# Starts 5 writer threads.
for writer_no in range(5):
    writerThread = threading.Thread(target=writeMessage, name='writer-'+str(writer_no),
    args=('mystream',),daemon=True)
    writerThread.start()

# Starts 10 reader threads
for reader_no in range(10):
    readerThread = threading.Thread(target=readMessage, name='reader-'+str(reader_no),
    args=('myworkergroup','mystream',),daemon=True)
    readerThread.daemon = True
    readerThread.start()

# Keep the code running for 30 seconds
time.sleep(30)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ReadWriteStream.py
```

Tutorial: Lambda für den Zugriff ElastiCache in einer VPC konfigurieren

In diesem Tutorial erfahren Sie, wie Sie einen ElastiCache serverlosen Cache erstellen, eine Lambda-Funktion erstellen, dann die Lambda-Funktion testen und optional anschließend bereinigen.

Themen

- [Schritt 1: Einen serverlosen Cache erstellen ElastiCache .](#)
- [Schritt 2: Erstellen Sie eine Lambda-Funktion für ElastiCache](#)
- [Schritt 3: Testen Sie die Lambda-Funktion mit ElastiCache](#)
- [Schritt 4: Aufräumen \(optional\)](#)

Schritt 1: Einen serverlosen Cache erstellen ElastiCache .

Gehen Sie folgendermaßen vor, um einen serverlosen Cache zu erstellen.

Schritt 1.1: Erstellen Sie einen serverlosen Cache

In diesem Schritt erstellen Sie mithilfe der (CLI) einen serverlosen Cache in der Standard-Amazon-VPC in der Region us-east-1 in Ihrem Konto. AWS Command Line Interface Informationen zum Erstellen eines serverlosen Caches mithilfe der ElastiCache Konsole oder API finden Sie unter.

[Erstellen Sie einen serverlosen Redis OSS-Cache](#)

```
aws elasticache create-serverless-cache \
  --serverless-cache-name cache-01 \
  --description "ElastiCache IAM auth application" \
  --engine valkey
```

Beachten Sie, dass für das Feld „Status“ der Wert CREATING festgelegt ist. Es kann eine Minute dauern ElastiCache , bis der Cache fertig erstellt ist.

Schritt 1.2: Serverlosen Cache-Endpunkt kopieren

Stellen Sie sicher, dass ElastiCache für Redis OSS die Erstellung des Caches mit dem `describe-serverless-caches` Befehl abgeschlossen ist.

```
aws elasticache describe-serverless-caches \
  --serverless-cache-name cache-01
```

Kopieren Sie die in der Ausgabe angezeigte Endpunktadresse. Sie benötigen diese Adresse, wenn Sie das Bereitstellungspaket für Ihre Lambda-Funktion erstellen.

Schritt 1.3: IAM-Rolle erstellen

1. Erstellen Sie, wie unten dargestellt, ein Dokument mit den IAM-Vertrauensrichtlinien für Ihre Rolle, sodass Ihr Konto die neue Rolle übernehmen kann. Speichern Sie die Richtlinie in einer Datei namens `trust-policy.json`.

JSON

```
{
```

```

"Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::*:role/*"
  },
  {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::*:role/*"
  }]
}

```

- Erstellen Sie ein IAM-Richtliniendokument wie im Folgenden dargestellt. Speichern Sie die Richtlinie in einer Datei namens `policy.json`.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticache:Connect"
      ],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:cache-01",
        "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"
      ]
    }
  ]
}

```

- Erstellen Sie eine IAM-Rolle.

```

aws iam create-role \
--role-name "elasticache-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json

```

- Erstellen Sie die IAM-Richtlinie.

```
aws iam create-policy \  
  --policy-name "elasticache-allow-all" \  
  --policy-document file:///policy.json
```

5. Fügen Sie die IAM-Richtlinie an die Rolle an.

```
aws iam attach-role-policy \  
  --role-name "elasticache-iam-auth-app" \  
  --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

Schritt 1.4: Erstellen Sie einen Standardbenutzer

1. Erstellen Sie einen neuen Standardbenutzer.

```
aws elasticache create-user \  
  --user-name default \  
  --user-id default-user-disabled \  
  --engine redis \  
  --authentication-mode Type=no-password-required \  
  --access-string "off +get ~keys*"
```

2. Erstellen Sie einen neuen IAM-fähigen Benutzer.

```
aws elasticache create-user \  
  --user-name iam-user-01 \  
  --user-id iam-user-01 \  
  --authentication-mode Type=iam \  
  --engine redis \  
  --access-string "on ~* +@all"
```

3. Erstellen Sie eine Benutzergruppe und fügen Sie einen Benutzer an.

```
aws elasticache create-user-group \  
  --user-group-id iam-user-group-01 \  
  --engine redis \  
  --user-ids default-user-disabled iam-user-01  
  
aws elasticache modify-serverless-cache \  
  --serverless-cache-name cache-01 \  
  --user-group-id iam-user-group-01
```

Schritt 2: Erstellen Sie eine Lambda-Funktion für ElastiCache

Gehen Sie wie folgt vor, um eine Lambda-Funktion für den Zugriff auf den ElastiCache Cache zu erstellen.

Schritt 2.1: Erstellen Sie eine Lambda-Funktion

In diesem Tutorial stellen wir Beispielcode in Python für Ihre Lambda-Funktion bereit.

Python

Der folgende Python-Beispielcode liest und schreibt ein Element in Ihren ElastiCache Cache.

Kopieren Sie den Code in eine Datei und speichern Sie diese mit dem Namen `app.py`. Achten Sie darauf, den `elasticache_endpoint` Wert im Code durch die Endpunktadresse zu ersetzen, die Sie im vorherigen Schritt kopiert haben.

```
from typing import Tuple, Union
from urllib.parse import ParseResult, urlencode, urlunparse

import boto3.session
import redis
from boto3.model import ServiceId
from boto3.signers import RequestSigner
from cachetools import TTLCache, cached
import uuid

class ElastiCacheIAMProvider(redis.CredentialProvider):
    def __init__(self, user, cache_name, is_serverless=False, region="us-east-1"):
        self.user = user
        self.cache_name = cache_name
        self.is_serverless = is_serverless
        self.region = region

        session = boto3.session.get_session()
        self.request_signer = RequestSigner(
            ServiceId("elasticache"),
            self.region,
            "elasticache",
            "v4",
            session.get_credentials(),
            session.get_component("event_emitter"),
        )
```

```

# Generated IAM tokens are valid for 15 minutes
@cached(cache=TTLCache(maxsize=128, ttl=900))
def get_credentials(self) -> Union[Tuple[str], Tuple[str, str]]:
    query_params = {"Action": "connect", "User": self.user}
    if self.is_serverless:
        query_params["ResourceType"] = "ServerlessCache"
    url = urlunparse(
        ParseResult(
            scheme="https",
            netloc=self.cache_name,
            path="/",
            query=urlencode(query_params),
            params="",
            fragment="",
        )
    )
    signed_url = self.request_signer.generate_presigned_url(
        {"method": "GET", "url": url, "body": {}, "headers": {}, "context": {}},
        operation_name="connect",
        expires_in=900,
        region_name=self.region,
    )
    # RequestSigner only seems to work if the URL has a protocol, but
    # ElastiCache only accepts the URL without a protocol
    # So strip it off the signed URL before returning
    return (self.user, signed_url.removeprefix("https://"))

def lambda_handler(event, context):
    username = "iam-user-01" # replace with your user id
    cache_name = "cache-01" # replace with your cache name
    elasticache_endpoint = "cache-01-xxxxx.serverless.us-east-1.cache.amazonaws.com" #
    replace with your cache endpoint
    creds_provider = ElastiCacheIAMProvider(user=username, cache_name=cache_name,
    is_serverless=True)
    redis_client = redis.Redis(host=elasticache_endpoint, port=6379,
    credential_provider=creds_provider, ssl=True, ssl_cert_reqs="none")

    key='uuid'
    # create a random UUID - this will be the sample element we add to the cache
    uuid_in = uuid.uuid4().hex
    redis_client.set(key, uuid_in)
    result = redis_client.get(key)
    decoded_result = result.decode("utf-8")
    # check the retrieved item matches the item added to the cache and print

```



```
# the results
if decoded_result == uuid_in:
    print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from Valkey.")
else:
    raise Exception(f"Bad value retrieved. Expected {uuid_in}, got {decoded_result}")

return "Fetched value from Valkey"
```

Dieser Code verwendet die Python-Bibliothek redis-py, um Elemente in Ihren Cache zu legen und abzurufen. Dieser Code verwendet Cachetools, um generierte IAM-Auth-Token 15 Minuten lang zwischenspeichern. Gehen Sie wie folgt vor, um ein Bereitstellungspaket zu erstellen, das redis-py und cachetools enthält.

Erstellen Sie in Ihrem Projektverzeichnis, das die Quelldatei app.py enthält, ein Ordnerpaket, in dem Sie die Bibliotheken redis-py und cachetools installieren können.

```
mkdir package
```

Installieren Sie redis-py, cachetools mit pip.

```
pip install --target ./package redis
pip install --target ./package cachetools
```

Erstellen Sie eine ZIP-Datei, die die Bibliotheken redis-py und cachetools enthält. Führen Sie unter Linux oder macOS den folgenden Befehl aus: Verwenden Sie in Windows Ihr bevorzugtes ZIP-Hilfsprogramm, um eine .zip-Datei mit den Bibliotheken redis-py und cachetools im Stammverzeichnis zu erstellen.

```
cd package
zip -r ../my_deployment_package.zip .
```

Fügen Sie den Funktionscode in die ZIP-Datei ein. Führen Sie unter Linux oder macOS den folgenden Befehl aus: Verwenden Sie in Windows Ihr bevorzugtes ZIP-Programm, um app.py zum Stammverzeichnis Ihrer ZIP-Datei hinzuzufügen.

```
cd ..
zip my_deployment_package.zip app.py
```

Schritt 2.2: Eine IAM-Rolle erstellen (Ausführungsrolle)

Hängen Sie die AWS verwaltete Richtlinie mit dem Namen der Rolle `AWSLambdaVPCAccessExecutionRole` an.

```
aws iam attach-role-policy \  
  --role-name "elasticache-iam-auth-app" \  
  --policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"
```

Schritt 2.3: Aktualisieren des Bereitstellungspakets (Erstellen der Lambda-Funktion)

In diesem Schritt erstellen Sie die Lambda-Funktion (`AccessValkey`) mit dem Befehl `AWS CLI create-function`.

Führen Sie in dem Projektverzeichnis, das die ZIP-Datei Ihres Bereitstellungspakets enthält, den folgenden `create-function` Lambda-CLI-Befehl aus.

Verwenden Sie für die Rollenoption den ARN der Ausführungsrolle, die Sie im vorherigen Schritt erstellt haben. Geben Sie für die `vpc-config` durch Kommas getrennte Listen der Subnetze Ihrer Standard-VPC und die Sicherheitsgruppen-ID Ihrer Standard-VPC ein. Sie finden diese Werte in der Amazon-VPC-Konsole. Um die Subnetze Ihrer Standard-VPC zu finden, wählen Sie Ihre VPCs und dann die Standard-VPC Ihres AWS Kontos aus. Um die Sicherheitsgruppe für diese VPC zu finden, gehen Sie zu Sicherheit und wählen Sie Sicherheitsgruppen aus. Stellen Sie sicher, dass die Region `us-east-1` ausgewählt ist.

```
aws lambda create-function \  
  --function-name AccessValkey \  
  --region us-east-1 \  
  --zip-file fileb://my_deployment_package.zip \  
  --role arn:aws:iam::123456789012:role/elasticache-iam-auth-app \  
  --handler app.lambda_handler \  
  --runtime python3.12 \  
  --timeout 30 \  
  --vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-security-group-id
```

Schritt 3: Testen Sie die Lambda-Funktion mit ElastiCache

In diesem Schritt rufen Sie die Lambda-Funktion manuell mit dem Befehl `invoke` auf. Wenn die Lambda-Funktion ausgeführt wird, generiert sie eine UUID und schreibt sie in den ElastiCache

Cache, den Sie in Ihrem Lambda-Code angegeben haben. Die Lambda-Funktion ruft das Element dann aus dem Cache ab.

1. Rufen Sie die Lambda-Funktion (AccessValkey) mit dem Befehl AWS Lambda invoke auf.

```
aws lambda invoke \  
--function-name AccessValkey \  
--region us-east-1 \  
output.txt
```

2. Stellen Sie sicher, dass die Lambda-Funktion erfolgreich ausgeführt wurde:

- Überprüfen Sie die Datei "output.txt".
- Überprüfen Sie die Ergebnisse in CloudWatch Logs, indem Sie die CloudWatch Konsole öffnen und die Protokollgruppe für Ihre Funktion auswählen (/). aws/lambda/AccessValkey Die Ausgabe dieses Protokollstreams sollte ähnlich wie folgt aussehen:

```
Success: Inserted 826e70c5f4d2478c8c18027125a3e01e. Fetched  
826e70c5f4d2478c8c18027125a3e01e from Valkey.
```

- Überprüfen Sie die Ergebnisse in der AWS Lambda Konsole.

Schritt 4: Aufräumen (optional)

Gehen Sie zum Aufräumen wie folgt vor.

Schritt 4.1: Lambda-Funktion löschen

```
aws lambda delete-function \  
--function-name AccessValkey
```

Schritt 4.2: Löschen Sie den serverlosen Cache

Löschen Sie den Cache.

```
aws elasticache delete-serverless-cache \  
--serverless-cache-name cache-01
```

Benutzer und Benutzergruppen entfernen.

```
aws elasticache delete-user \  
  --user-id default-user-disabled  
  
aws elasticache delete-user \  
  --user-id iam-user-01  
  
aws elasticache delete-user-group \  
  --user-group-id iam-user-group-01
```

Schritt 4.3: Entfernen Sie die IAM-Rolle und -Richtlinien

```
aws iam detach-role-policy \  
  --role-name "elasticache-iam-auth-app" \  
  --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"  
  
aws iam detach-role-policy \  
  --role-name "elasticache-iam-auth-app" \  
  --policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"  
  
aws iam delete-role \  
  --role-name "elasticache-iam-auth-app"  
  
aws iam delete-policy \  
  --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

Einen knotenbasierten Cluster erstellen und verwalten

ElastiCache

Wenn Sie eine genaue Kontrolle über Ihren ElastiCache Cluster benötigen, können Sie einen knotenbasierten Cluster erstellen. ElastiCache ermöglicht Ihnen den Betrieb eines knotenbasierten Clusters, indem Sie den Knotentyp, die Anzahl der Knoten und die Platzierung der Knoten in den Availability Zones für Ihren Cluster auswählen. AWS Da es ElastiCache sich um einen vollständig verwalteten Service handelt, verwaltet er automatisch die Hardwarebereitstellung, die Überwachung, den Austausch von Knoten und das Software-Patching für Ihren Cluster.

Weitere Informationen zum Einrichten finden Sie unter [Einrichten ElastiCache](#). Details zur Verwaltung, Aktualisierung oder Löschung von Knoten oder Clustern finden Sie unter [Knoten verwalten in ElastiCache](#). Einen Überblick über die wichtigsten Komponenten einer ElastiCache Amazon-Bereitstellung bei der Erstellung eines knotenbasierten ElastiCache Clusters finden Sie in diesen [Schlüsselkonzepten](#).

Themen

- [ElastiCache Komponenten und Funktionen](#)
- [ElastiCache Terminologie](#)
- [Tutorial: So erstellen Sie einen knotenbasierten Cluster ElastiCache](#)
- [Löschen eines Clusters](#)
- [Weitere ElastiCache Tutorials und Videos](#)
- [Knoten verwalten in ElastiCache](#)
- [Verwaltung von Clustern in ElastiCache](#)
- [Vergleich knotenbasierter Valkey-, Memcached- und Redis OSS-Cluster](#)
- [Online-Migration für Valkey oder Redis OSS](#)
- [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#)

ElastiCache Komponenten und Funktionen

Im Folgenden finden Sie einen Überblick über die wichtigsten Komponenten einer ElastiCache Amazon-Bereitstellung.

Themen

- [ElastiCache Knoten](#)
- [ElastiCache Scherben](#)
- [ElastiCache Cluster](#)
- [ElastiCache Replikation](#)
- [ElastiCache Endpunkte](#)
- [ElastiCache Parametergruppen](#)
- [ElastiCache Sicherheit](#)
- [ElastiCache Subnetzgruppen](#)
- [ElastiCache Backups](#)
- [ElastiCache Ereignisse](#)

ElastiCache Knoten

Ein Knoten ist der kleinste Baustein einer ElastiCache Bereitstellung. Ein Knoten kann isoliert von anderen Knoten oder in einer bestimmten Beziehung zu anderen Knoten existieren.

Ein Knoten ist Teil eines sicheren, mit dem Netzwerk verbundenen RAMs mit fester Größe. Jeder Knoten führt eine Instance der Engine und der Version aus, die gewählt wurden, als Sie Ihren Cluster erstellt haben. Falls erforderlich, können Sie die Knoten in einem Cluster auf einen anderen Instance-Typ nach oben oder unten skalieren. Weitere Informationen finden Sie unter [Skalierung ElastiCache](#).

Jeder Knoten in einem Cluster ist vom selben Instance-Typ und führt dieselbe Cache-Engine aus. Jeder Cache-Knoten besitzt einen eigenen Domain Name Service (DNS)-Namen und Port. Mehrere Arten von Cache-Knoten werden unterstützt, jeder mit unterschiedlich viel zugewiesenem Arbeitsspeicher. Eine Liste der unterstützten Knoten-Instance-Typen finden Sie unter [Unterstützte Knotentypen](#).

Sie können Knoten auf einer pay-as-you-go Basis erwerben, bei der Sie nur für die Nutzung eines Knotens bezahlen. Oder Sie können reservierte Knotenpunkte zu einem deutlich reduzierten Stundensatz erwerben. Wenn Ihre Nutzungsrate hoch ist, können Sie durch den Kauf reservierter Knoten Geld sparen. Angenommen, Ihr Cluster wird fast immer verwendet, und Sie fügen gelegentlich Knoten hinzu, um Spitzen abzufangen. In diesem Fall können Sie eine Reihe von reservierten Knoten erwerben, die die meiste Zeit laufen.. Sie können dann pay-as-you-go Knoten für die Zeiten kaufen, in denen Sie gelegentlich Knoten hinzufügen müssen. Weitere Informationen zu reservierten Knoten finden Sie unter [Reservierte Knoten](#).

Weitere Informationen zu Knoten finden Sie unter [Knoten verwalten in ElastiCache](#).

ElastiCache Scherben

Ein Valkey- oder Redis-OSS-Shard (in der API und CLI als Knotengruppe bezeichnet) ist eine Gruppierung von ein bis sechs verwandten Knoten. Ein Valkey- oder Redis OSS-Cluster mit aktiviertem Clustermodus hat immer mindestens einen Shard.

Sharding ist eine Methode der Datenbankpartitionierung, die große Datenbanken in kleinere, schnellere und einfacher zu verwaltende Teile unterteilt, die als Daten-Shards bezeichnet werden. Dadurch kann die Datenbankeffizienz erhöht werden, indem Operationen auf mehrere separate Abschnitte verteilt werden. Die Verwendung von Shards kann viele Vorteile bieten, darunter eine verbesserte Leistung, Skalierbarkeit und Kosteneffizienz.

Valkey- und Redis OSS-Cluster mit aktiviertem Clustermodus können bis zu 500 Shards enthalten, wobei Ihre Daten auf die Shards verteilt sind. Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Valkey- oder Redis OSS-Engine-Version 5.0.6 oder höher ist. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#). Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, siehe [AWS-Service-Limits](#) und wählen Sie den Limittyp Knoten pro Cluster pro Instance-Typ.

Ein Shard mit mehreren Knoten implementiert die Replikation durch einen Primärknoten mit Lese-/Schreibzugriff und 1-5 Replikationsknoten. Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

Weitere Informationen zu Shards finden Sie unter [Arbeiten mit Shards in ElastiCache](#).

ElastiCache Cluster

Ein Cluster ist eine logische Gruppierung von einem oder mehreren [Knoten](#). Daten werden auf die Knoten in einem Memcache-Cluster und auf die Shards in einem Valkey- oder Redis OSS-Cluster, für den der Clustermodus aktiviert ist, partitioniert.

Viele Operationen sind ElastiCache auf Cluster ausgerichtet:

- Einen Cluster erstellen
- Modifizieren eines Clusters
- Erstellen von Snapshots eines Clusters (alle Redis-Versionen)
- Löschen eines Clusters
- Anzeigen der Elemente in einem Cluster
- Hinzufügen oder Entfernen von Kostenzuordnungs-Tags in einem Cluster

Detailliertere Informationen finden Sie in den folgenden verwandten Themen:

- [Verwaltung von Clustern in ElastiCache](#) und [Knoten verwalten in ElastiCache](#)

Informationen zu Clustern, Knoten und verwandten Operationen

- [AWS Servicebeschränkungen: Amazon ElastiCache](#)

Informationen zu ElastiCache Grenzwerten, wie z. B. der maximalen Anzahl von Knoten oder Clustern. Um bestimmte dieser Grenzwerte zu überschreiten, können Sie über das [Antragsformular für ElastiCache Amazon-Cache-Knoten eine Anfrage stellen](#).

- [Minimieren von Ausfällen](#)

Informationen zur Verbesserung der Fehlertoleranz Ihrer Cluster und Valkey- oder Redis OSS-Replikationsgruppen.

Typische Cluster-Konfigurationen

Folgende sind typische Clusterkonfigurationen.

Valkey- oder Redis OSS-Cluster

Valkey- oder Redis-OSS-Cluster mit deaktiviertem Clustermodus enthalten immer nur einen Shard (in der API und CLI eine Knotengruppe). Ein Valkey- oder Redis OSS-Shard enthält ein bis sechs Knoten. Wenn sich mehr als ein Knoten in einer Shard befindet, unterstützt die Shard die Replikation. In diesem Fall ist ein Knoten der read/write primäre Knoten und die anderen sind schreibgeschützte Replikatknoten.

Für eine verbesserte Fehlertoleranz empfehlen wir, mindestens zwei Knoten in einem Valkey- oder Redis-OSS-Cluster zu haben und Multi-AZ zu aktivieren. Weitere Informationen finden Sie unter [Minimieren von Ausfällen](#).

Wenn sich die Nachfrage nach Ihrem Valkey- oder Redis OSS-Cluster ändert, können Sie nach oben oder unten skalieren. Verschieben Sie dazu Ihren Cluster auf einen anderen Knoteninstanztyp. Wenn Ihre Anwendung leseintensiv ist, empfehlen wir, dem Cluster schreibgeschützte Replikate hinzuzufügen. So können Sie die Lesevorgänge auf eine angemessenere Anzahl von Knoten verteilen.

Sie können auch Daten-Tiering verwenden. Daten, auf die häufiger zugegriffen wird, werden im Speicher gespeichert, und Daten, auf die seltener zugegriffen wird, werden auf der Festplatte gespeichert. Der Vorteil der Verwendung von Daten-Tiering besteht darin, dass der Speicherbedarf verringert wird. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

ElastiCache unterstützt das dynamische Ändern des Knotentyps eines Valkey- oder Redis OSS-Clusters in einen größeren Knotentyp. Information zur Skalierung nach oben oder unten finden Sie unter [Skalierung für Valkey- oder Redis OSS-Cluster \(Clustermodus deaktiviert\)](#) oder [Skalierung von Replikatknoten für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#).

Typische Cluster-Konfigurationen für Memcached

Memcached unterstützt bis zu 300 Knoten pro Kunde für jede AWS Region, wobei jeder Cluster aus 1—60 Knoten besteht. In einem Memcached-Cluster können Sie die Daten über die Knoten partitionieren.

Wenn Sie die Memcached-Engine ausführen, können Cluster aus 1—60 Knoten bestehen. Sie partitionieren die Datenbank über die Knoten. Ihre Anwendung liest und schreibt auf den Endpunkt eines jeden Knotens. Weitere Informationen finden Sie unter [Auto Discovery](#).

Um die Fehlertoleranz zu verbessern, platzieren Sie Ihre Memcached-Knoten in verschiedenen Availability Zones (AZs) innerhalb der Clusterregion. AWS. Dadurch wirkt sich ein Fehler in einer AZ nur minimal auf den gesamten Cluster und die Anwendung aus. Weitere Informationen finden Sie unter [Minimieren von Ausfällen](#).

Wenn sich die Anforderungen an den Memcached-Cluster ändern, können Sie ihn durch Hinzufügen oder Entfernen von Knoten skalieren. Dabei werden die Daten entsprechend der neuen Anzahl an Knoten neu partitioniert. Zur Partitionierung der Daten empfehlen wir, konsistentes Hashing zu verwenden. Weitere Informationen zu konsistentem Hashing finden Sie unter [Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich \(Memcached\)](#).

ElastiCache Replikation

Für Valkey und Redis OSS wird die Replikation implementiert, indem zwei bis sechs Knoten in einem Shard gruppiert werden (in der API und CLI, als Knotengruppe bezeichnet). Einer dieser Knoten ist der primäre Knoten für Lese- und Schreibvorgänge. Alle anderen Knoten sind schreibgeschützte Replikationsknoten. Replikationen sind nur für Valkey und Redis OSS verfügbar und nicht ElastiCache für Memcached. ElastiCache

Jeder Replikationsknoten speichert eine Kopie der Daten vom primären Knoten. Replikationsknoten verwenden asynchrone Replikationsmechanismen, um die Synchronisierung mit dem primären Knoten aufrecht zu erhalten. Anwendungen können von jedem Knoten im Cluster lesen, aber nur auf primäre Knoten schreiben. Read Replicas erhöhen die Skalierbarkeit, indem sie Lesevorgänge über mehrere Endpunkte verteilen. Read Replicas verbessern auch die Fehlertoleranz, indem sie mehrere Kopien der Daten speichern. Durch die Verteilung von Read Replicas auf mehrere Availability Zones wird die Fehlertoleranz noch weiter verbessert. Weitere Informationen zur Fehlertoleranz finden Sie unter [Minimieren von Ausfällen](#).

Valkey- oder Redis-OSS-Cluster unterstützen einen Shard (in der API und CLI, als Knotengruppe bezeichnet).

Die Replikation aus Perspektive von API und CLI verwendet eine andere Terminologie, um die Kompatibilität mit früheren Versionen zu gewährleisten. Die Ergebnisse sind jedoch dieselben. Die folgende Tabelle zeigt die API- und CLI-Bedingungen für das Ausführen von Replikationen.

Vergleich der Replikation: Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert) --> Valkey oder Redis OSS-Cluster mit aktiviertem Clustermodus im Vergleich zu Valkey oder Redis OSS-Cluster mit deaktiviertem Clustermodus

In der folgenden Tabelle finden Sie einen Vergleich der Funktionen der Replikationsgruppen Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert).

	Valkey- oder Redis OSS-Cluster mit deaktiviertem Clustermodus	Valkey- oder Redis OSS-Cluster mit aktiviertem Clustermodus
Shards (Knotengruppen)	1	1–500
Replikate pro Shard (Knotengruppe)	0–5	0–5

	Valkey- oder Redis OSS-Cluster mit deaktiviertem Clustermodus	Valkey- oder Redis OSS-Cluster mit aktiviertem Clustermodus
Datenpartitionierung	Nein	Ja
Hinzufügen/Löschen von Replikaten	Ja	Ja
Hinzufügen/Löschen von Knotengruppen	Nein	Ja
Unterstützung einer Erweiterung	Ja	Ja
Unterstützung von Engine-Updates	Ja	Ja
Hochstufen eines Replikats auf Primär	Ja	Automatisch
Multi-AZ	Optional	Erforderlich
Sicherung/Wiederherstellung	Ja	Ja

Hinweise:

Wenn sich auf einem Primärsystem keine Replikate befinden, verlieren Sie bei einem Ausfall des Primärsystems alle darauf befindlichen Daten.

Sie können Backup und Restore verwenden, um zu Valkey oder Redis OSS zu migrieren (Clustermodus aktiviert).

Sie können Backup und Restore verwenden, um die Größe Ihres Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) zu ändern.

Sämtliche Shards (in der API und CLI, Knotengruppen) und Knoten müssen sich in derselben AWS-Region befinden. Sie können die einzelnen Knoten jedoch in mehreren Availability Zones innerhalb dieser Region bereitstellen.

Lesereplikate schützen vor möglichem Datenverlust, da Ihre Daten über zwei oder mehr Knoten repliziert werden – den primären und einen oder mehrere Lesereplikate. Für eine höhere Zuverlässigkeit und schnellere Wiederherstellung empfehlen wir, ein oder mehrere Lesereplikate in verschiedenen Availability Zones zu erstellen.

Sie können auch globale Datenspeicher nutzen. Mithilfe der OSS-Funktion Global Datastore for Redis können Sie mit einer vollständig verwalteten, schnellen, zuverlässigen und sicheren Replikation über Regionen hinweg arbeiten. AWS Mit dieser Funktion können Sie regionsübergreifende Read Replica-Cluster erstellen, um Lesevorgänge mit niedriger Latenz und regionsübergreifende Disaster Recovery ElastiCache zu ermöglichen. AWS Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe](#) globaler Datenspeicher.

Replikation: Einschränkungen und Ausnahmen

- Multi-AZ wird für den Knotentypen T1 nicht unterstützt.

ElastiCache Endpunkte

Ein Endpunkt ist die eindeutige Adresse, die Ihre Anwendung verwendet, um eine Verbindung zu einem ElastiCache Knoten oder Cluster herzustellen.

Einzelknoten-Endpunkte für Valkey oder Redis OSS mit deaktiviertem Clustermodus

Der Endpunkt für einen Valkey- oder Redis OSS-Cluster mit einem Knoten wird verwendet, um eine Verbindung zum Cluster sowohl für Lese- als auch für Schreibvorgänge herzustellen.

Endpunkte mit mehreren Knoten für Valkey oder Redis OSS mit deaktiviertem Clustermodus

Ein Valkey- oder Redis OSS-Cluster mit mehreren Knoten und deaktiviertem Clustermodus hat zwei Arten von Endpunkten. Über den primären Endpunkt wird immer die Verbindung zum primären Knoten im Cluster hergestellt, auch wenn sich die primäre Rolle des Knotens ändert. Verwenden Sie den primären Endpunkt für alle Schreibvorgänge im Cluster.

Verwenden Sie Leser-Endpunkt, um am Endpunkt ankommende Verbindungen auf alle Lesereplikate zu verteilen. Verwenden Sie die einzelnen Node-Endpoints für Lesevorgänge (im Folgenden werden API/CLI diese als Lese-Endpunkte bezeichnet).

Valkey- oder Redis OSS-Endpunkte (Cluster Mode Enabled)

Ein Valkey- oder Redis OSS-Cluster mit aktiviertem Clustermodus hat einen einzigen Konfigurationsendpunkt. Über die Verbindung zum Konfigurationsendpunkt findet die Anwendung für jede Shard im Cluster die primären Endpunkte sowie jene für Schreibvorgänge.

Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).

ElastiCache für Memcached-Endpunkte

Jeder Knoten in einem Memcached-Cluster hat einen eigenen Endpunkt. Der Cluster verfügt auch über einen sogenannten Konfigurationsendpunkt. Wenn Sie Auto Discovery aktivieren und eine Verbindung zum Konfigurationsendpunkt herstellen, kennt die Anwendung jeden Knotenendpunkt, auch wenn danach Knoten im Cluster hinzugefügt oder entfernt wurden. Weitere Informationen finden Sie unter [Auto Discovery](#).

Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).

ElastiCache Parametergruppen

Cache-Parametergruppen vereinfachen die Verwaltung von Laufzeiteinstellungen für unterstützte Engine-Software. Parameter werden zur Steuerung von Speichernutzung, Bereinigungsrichtlinien, Elementgrößen usw. verwendet. Eine ElastiCache Parametergruppe ist eine benannte Sammlung von Engine-spezifischen Parametern, die Sie auf einen Cluster anwenden können. Dadurch stellen Sie sicher, dass alle Knoten in diesem Cluster identisch konfiguriert werden.

Eine Liste der unterstützten Parameter, der Standardwerte und Parameter, die geändert werden können, finden Sie unter [DescribeEngineDefaultParameters](#) (CLI: [describe-engine-default-parameters](#)).

Ausführlichere Informationen zu ElastiCache Parametergruppen finden Sie unter [Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen](#).

ElastiCache Sicherheit

Aus Sicherheitsgründen ist der ElastiCache Knotenzugriff auf Anwendungen beschränkt, die auf den von Ihnen zugelassenen EC2 Amazon-Instances ausgeführt werden. Sie können die EC2 Amazon-Instances, die auf Ihren Cluster zugreifen können, mithilfe von Sicherheitsgruppen steuern.

Standardmäßig werden alle neuen ElastiCache Cluster in einer Amazon Virtual Private Cloud (Amazon VPC) -Umgebung gestartet. Sie können Subnetzgruppen verwenden, um Cluster-Zugriff von EC2 Amazon-Instances aus zu gewähren, die in bestimmten Subnetzen ausgeführt werden.

Unterstützt nicht nur den Knotenzugriff, sondern ElastiCache unterstützt auch TLS und In-Place-Verschlüsselung für Knoten, auf denen bestimmte Versionen von ausgeführt werden. ElastiCache Weitere Informationen finden Sie hier:

- [Datensicherheit bei Amazon ElastiCache](#)
- [Authentifizierung mit dem Befehl Valkey und Redis OSS AUTH](#)

ElastiCache Subnetzgruppen

Eine Subnetzgruppe ist eine Sammlung von Subnetzen (in der Regel privat), die Sie für Ihre Cluster in einer Amazon-VPC-Umgebung festlegen können.

Wenn Sie einen Cluster in einer Amazon VPC erstellen, müssen Sie eine Subnetzgruppe angeben. ElastiCache verwendet diese Cache-Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen und diese mit Ihren Cache-Knoten zu verknüpfen.

Weitere Informationen zur Verwendung von Cache-Subnetzgruppen in einer Amazon-VPC-Umgebung finden Sie im Folgenden:

- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#)
- [Subnetze und Subnetzgruppen](#)

ElastiCache Backups

Ein Backup ist eine point-in-time Kopie eines Valkey- oder Redis OSS-Clusters oder eines serverlosen Caches oder eines serverlosen Memcached-Caches. Backups können zur Wiederherstellung eines bestehenden Clusters oder zum Seeding eines neuen Clusters verwendet werden. Sicherungen umfassen alle Daten in einem Cluster sowie zusätzlich einige Metadaten.

Abhängig von der Version von Valkey oder Redis OSS, die auf Ihrem Cluster ausgeführt wird, benötigt der Backup-Prozess unterschiedliche Mengen an reserviertem Speicher, um erfolgreich zu sein. Weitere Informationen finden Sie hier:

- [Snapshot und Wiederherstellung](#)
- [So werden Synchronisation und Backup implementiert](#)
- [Auswirkungen von Backups knotenbasierter Cluster auf die Leistung](#)
- [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#)

ElastiCache Ereignisse

ElastiCache Sendet bei wichtigen Ereignissen in einem Cluster eine Benachrichtigung an ein bestimmtes Amazon SNS SNS-Thema. Zu diesen Ereignissen gehören z. B. das fehlgeschlagene oder erfolgreiche Hinzufügen eines Knotens, die Änderung einer Sicherheitsgruppe usw. Durch die Überwachung von Schlüsselereignissen können Sie den aktuellen Zustand Ihrer Cluster erkennen und in vielen Fällen Korrekturmaßnahmen ergreifen.

Weitere Informationen zu ElastiCache Ereignissen finden Sie unter [Amazon SNS SNS-Überwachung von Ereignissen ElastiCache](#).

ElastiCache Terminologie

Im Oktober 2016 ElastiCache hat Amazon die Unterstützung für Redis OSS 3.2 eingeführt. Zu diesem Zeitpunkt haben wir Unterstützung für die Partitionierung Ihrer Daten auf bis zu 500 Shards (in der ElastiCache API als Knotengruppen bezeichnet) hinzugefügt. AWS CLI Um die Kompatibilität mit früheren Versionen zu gewährleisten, haben wir den Betrieb der API-Version 2015-02-02 um die neue Redis OSS-Funktionalität erweitert.

Gleichzeitig haben wir begonnen, in der ElastiCache Konsole eine Terminologie zu verwenden, die in dieser neuen Funktion verwendet wird und in der gesamten Branche üblich ist. Durch diese Änderung kann es stellenweise dazu kommen, dass sich die in der API und der CLI verwendete Terminologie von der in der Konsole verwendeten unterscheidet. Die folgende Liste enthält Begriffe, die sich zwischen API, CLI und der Konsole unterscheiden können.

Cache-Cluster oder Knoten vs. Knoten

Es besteht eine one-to-one Beziehung zwischen einem Knoten und einem Cluster, wenn es keine Replikatknoten gibt. Daher wurden die Begriffe in der ElastiCache Konsole häufig synonym verwendet. Die Konsole verwendet jetzt durchgängig den Begriff Knoten. Einzige Ausnahme ist die Schaltfläche Create Cluster, über die der Prozess zur Erstellung eines Clusters mit oder ohne Replikationsknoten gestartet wird.

Die ElastiCache API und AWS CLI ich verwenden die Begriffe weiterhin wie in der Vergangenheit.

Cluster im Vergleich zu Valkey oder Redis OSS-Replikationsgruppe

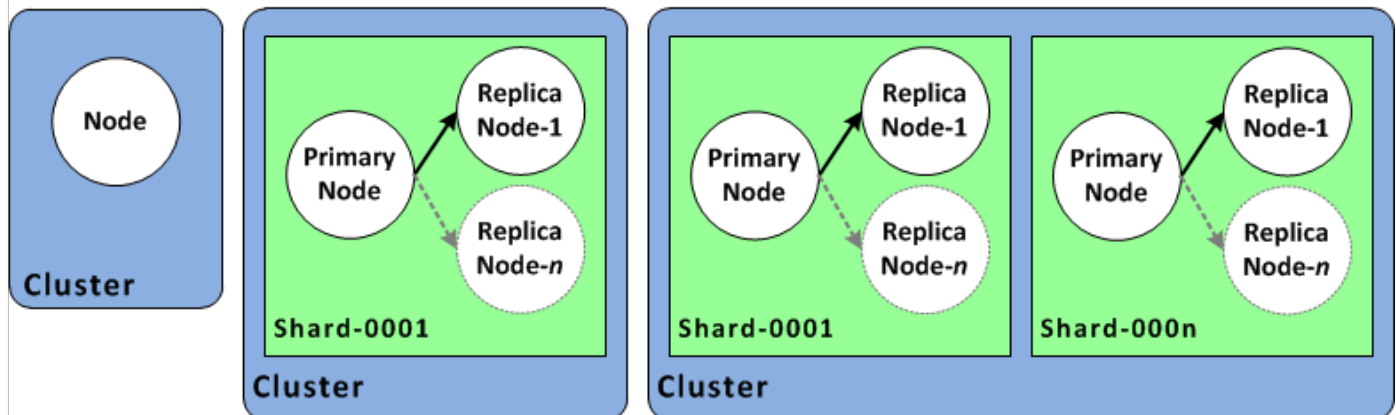
Die Konsole verwendet jetzt den Begriff Cluster for all ElastiCache für Redis OSS-Cluster. Die Konsole verwendet den Begriff "Cluster" in den folgenden Fällen:

- Wenn es sich bei dem Cluster um einen Valkey- oder Redis OSS-Cluster mit einem Knoten handelt.
- Wenn es sich bei dem Cluster um einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) handelt, der die Replikation innerhalb eines einzelnen Shards unterstützt (in der API und CLI, als Knotengruppe bezeichnet).
- Wenn es sich bei dem Cluster um einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) handelt, der die Replikation innerhalb von 1—90 Shards oder bis zu 500 Shards mit einer Anforderung zur Erhöhung des Limits unterstützt. Um eine Erhöhung des Limits zu beantragen, siehe [AWS-Service-Limits](#) nach und wählen Sie den Limittyp Knoten pro Cluster pro Instance-Typ.

Weitere Informationen zu Valkey- oder Redis OSS-Replikationsgruppen finden Sie unter. [Hohe Verfügbarkeit mit Replikationsgruppen](#)

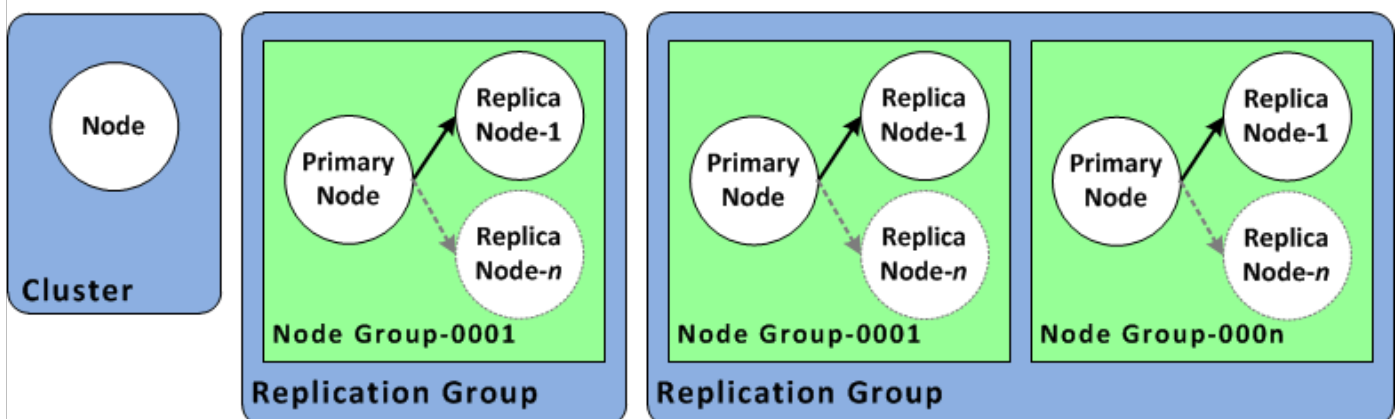
Das folgende Diagramm veranschaulicht die verschiedenen Topologien von ElastiCache Redis OSS-Clustern aus Sicht der Konsole.

ElastiCache (Redis OSS): Console View



Die ElastiCache API und die AWS CLI Operationen unterscheiden immer noch zwischen einzelnen Knoten ElastiCache für Redis OSS-Cluster und Valkey- oder Redis OSS-Replikationsgruppen mit mehreren Knoten. Das folgende Diagramm veranschaulicht die verschiedenen OSS-Topologien ElastiCache für Redis aus der API und der Perspektive. ElastiCache AWS CLI

ElastiCache (Redis OSS): API/CLI View



Valkey- oder Redis OSS-Replikationsgruppe im Vergleich zu globalem Datenspeicher

Ein globaler Datenspeicher ist eine Sammlung von einem oder mehreren Clustern, die sich regionsübergreifend aufeinander replizieren, wohingegen eine Valkey- oder Redis-OSS-

Replikationsgruppe Daten über einen Cluster mit aktiviertem Clustermodus mit mehreren Shards repliziert. Ein globaler Datenspeicher besteht aus folgenden Komponenten:

- Primärer (aktiver) Cluster – Ein primärer Cluster nimmt Schreibvorgänge entgegen, die auf alle Cluster innerhalb des globalen Datenspeichers repliziert werden. Ein primärer Cluster akzeptiert auch Leseanfragen.
- Sekundärer (passiver) Cluster – Ein sekundärer Cluster nimmt nur Leseanforderungen entgegen und repliziert Datenaktualisierungen von einem primären Cluster. Ein sekundärer Cluster muss sich in einer anderen AWS Region befinden als der primäre Cluster.

Informationen zu globalen Datenspeichern finden Sie unter [AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#).

Tutorial: So erstellen Sie einen knotenbasierten Cluster ElastiCache

So erstellen Sie einen knotenbasierten ElastiCache Cluster für Valkey, Memcached und Redis OSS.

Themen

- [Einen knotenbasierten ElastiCache Cluster für Valkey erstellen](#)
- [Einen knotenbasierten ElastiCache Cluster für Redis OSS erstellen](#)

Einen knotenbasierten ElastiCache Cluster für Valkey erstellen

Im Folgenden sind die einmaligen Aktionen aufgeführt, die Sie ergreifen müssen, um einen knotenbasierten Cluster für Valkey zu erstellen. ElastiCache

Schritt 1: Erstellung einer Subnetzgruppe

Bevor Sie einen ElastiCache (Valkey-) Cluster erstellen, erstellen Sie zunächst eine Subnetzgruppe. Eine Cache-Subnetzgruppe ist eine Sammlung von Subnetzen, die Sie möglicherweise für Ihre Cluster in einer VPC festlegen möchten. Wenn Sie einen Cluster in einer VPC starten, müssen Sie eine Cache-Subnetzgruppe auswählen. ElastiCache verwendet dann diese Cache-Subnetzgruppe, um jedem Cache-Knoten im Cluster IP-Adressen innerhalb dieses Subnetzes zuzuweisen.

Wenn Sie eine neue Subnetzgruppe erstellen, notieren Sie sich die Anzahl der verfügbaren IP-Adressen. Wenn das Subnetz nur über sehr wenige freie IP-Adressen verfügt, sind Sie

möglicherweise hinsichtlich der Anzahl der Knoten, die Sie dem Cluster hinzufügen können, eingeschränkt. Um dieses Problem zu beheben, können Sie einer Subnetzgruppe ein oder mehrere Subnetze zuweisen, sodass Sie über eine ausreichende Anzahl von IP-Adressen in der Availability Zone Ihres Clusters verfügen. Danach können Sie dem Cluster weitere Knoten hinzufügen.

Weitere Informationen zur Einrichtung ElastiCache finden Sie unter [Einrichten ElastiCache](#)

Die folgenden Verfahren zeigen Ihnen, wie Sie eine Subnetzgruppe mit dem Namen mysubnetgroup (Konsole) und die AWS CLI erstellen.

Erstellen einer Subnetzgruppe (Konsole)

Im folgenden Verfahren wird das Erstellen einer Subnetzgruppe (Konsole) erläutert.

Erstellen einer DB-Sicherheitsgruppe (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie in der Navigationsliste auf Subnet Groups.
3. Klicken Sie auf Create Subnet Group (Subnetzgruppe ändern).
4. Gehen Sie im Assistenten Create Subnet Group wie folgt vor. Wenn Sie alle gewünschten Einstellungen vorgenommen haben, klicken Sie auf Yes, Create.
 - a. Geben Sie im Feld Name einen Namen für Ihre Subnetzgruppe ein.
 - b. Geben Sie im Feld Description eine Beschreibung für Ihre Subnetzgruppe ein.
 - c. Wählen Sie im Feld VPC ID die erstellte Amazon VPC aus.
 - d. Wählen Sie in den Listen Availability Zone und Subnet ID die Availability Zone oder [Verwenden von lokalen Zonen mit ElastiCache](#) die ID Ihres privaten Subnetzes aus und klicken Sie dann auf Hinzufügen.

Subnet group settings

A subnet group is a collection of subnets (typically private). Designate a subnet group for your clusters running in an Amazon Virtual Private Cloud (VPC) environment.

Name

The name is required, can have up to 255 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Description - optional

VPC ID

The identifier for the VPC environment where your cluster is to run.

 ▼ [Create VPC](#)

For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.

Selected subnets (6) [Manage](#)

Availability Zone ▲	Subnet ID ▼	Outpost ID ▼	CIDR block ▼
us-east-1a	subnet-		172.31.16.0/20
us-east-1b	subnet-		172.31.32.0/20
us-east-1c	subnet-		172.31.0.0/20
us-east-1d	subnet-		172.31.80.0/20

5. Klicken Sie in der angezeigten Bestätigungsmeldung auf Close.

Ihre neue Subnetzgruppe wird in der Liste der Subnetzgruppen der Konsole angezeigt. ElastiCache Unten im Fenster können Sie die Subnetzgruppe auswählen, um Details wie die der Gruppe zugeordneten Subnetze anzuzeigen.

Erstellen einer Subnetzgruppe (AWS CLI)

Geben Sie in einem Befehlszeilenfenster den Befehl `create-cache-subnet-group` ein, um eine Subnetzgruppe zu erstellen.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-subnet-group \
  --cache-subnet-group-name mysubnetgroup \
  --cache-subnet-group-description "Testing" \
```

```
--subnet-ids subnet-53df9c3a
```

Für Windows:

```
aws elasticache create-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup ^  
  --cache-subnet-group-description "Testing" ^  
  --subnet-ids subnet-53df9c3a
```

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

```
{  
  "CacheSubnetGroup": {  
    "VpcId": "vpc-37c3cd17",  
    "CacheSubnetGroupDescription": "Testing",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-53df9c3a",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        }  
      }  
    ],  
    "CacheSubnetGroupName": "mysubnetgroup"  
  }  
}
```

Weitere Informationen finden Sie im AWS CLI Thema. [create-cache-subnet-group](#)

Schritt 2: Erstellen eines Clusters

Bevor Sie einen Cluster für die Produktion erstellen, müssen Sie natürlich überlegen, wie Sie den Cluster entsprechend Ihren geschäftlichen Anforderungen konfigurieren. Diese Probleme werden im [Einen Cluster vorbereiten in ElastiCache](#)-Abschnitt adressiert. Für die Zwecke dieser Übung „Erste Schritte“ erstellen Sie einen Cluster mit deaktiviertem Cluster-Modus. Sie können die Standardkonfigurationswerte dort akzeptieren, wo sie zutreffen.

Der Cluster, den Sie gleich starten werden, wird live sein und nicht in einer Sandbox ausgeführt. Es fallen die ElastiCache Standardnutzungsgebühren für die Instance an, bis Sie sie löschen. Die Gesamtkosten werden minimal sein (meistens geringer als ein Dollar), wenn Sie diese Übung in einer

Sitzung durchlaufen und den Cluster löschen, sobald Sie die Übung abgeschlossen haben. Weitere Informationen zu den ElastiCache Nutzungsraten finden Sie auf [Amazon ElastiCache](https://aws.amazon.com/elasticache/).

Ihr Cluster wird, basierend auf dem Amazon-VPC-Service, in einer Virtual Private Cloud (VPC) gestartet.

Erstellen eines Valkey-Clusters (Cluster-Modus deaktiviert) (Konsole)

So erstellen Sie mit der Konsole einen Valkey-Cluster (Clustermodus deaktiviert) ElastiCache

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der Sie diesen Cluster starten möchten.
3. Wählen Sie im Navigationsbereich Get started (Erste Schritte) aus.
4. Wählen Sie Create VPC (VPC erstellen) aus und befolgen Sie die Schritte unter [Erstellen einer Virtual Private Cloud \(VPC\)](#).
5. Wählen Sie auf der ElastiCache Dashboard-Seite Valkey-Cache oder Redis-OSS-Cache und dann Valkey-Cache erstellen oder Redis OSS-Cache erstellen aus.
6. Führen Sie unter Cluster settings (Cluster-Einstellungen) die folgenden Schritte aus:
 - a. Wählen Sie Configure and create a new cluster (Neuen Cluster konfigurieren und erstellen) aus.
 - b. Wählen Sie für Cluster mode (Cluster-Modus) Disabled (Deaktiviert) aus.
 - c. Geben Sie für Cluster info (Cluster-Info) einen Wert für Name (Name) ein.
 - d. (Optional) Geben Sie einen Wert für Description (Beschreibung) ein.
7. Unter Location (Speicherort):

AWS Cloud

1. Wir empfehlen für AWS Cloud die Voreinstellungen für Multi-AZ und Auto-failover (Automatisches Failover) zu akzeptieren. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache](#) Redis OSS mit Multi-AZ.
2. Unter Cluster settings (Cluster-Einstellungen):
 - a. Wählen Sie unter Engine version (Engine-Version) eine verfügbare Version aus.

- b. Verwenden Sie für Port den Standardport 6379. Wenn es einen Grund gibt, einen anderen Port zu verwenden, geben Sie die betreffende Portnummer ein.
- c. Wählen Sie für die Parameter group (Parametergruppe) eine Parametergruppe aus oder erstellen Sie eine neue Parametergruppe. Parametergruppen steuern die Laufzeitparameter Ihres Clusters. Weitere Informationen zu Parametergruppen finden Sie unter [Valkey- und Redis OSS-Parameter](#) und [Eine ElastiCache Parametergruppe erstellen](#).

 Note

Wenn Sie eine Parametergruppe zum Festlegen der Engine-Konfigurationswerte auswählen, wird diese Parametergruppe auf alle Cluster im globalen Datenspeicher angewendet. Auf der Seite Parameter Groups (Parametergruppen) gibt das Ja/nein-Attribut Global an, ob eine Parteigruppe Teil eines globalen Datenspeichers ist.

- d. Klicken Sie für Node type (Knotentyp) auf den Abwärtspfeil (▼)). Wählen Sie im Dialogfeld Change node type (Knotentyp ändern) einen Wert für Instance family (Instance-Familie) für den gewünschten Knotentyp aus. Wählen Sie dann den Knotentyp aus, den Sie für diesen Cluster verwenden möchten, und wählen Sie dann Save (Speichern).

Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Wenn Sie einen r6gd-Knotentyp wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

- e. Wählen Sie unter Number of replicas (Anzahl der Repliken) die für diesen Cluster gewünschte Anzahl von Read Replicas aus. Wenn Multi-AZ aktiviert ist, muss die Zahl zwischen 1–5 liegen.
3. Unter Connectivity (Konnektivität)
- a. Wählen Sie als Network type (Netzwerktyp) die IP-Version(en) aus, die dieser Cluster unterstützen soll.
 - b. Wählen Sie für Subnetzgruppen das Subnetz aus, das Sie auf diesen Cluster anwenden möchten. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten

zugeordnet werden sollen. ElastiCache Cluster benötigen ein Dual-Stack-Subnetz mit beiden IPv4 und ihnen zugewiesenen IPv6 Adressen, um im Dual-Stack-Modus betrieben werden zu können, und ein Subnetz nur für den Betrieb als IPv6 -only. IPv6

Geben Sie beim Erstellen einer neuen Subnetzgruppe die VPC ID (VPC-ID) ein, zu der sie gehört.

Weitere Informationen finden Sie unter:

- [Auswahl eines Netzwerktyps in ElastiCache.](#)
- [Erstellen eines Subnetzes in der VPC](#)

Wenn Sie [Verwenden von lokalen Zonen mit ElastiCache](#) sind, müssen Sie ein Subnetz erstellen oder auswählen, das in der lokalen Zone liegt.

Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).

4. Für Availability zone placements (Availability-Zone-Platzierungen) haben Sie zwei Optionen:

- Keine Präferenz — wählt die Availability Zone. ElastiCache
- Availability Zones angeben – Sie geben die Availability Zone für jeden Cluster an.

Wenn Sie die Availability Zones angeben, wählen Sie für jeden Cluster in jedem Shard die Availability Zone aus der Liste aus.

Weitere Informationen finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

5. Wählen Sie Next (Weiter).
6. Unter Advanced Valkey oder Redis OSS-Einstellungen

- Für Security (Sicherheit):
 - i. Zur Verschlüsselung Ihrer Daten haben Sie die folgenden Optionen:
 - Verschlüsselung im Ruhezustand – Ermöglicht die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

 Note

Sie haben die Möglichkeit, einen anderen Verschlüsselungsschlüssel anzugeben, indem Sie Customer Managed AWS KMS Key und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwendung von kundenverwalteten Schlüsseln aus AWS-KMS](#).

- Verschlüsselung während der Übertragung – Ermöglicht die Verschlüsselung von Daten während der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey und für Redis OSS 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:
 - Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
 - Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern aus, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
 - AUTH Default User — Ein Authentifizierungsmechanismus für den Redis OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).
- AUTH — Ein Authentifizierungsmechanismus für den Redis OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).

 Note

Für Valkey und für Redis OSS-Versionen zwischen 3.2.6 und höher, mit Ausnahme von Version 3.2.10, ist Redis OSS AUTH die einzige Option.

- ii. Wählen Sie für Security groups (Sicherheitsgruppen) die gewünschten Sicherheitsgruppen für diesen Cluster aus. Eine security group (Sicherheitsgruppe) fungiert als Firewall, um den Netzwerkzugriff auf Ihren Cluster zu steuern. Sie können die Standardsicherheitsgruppe für Ihre VPC verwenden oder eine neue erstellen.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Sicherheitsgruppen für Ihre VPC](#) im Benutzerhandbuch zu Amazon VPC.

7. Wenn Sie regelmäßig geplante automatische Sicherungen möchten, aktivieren Sie Enable automatic backups und geben Sie ein, wie viele Tage lang jede automatische Sicherung beibehalten werden soll, bevor sie automatisch gelöscht wird. Wenn Sie keine regelmäßig geplanten automatischen Sicherungen möchten, deaktivieren Sie das Kontrollkästchen Enable automatic backups. In beiden Fällen haben Sie jederzeit die Option, manuelle Sicherungen zu erstellen.

Weitere Informationen zur Sicherung und Wiederherstellung von Redis OSS finden Sie unter [Snapshot und Wiederherstellung](#)

8. (Optional) Geben Sie ein Wartungsfenster an. Das Maintenance window (Wartungsfenster) ist der (in der Regel eine Stunde lange) allwöchentliche Zeitraum, für den ElastiCache die Systemwartung Ihres Clusters plant. Sie können es ElastiCache überlassen, einen Tag und eine Uhrzeit für das Wartungsfenster auszuwählen (No preference (Keine Präferenz)) oder Sie können den Tag, die Uhrzeit und die Dauer selbst wählen (Specify maintenance window (Wartungsfenster angeben)). Treffen Sie bei Wahl von Specify maintenance window eine Auswahl in den Listen Start day, Start time und Duration (in Stunden) für Ihr Wartungsfenster. Alle Zeiten sind UCT-Zeiten.

Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

9. (Optional) Für Protokolle:
 - Unter Protokollformat wählen Sie entweder Text oder JSON aus.
 - Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen aus und geben Sie entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und wählen Sie dann entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Streamnamen aus.
10. Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen unter Tags zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).
11. Wählen Sie Weiter aus.

12. Überprüfen Sie alle Ihre Einträge und ausgewählten Optionen und machen Sie dann evtl. erforderliche Korrekturen. Sobald Sie bereit sind, klicken Sie auf Create (Erstellen).

On premises

1. Für On premises (On-Premises) empfehlen wir Ihnen, Auto-failover (Automatisches Failover) aktiviert zu lassen. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache für Redis OSS](#) mit Multi-AZ
2. Um die Erstellung des Clusters abzuschließen, befolgen Sie die Schritte unter [Verwenden von Outposts](#).

Sobald der Status Ihres Clusters verfügbar ist, können Sie Amazon EC2 Zugriff darauf gewähren, eine Verbindung herstellen und mit der Nutzung beginnen. Weitere Informationen erhalten Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Informationen zum Arbeiten mit aktiviertem Cluster-Modus finden Sie in den folgenden Themen:

- Wenn Sie die Konsole verwenden möchten, lesen Sie [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#).
- Informationen zur Verwendung von AWS CLI finden Sie unter [Erstellen eines Valkey- oder Redis OSS-Clusters \(Clustermodus aktiviert\) \(AWS CLI\)](#).

Schritt 3. Autorisieren Sie den Zugriff auf den Cluster

In diesem Abschnitt wird davon ausgegangen, dass Sie mit dem Starten und Herstellen einer Verbindung zu EC2 Amazon-Instances vertraut sind. Weitere Informationen finden Sie im [Amazon-Leitfaden „EC2 Erste Schritte“](#).

Alle ElastiCache Cluster sind so konzipiert, dass sie von einer EC2 Amazon-Instance aus aufgerufen werden können. Das gängigste Szenario ist der Zugriff auf einen ElastiCache Cluster von einer EC2 Amazon-Instance in derselben Amazon Virtual Private Cloud (Amazon VPC), was bei dieser Übung der Fall sein wird.

Standardmäßig ist der Netzwerkzugriff auf Ihren Cluster auf das Konto beschränkt, über das er erstellt wurde. Bevor Sie von einer EC2 Instance aus eine Verbindung zu einem Cluster herstellen können, müssen Sie die EC2 Instance für den Zugriff auf den Cluster autorisieren.

Der häufigste Anwendungsfall ist, wenn eine auf einer EC2 Instance bereitgestellte Anwendung eine Verbindung zu einem Cluster in derselben VPC herstellen muss. Der einfachste Weg, den Zugriff zwischen EC2 Instances und Clustern in derselben VPC zu verwalten, ist wie folgt:

1. Erstellen Sie eine VPC-Sicherheitsgruppe für Ihren Cluster. Diese Sicherheitsgruppe kann verwendet werden, um den Zugriff auf die Cluster-Instances zu beschränken. Sie können für diese Sicherheitsgruppe beispielsweise eine benutzerdefinierte Regel erstellen, die TCP-Zugriff über den Port, den Sie dem Cluster bei seiner Erstellung zugeordnet haben, und eine IP-Adresse gewährt, mit der Sie auf den Cluster zugreifen.

Der Standardport für Valkey- oder Redis OSS-Cluster und Replikationsgruppen ist 6379

Important

ElastiCache Amazon-Sicherheitsgruppen gelten nur für Cluster, die nicht in einer Amazon Virtual Private Cloud-Umgebung (VPC) ausgeführt werden. Wenn Sie ein Cluster in einer Amazon Virtual Private Cloud betreiben, ist die Option Sicherheitsgruppen im Navigationsbereich der Konsole nicht verfügbar.

Wenn Sie Ihre ElastiCache Knoten in einer Amazon VPC ausführen, kontrollieren Sie den Zugriff auf Ihre Cluster mit Amazon VPC-Sicherheitsgruppen, die sich von ElastiCache Sicherheitsgruppen unterscheiden. Weitere Informationen zur Verwendung ElastiCache in einer Amazon VPC finden Sie unter [Amazon VPCs und ElastiCache Sicherheit](#)

2. Erstellen Sie eine VPC-Sicherheitsgruppe für Ihre EC2 Instances (Web- und Anwendungsserver). Diese Sicherheitsgruppe kann bei Bedarf den Zugriff auf die EC2 Instance aus dem Internet über die Routingtabelle der VPC ermöglichen. Sie können beispielsweise Regeln für diese Sicherheitsgruppe festlegen, um den TCP-Zugriff auf die EC2 Instance über Port 22 zu ermöglichen.
3. Erstellen Sie in der Sicherheitsgruppe für Ihren Cluster benutzerdefinierte Regeln, die Verbindungen von der Sicherheitsgruppe aus zulassen, die Sie für Ihre EC2 Instances erstellt haben. Damit wird jedem Mitglied der Sicherheitsgruppe der Zugriff auf die DB-Instances gestattet.

 Note

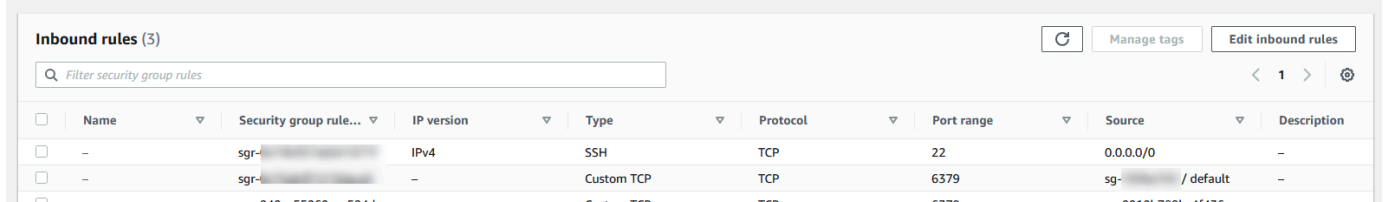
Wenn Sie planen, [Local Zones](#) zu nutzen, stellen Sie sicher, dass Sie sie aktiviert haben. Wenn Sie eine Subnetzgruppe in dieser Local Zone erstellen, wird Ihre VPC auf diese Local Zone erweitert, und Ihre VPC behandelt das Subnetz wie jedes Subnetz in einer anderen Availability Zone. Alle relevanten Gateways und Routing-Tabellen werden automatisch angepasst.

So erstellen Sie eine Regel in einer VPC-Sicherheitsgruppe, die Verbindungen über eine andere Sicherheitsgruppe zulässt

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc>.
2. Wählen Sie im Navigationsbereich Security Groups (Sicherheitsgruppen) aus.
3. Wählen Sie eine Sicherheitsgruppe aus oder erstellen Sie eine, die Sie für Ihre Cluster-Instances verwenden werden. Wählen Sie unter Inbound Rules (Eingangsregeln) die Option Edit Inbound Rules (Eingangsregeln bearbeiten) und dann Add Rule (Regeln hinzufügen). Diese Sicherheitsgruppe gewährt Mitgliedern einer anderen Sicherheitsgruppe Zugriff.
4. Wählen Sie für Type die Option Custom TCP Rule aus.
 - a. Geben Sie für Port Range den Port an, den Sie beim Erstellen des Clusters verwendet haben.

Der Standardport für Valkey- oder Redis OSS-Cluster und Replikationsgruppen ist 6379

- b. Geben Sie in das Feld Source die ersten Zeichen der ID der Sicherheitsgruppe ein. Wählen Sie aus der Liste die Sicherheitsgruppe aus, die Sie für Ihre EC2 Amazon-Instances verwenden möchten.
5. Wählen Sie Save, wenn Sie fertig sind.



Wenn Sie den Zugriff aktiviert haben, können Sie eine Verbindung zum Knoten herstellen, wie im nächsten Abschnitt beschrieben.

Informationen zum Zugriff auf Ihren ElastiCache Cluster von einer anderen Amazon VPC, einer anderen AWS Region oder sogar von Ihrem Unternehmensnetzwerk aus finden Sie im Folgenden:

- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#)
- [Zugriff auf ElastiCache Ressourcen von außen AWS](#)

Schritt 4. Connect zum Knoten des Clusters her

Bevor Sie fortfahren, müssen Sie [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) abschließen.

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon EC2 Getting Started Guide](#).

Eine EC2 Amazon-Instance kann nur dann eine Verbindung zu einem Clusterknoten herstellen, wenn Sie sie dazu autorisiert haben.

Finden Sie Ihre Knotenendpunkte

Wenn sich Ihr Cluster im Status „Verfügbar“ befindet und Sie den Zugriff darauf autorisiert haben, können Sie sich bei einer EC2 Amazon-Instance anmelden und eine Verbindung zum Cluster herstellen. Hierzu müssen Sie zuerst den Endpunkt bestimmen.

Finden Sie die Endpunkte eines Valkey-Clusters (Cluster-Modus deaktiviert) (Konsole)

Wenn ein Valkey-Cluster (Clustermodus deaktiviert) nur einen Knoten hat, wird der Endpunkt des Knotens sowohl für Lese- als auch für Schreibvorgänge verwendet. Wenn ein Cluster mehrere Knoten enthält, gibt es drei Arten von Endpunkten: den primären Endpunkt, den Reader-Endpunkt und die Knotenendpunkte.

Der primäre Endpunkt ist ein DNS-Name, der immer auf den primären Knoten im Cluster aufgelöst wird. Der primäre Endpunkt ist Änderungen an Ihrem Cluster, wie Heraufstufen einer Read Replica in die Rolle des primären Knotens, gegenüber immun. Für Schreibvorgänge empfehlen wir, dass Anwendungen sich ausschließlich mit dem primären Endpunkt verbinden.

Ein Reader-Endpunkt verteilt eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Read Replicas in einem ElastiCache Cluster. Zusätzliche Faktoren, z. B. wenn die Anwendung die Verbindungen erstellt oder wie die Anwendung die Verbindungen (erneut) verwendet, bestimmen die Verteilung des Datenverkehrs. Reader-Endpunkte bleiben hinsichtlich der Cluster-Änderungen beim Hinzufügen oder Entfernen von Replicas in Echtzeit auf dem aktuellen Stand. Sie können die mehreren Read Replicas Ihres ElastiCache Clusters in verschiedenen AWS Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Reader-Endpunkte sicherzustellen.

Note

Ein Leser-Endpunkt ist kein Load Balancer. Es handelt sich um einen DNS-Eintrag, der nach dem Rotationsprinzip zu einer IP-Adresse eines der Replikationsknoten aufgelöst wird.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpunkt werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren.

So finden Sie die Endpunkte eines Valkey-Clusters (Cluster-Modus deaktiviert)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Caches oder Redis OSS-Caches aus.

Der Cluster-Bildschirm wird mit einer Liste angezeigt, die alle vorhandenen serverlosen Valkey- oder Redis OSS-Caches, Valkey-Cluster (Cluster-Modus deaktiviert) und Valkey-Cluster (Cluster-Modus aktiviert) enthält. Wählen Sie den Cluster, den Sie im Abschnitt [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#) erstellt haben.

3. Um die Primary and/or Reader-Endpunkte des Clusters zu finden, wählen Sie den Namen des Clusters (nicht das Optionsfeld).

▼ Cluster details			
Cluster name	Description	Node type cache.r6g.large	Status Available
Engine Redis OSS	Engine version 6.0.5	Global datastore -	Global datastore role -
Update status Update available	Cluster mode Off	Shards 1	Number of nodes 3
Data tiering Disabled	Multi-AZ Enabled	Auto-failover Enabled	Encryption in transit Disabled
Encryption at rest Disabled	Parameter group default.redis6.x	Outpost ARN -	Configuration endpoint -
Primary endpoint [icon] [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [icon] [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Primär- und Reader-Endpunkte für einen Valkey-Cluster (Cluster-Modus deaktiviert)

Wenn sich im Cluster nur ein Knoten befindet, ist kein primärer Endpunkt vorhanden. Sie können mit dem nächsten Schritt fortfahren.

4. Wenn der Valkey-Cluster (Cluster-Modus deaktiviert) über Replikatknoten verfügt, können Sie die Replikatknoten-Endpunkte des Clusters finden, indem Sie den Namen des Clusters und dann die Registerkarte Knoten auswählen.

Auf dem angezeigten Knotenbildschirm wird jeder Knoten im Cluster, –primärer Knoten und Replikate, – mit seinem Endpunkt aufgelistet.

☐	Node Name	Status	Current Role	Port	Endpoint
☐	test-no-001	available	primary	6379	test-no-001.usw2.cache.amazonaws.com:6379
☐	test-no-002	available	replica	6379	test-no-002.usw2.cache.amazonaws.com:6379
☐	test-no-003	available	replica	6379	test-no-003.usw2.cache.amazonaws.com:6379

Knotenendpunkte für einen Valkey-Cluster (Clustermodus deaktiviert)

5. So kopieren Sie einen Endpunkt in Ihre Zwischenablage:
 - a. Suchen Sie einen Endpunkt nach dem anderen nach dem zu kopierenden Endpunkt.
 - b. Wählen Sie das Kopiersymbol direkt vor dem Endpunkt aus.

Der Endpunkt wird nun in Ihre Zwischenablage kopiert. Informationen zur Verwendung des Endpunkts zur Verbindung mit einem Knoten finden Sie unter [Mit Knoten verbinden](#).

Ein primärer Valkey-Endpunkt (Clustermodus deaktiviert) sieht etwa wie folgt aus. Abhängig davon, ob die Verschlüsselung während der Übertragung aktiviert ist oder nicht, unterscheidet sich dies.

Verschlüsselung während der Übertragung nicht aktiviert

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port
```

```
redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Verschlüsselung während der Übertragung aktiviert

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port
```

```
master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Informationen zu den Endpunkten finden Sie im entsprechenden Thema für die Engine und den Cluster-Typ, die von Ihnen ausgeführt werden.

- [Verbindungsendpunkte finden in ElastiCache](#)
- [Finden von Endpunkten für einen Valkey- oder Redis OSS-Cluster \(Clustermodus aktiviert\) \(Konsole\)](#)—Sie benötigen den Konfigurationsendpunkt des Clusters.
- [Suchen von Endpunkten \(AWS CLI\)](#)
- [Endpunkte finden \(API\) ElastiCache](#)

Stellen Sie eine Connect zu einem Valkey- oder Redis-OSS-Cluster oder einer Replikationsgruppe her (Linux)

Da Sie nun den benötigten Endpunkt haben, können Sie sich bei einer EC2 Instanz anmelden und eine Verbindung zum Cluster oder zur Replikationsgruppe herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Die neueste Version von `valkey-cli` unterstützt auch die Verbindung aktivierter Cluster. SSL/TLS encryption/authentication

Im folgenden Beispiel werden EC2 Amazon-Instances verwendet, auf denen Amazon Linux und Amazon Linux 2 ausgeführt werden. Einzelheiten zur Installation und Kompilierung von Valkey-Cli mit anderen Linux-Distributionen finden Sie in der Dokumentation für Ihr spezielles Betriebssystem.

Note

Dieser Prozess umfasst das Testen einer Verbindung mit dem Valkey-Cli-Hilfsprogramm nur für den ungeplanten Gebrauch. [Eine Liste der unterstützten Valkey- und Redis OSS-Clients finden Sie in der Valkey-Dokumentation.](#) Beispiele für die Verwendung von `with` finden Sie unter AWS SDKs . ElastiCache [Tutorials: Erste Schritte mit Python und ElastiCache](#)

Herstellen einer Verbindung zu einem unverschlüsselten Cluster mit deaktiviertem Cluster-Modus

1. Führen Sie den folgenden Befehl aus, um eine Verbindung zum Cluster herzustellen *primary-endpoint* und *port number* durch den Endpunkt Ihres Clusters und Ihre Portnummer zu ersetzen. (Der Standardport für Valkey oder Redis OSS ist 6379.)

```
src/valkey-cli -h primary-endpoint -p port number
```

Das Ergebnis einer Valkey- oder Redis OSS-Eingabeaufforderung sieht in etwa wie folgt aus:

```
primary-endpoint:port number
```

2. Sie können jetzt Valkey- oder Redis OSS-Befehle ausführen.

```
set x Hello
OK

get x
"Hello"
```

Herstellen einer Verbindung mit einem unverschlüsselten Cluster mit aktiviertem Cluster-Modus

1. Führen Sie den folgenden Befehl aus, um eine Verbindung zum Cluster herzustellen und zu ersetzen *configuration-endpoint* und *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer zu ersetzen. (Der Standardport für Valkey oder Redis OSS ist 6379.)

```
src/valkey-cli -h configuration-endpoint -c -p port number
```

Note

Im vorherigen Befehl aktiviert Option -c den Cluster-Modus nach den [-ASK und -MOVED-Umleitungen](#).

Das Ergebnis einer Valkey- oder Redis OSS-Eingabeaufforderung sieht in etwa wie folgt aus:

```
configuration-endpoint:port number
```

2. Sie können jetzt Valkey- oder Redis OSS-Befehle ausführen. Beachten Sie, dass die Umleitung erfolgt, weil Sie diese mit der Option -c aktiviert haben. Wenn die Umleitung nicht aktiviert ist, gibt der Befehl die Fehlermeldung MOVED zurück. Weitere Informationen zum MOVED-Fehler finden Sie in der [Redis OSS-Clusterspezifikation](#).

```
set x Hi
-> Redirected to slot [16287] located at 172.31.28.122:6379
OK
set y Hello
OK
get y
"Hello"
set z Bye
-> Redirected to slot [8157] located at 172.31.9.201:6379
OK
get z
"Bye"
get x
-> Redirected to slot [16287] located at 172.31.28.122:6379
"Hi"
```

Verbindung zu einem Encryption/Authentication aktivierten Cluster herstellen

Standardmäßig verwendet valkey-cli eine unverschlüsselte TCP-Verbindung, wenn eine Verbindung zu Valkey oder Redis OSS hergestellt wird. Die Option wird zum SSL/TLS Zeitpunkt der Valkey-CLI-Kompilierung BUILD_TLS=yes aktiviert, wie im vorherigen Abschnitt gezeigt. [Laden Sie den Befehlszeilenzugriff herunter und richten Sie ihn ein](#) Die Aktivierung von AUTH ist optional. Sie müssen jedoch Verschlüsselung während der Übertragung aktivieren, um AUTH zu aktivieren. Weitere Informationen zur ElastiCache Verschlüsselung und Authentifizierung finden Sie unter [ElastiCache Verschlüsselung bei der Übertragung \(TLS\)](#)

Note

Sie können die Option `--tls` mit valkey-cli verwenden, um eine Verbindung zu verschlüsselten Clustern herzustellen, die im Clustermodus aktiviert oder deaktiviert sind. Wenn für einen Cluster ein AUTH-Token festgelegt ist, können Sie die Option `-a` verwenden, um ein AUTH-Passwort anzugeben.

Stellen Sie in den folgenden Beispielen sicher, dass Sie *cluster-endpoint* und *port number* durch den Endpunkt Ihres Clusters und Ihre Portnummer ersetzen. (Der Standardport für Valkey oder Redis OSS ist 6379.)

Verbindung zu verschlüsselten Clustern mit Cluster-Modus deaktiviert

Das folgende Beispiel verbindet zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung:

```
src/valkey-cli -h cluster-endpoint --tls -a your-password -p port number
```

Das folgende Beispiel verbindet zu einem Cluster nur mit aktivierter Verschlüsselung:

```
src/valkey-cli -h cluster-endpoint --tls -p port number
```

Verbindung zu verschlüsselten Clustern mit Cluster-Modus aktiviert

Das folgende Beispiel verbindet zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung:

```
src/valkey-cli -c -h cluster-endpoint --tls -a your-password -p port number
```

Das folgende Beispiel verbindet zu einem Cluster nur mit aktivierter Verschlüsselung:

```
src/valkey-cli -c -h cluster-endpoint --tls -p port number
```

Nachdem Sie eine Verbindung zum Cluster hergestellt haben, können Sie die Befehle Valkey oder Redis OSS ausführen, wie in den vorherigen Beispielen für unverschlüsselte Cluster gezeigt.

Alternative zu Valkey-Cli

Wenn für den Cluster der Clustermodus nicht aktiviert ist und Sie für einen kurzen Test eine Verbindung zum Cluster herstellen müssen, ohne die Valkey-CLI-Kompilierung durchzuführen, können Sie Telnet oder OpenSSL verwenden. Stellen Sie sicher, dass Sie in den folgenden Beispielbefehlen *cluster-endpoint* und *port number* durch den Endpunkt Ihres Clusters und Ihre Portnummer ersetzen. (Der Standardport für Valkey oder Redis OSS ist 6379.)

Im folgenden Beispiel wird eine Verbindung zu einem Cluster mit aktivierter Verschlüsselung, and/or Authentifizierung und deaktiviertem Clustermodus hergestellt:

```
openssl s_client -connect cluster-endpoint:port number
```

Wenn für den Cluster ein Passwort festgelegt ist, stellen Sie zunächst eine Verbindung mit dem Cluster her. Nachdem Sie eine Verbindung hergestellt haben, authentifizieren Sie den Cluster mit

dem folgenden Befehl, und drücken Sie dann die Enter-Taste. Im folgenden Beispiel ersetzen Sie es *your-password* durch das Passwort für Ihren Cluster.

```
Auth your-password
```

Im folgenden Beispiel wird eine Verbindung zu einem deaktivierten Cluster im Cluster-Modus mit Verschlüsselung und / oder Authentifizierung hergestellt:

```
telnet cluster-endpoint port number
```

Stellen Sie eine Connect zu einem Valkey- oder Redis-OSS-Cluster oder einer Replikationsgruppe her (Windows)

Um von einer EC2 Windows-Instance aus mithilfe der Valkey CLI oder Redis OSS CLI eine Verbindung zum Valkey- oder Redis OSS-Cluster herzustellen, müssen Sie das valkey-cli-Paket herunterladen und valkey-cli.exe verwenden, um von einer Windows-Instanz aus eine Verbindung zum Valkey- oder Redis-OSS-Cluster herzustellen. EC2

Im folgenden Beispiel verwenden Sie das Valkey-CLI-Hilfsprogramm, um eine Verbindung zu einem Cluster herzustellen, der nicht verschlüsselungsfähig ist und auf dem Valkey oder Redis OSS ausgeführt wird. Weitere Informationen zu Valkey oder Redis OSS und verfügbaren Befehlen finden Sie unter Valkey- und Redis OSS-Befehle auf der [Valkey-Website](#).

So stellen Sie mithilfe von Valkey-CLI eine Verbindung zu einem Valkey- oder Redis-OSS-Cluster her, der nicht verschlüsselungsfähig ist

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her. Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2 Erste Schritte“](#).
2. Kopieren Sie den Link und fügen Sie ihn <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> in einen Internetbrowser ein, um die ZIP-Datei für den Valkey-Client aus der verfügbaren Version unter herunterzuladen GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Extrahieren Sie die ZIP-Datei in den gewünschten Ordner/Pfad.

Öffnen Sie die Befehlszeile, wechseln Sie in das Valkey-Verzeichnis und führen Sie den Befehl aus. c:\Valkey>valkey-cli -h *Redis_Cluster_Endpoint* -p 6379

Beispiel:

```
c:\Valkey>valkey-cli -h cmd.xxxxxxxx.ng.0001.usw2.cache.amazonaws.com -p 6379
```

3. Führen Sie die Befehle Valkey oder Redis OSS aus.

Sie sind jetzt mit dem Cluster verbunden und können Valkey- oder Redis OSS-Befehle wie die folgenden ausführen.

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5    // Set key "b" with a string value and a 5 second expiration
"Good-bye"
get b                   // Get value for key "b"
"Good-bye"

                        // wait >= 5 seconds

get b
(nil)                   // key has expired, nothing returned
quit                    // Exit from valkey-cli
```

Wie geht es weiter?

Nachdem Sie die Übung Erste Schritte ausprobiert haben, können Sie in den folgenden Abschnitten mehr über ElastiCache und verfügbare Tools erfahren:

- [Erste Schritte mit AWS](#)
- [Tools für Amazon Web Services](#)
- [AWS-Befehlszeilenschnittstelle](#)
- [ElastiCache Amazon-API-Referenz](#)

Nachdem Sie die Übung Erste Schritte abgeschlossen haben, können Sie die folgenden Abschnitte lesen, um mehr über die ElastiCache Verwaltung zu erfahren:

- [Auswahl der Knotengröße](#)

Das Cache muss groß genug für alle Daten sein, die in das Cache aufgenommen werden sollen. Dabei möchten Sie auch nicht für mehr Cache bezahlen, als Sie benötigen. Verwenden Sie dieses Thema, um zu erfahren, wie Sie die beste Knotengröße auswählen.

- [ElastiCache Best Practices und Caching-Strategien](#)

Identifizieren und Beheben von Problemen, die die Effizienz Ihres Clusters beeinträchtigen können.

Einen knotenbasierten ElastiCache Cluster für Redis OSS erstellen

Im Folgenden sind die einmaligen Aktionen aufgeführt, die Sie ergreifen müssen, um einen knotenbasierten ElastiCache Cluster für Redis OSS zu erstellen.

Weitere Informationen zur Einrichtung finden Sie unter. ElastiCache [Einrichten ElastiCache](#)

Themen

- [Schritt 1: Erstellung einer Subnetzgruppe](#)
- [Schritt 2: Erstellen eines Clusters](#)
- [Schritt 3: Zugriff auf den Cluster autorisieren](#)
- [Schritt 4: Herstellen einer Verbindung zu einem Cluster-Knoten](#)

Schritt 1: Erstellung einer Subnetzgruppe

Erstellen Sie eine Subnetz-Gruppe, bevor Sie einen Cluster erstellen. Eine Cache-Subnetzgruppe ist eine Sammlung von Subnetzen, die Sie möglicherweise für Ihre Cluster in einer VPC festlegen möchten. Wenn Sie einen Cluster in einer VPC starten, müssen Sie eine Cache-Subnetzgruppe auswählen. ElastiCache verwendet dann diese Cache-Subnetzgruppe, um jedem Cache-Knoten im Cluster IP-Adressen innerhalb dieses Subnetzes zuzuweisen.

Wenn Sie eine neue Subnetzgruppe erstellen, notieren Sie sich die Anzahl der verfügbaren IP-Adressen. Wenn das Subnetz nur über sehr wenige freie IP-Adressen verfügt, sind Sie möglicherweise hinsichtlich der Anzahl der Knoten, die Sie dem Cluster hinzufügen können, eingeschränkt. Um dieses Problem zu beheben, können Sie einer Subnetzgruppe ein oder mehrere Subnetze zuweisen, sodass Sie über eine ausreichende Anzahl von IP-Adressen in der Availability Zone Ihres Clusters verfügen. Danach können Sie dem Cluster weitere Knoten hinzufügen.

Die folgenden Verfahren zeigen Ihnen, wie Sie eine Subnetzgruppe mit dem Namen mysubnetgroup (Konsole) und die AWS CLI erstellen.

Erstellen einer Subnetzgruppe (Konsole)

Im folgenden Verfahren wird das Erstellen einer Subnetzgruppe (Konsole) erläutert.

Erstellen einer DB-Sicherheitsgruppe (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Klicken Sie in der Navigationsliste auf Subnet Groups.
3. Klicken Sie auf Create Subnet Group (Subnetzgruppe ändern).
4. Gehen Sie im Assistenten Create Subnet Group wie folgt vor. Wenn Sie alle gewünschten Einstellungen vorgenommen haben, klicken Sie auf Yes, Create.
 - a. Geben Sie im Feld Name einen Namen für Ihre Subnetzgruppe ein.
 - b. Geben Sie im Feld Description eine Beschreibung für Ihre Subnetzgruppe ein.
 - c. Wählen Sie im Feld VPC ID die erstellte Amazon VPC aus.
 - d. Wählen Sie unter Availability Zone und Subnetz-ID die Availability Zone oder [Local Zone](#) und die ID Ihres privaten Subnetzes aus und klicken Sie auf Hinzufügen.

Subnet group settings

A subnet group is a collection of subnets (typically private). Designate a subnet group for your clusters running in an Amazon Virtual Private Cloud (VPC) environment.

Name

The name is required, can have up to 255 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Description - optional

VPC ID

The identifier for the VPC environment where your cluster is to run.

 ▼ [Create VPC](#)

For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.

Selected subnets (6) [Manage](#)

Availability Zone ▲	Subnet ID ▼	Outpost ID ▼	CIDR block ▼
us-east-1a	subnet-		172.31.16.0/20
us-east-1b	subnet-		172.31.32.0/20
us-east-1c	subnet-		172.31.0.0/20
us-east-1d	subnet-		172.31.80.0/20

5. Klicken Sie in der angezeigten Bestätigungsmeldung auf Close.

Ihre neue Subnetzgruppe wird in der Liste der Subnetzgruppen der ElastiCache Konsole angezeigt. Unten im Fenster können Sie die Subnetzgruppe auswählen, um Details wie die der Gruppe zugeordneten Subnetze anzuzeigen.

Erstellen einer Subnetzgruppe (AWS CLI)

Geben Sie in einem Befehlszeilenfenster den Befehl `create-cache-subnet-group` ein, um eine Subnetzgruppe zu erstellen.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-subnet-group \
  --cache-subnet-group-name mysubnetgroup \
  --cache-subnet-group-description "Testing" \
```

```
--subnet-ids subnet-53df9c3a
```

Für Windows:

```
aws elasticache create-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup ^  
  --cache-subnet-group-description "Testing" ^  
  --subnet-ids subnet-53df9c3a
```

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

```
{  
  "CacheSubnetGroup": {  
    "VpcId": "vpc-37c3cd17",  
    "CacheSubnetGroupDescription": "Testing",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-53df9c3a",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        }  
      }  
    ],  
    "CacheSubnetGroupName": "mysubnetgroup"  
  }  
}
```

Weitere Informationen finden Sie im AWS CLI Thema. [create-cache-subnet-group](#)

Schritt 2: Erstellen eines Clusters

Bevor Sie einen Cluster für die Produktion erstellen, müssen Sie natürlich überlegen, wie Sie den Cluster entsprechend Ihren geschäftlichen Anforderungen konfigurieren. Diese Probleme werden im [Einen Cluster vorbereiten in ElastiCache](#)-Abschnitt adressiert. Für die Zwecke dieser Übung „Erste Schritte“ erstellen Sie einen Cluster mit deaktiviertem Cluster-Modus. Sie können die Standardkonfigurationswerte dort akzeptieren, wo sie zutreffen.

Der Cluster, den Sie gleich starten werden, wird live sein und nicht in einer Sandbox ausgeführt. Es fallen die ElastiCache Standardnutzungsgebühren für die Instance an, bis Sie sie löschen. Die Gesamtkosten werden minimal sein (meistens geringer als ein Dollar), wenn Sie diese Übung in einer

Sitzung durchlaufen und den Cluster löschen, sobald Sie die Übung abgeschlossen haben. Weitere Informationen zu den ElastiCache Nutzungsraten finden Sie auf [Amazon ElastiCache](https://aws.amazon.com/elasticache/).

Ihr Cluster wird, basierend auf dem Amazon-VPC-Service, in einer Virtual Private Cloud (VPC) gestartet.

Erstellen eines Redis OSS-Clusters (Cluster-Modus deaktiviert) (Konsole)

Um einen Redis OSS-Cluster (Clustermodus deaktiviert) mithilfe der Konsole zu erstellen ElastiCache

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der Sie diesen Cluster starten möchten.
3. Wählen Sie im Navigationsbereich Get started (Erste Schritte) aus.
4. Wählen Sie Create VPC (VPC erstellen) aus und befolgen Sie die Schritte unter [Erstellen einer Virtual Private Cloud \(VPC\)](#).
5. Wählen Sie auf der ElastiCache Dashboard-Seite Valkey-Cache oder Redis OSS-Cache aus. Für diese Übung wählen wir Redis OSS-Cache und dann Create Redis OSS-Cache.
6. Führen Sie unter Cluster settings (Cluster-Einstellungen) die folgenden Schritte aus:
 - a. Wählen Sie Configure and create a new cluster (Neuen Cluster konfigurieren und erstellen) aus.
 - b. Wählen Sie für Cluster mode (Cluster-Modus) Disabled (Deaktiviert) aus.
 - c. Geben Sie für Cluster info (Cluster-Info) einen Wert für Name (Name) ein.
 - d. (Optional) Geben Sie einen Wert für Description (Beschreibung) ein.
7. Unter Location (Speicherort):

AWS Cloud

1. Wir empfehlen für AWS Cloud die Voreinstellungen für Multi-AZ und Auto-failover (Automatisches Failover) zu akzeptieren. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache für Redis](#) OSS mit Multi-AZ.
2. Unter Cluster settings (Cluster-Einstellungen):
 - a. Wählen Sie unter Engine version (Engine-Version) eine verfügbare Version aus.

- b. Verwenden Sie für Port den Standardport 6379. Wenn es einen Grund gibt, einen anderen Port zu verwenden, geben Sie die betreffende Portnummer ein.
- c. Wählen Sie für die Parameter group (Parametergruppe) eine Parametergruppe aus oder erstellen Sie eine neue Parametergruppe. Parametergruppen steuern die Laufzeitparameter Ihres Clusters. Weitere Informationen zu Parametergruppen finden Sie unter [Valkey- und Redis OSS-Parameter](#) und [Eine ElastiCache Parametergruppe erstellen](#).

 Note

Wenn Sie eine Parametergruppe zum Festlegen der Engine-Konfigurationswerte auswählen, wird diese Parametergruppe auf alle Cluster im globalen Datenspeicher angewendet. Auf der Seite Parameter Groups (Parametergruppen) gibt das Ja/nein-Attribut Global an, ob eine Parteigruppe Teil eines globalen Datenspeichers ist.

- d. Klicken Sie für Node type (Knotentyp) auf den Abwärtspfeil (▼)).
Wählen Sie im Dialogfeld Change node type (Knotentyp ändern) einen Wert für Instance family (Instance-Familie) für den gewünschten Knotentyp aus. Wählen Sie dann den Knotentyp aus, den Sie für diesen Cluster verwenden möchten, und wählen Sie dann Save (Speichern).

Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Wenn Sie einen r6gd-Knotentyp wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

- e. Wählen Sie unter Number of replicas (Anzahl der Repliken) die für diesen Cluster gewünschte Anzahl von Read Replicas aus. Wenn Multi-AZ aktiviert ist, muss die Zahl zwischen 1–5 liegen.
3. Unter Connectivity (Konnektivität)
- a. Wählen Sie als Network type (Netzwerktyp) die IP-Version(en) aus, die dieser Cluster unterstützen soll.
 - b. Wählen Sie für Subnetzgruppen das Subnetz aus, das Sie auf diesen Cluster anwenden möchten. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten

zugeordnet werden sollen. ElastiCache Cluster benötigen ein Dual-Stack-Subnetz mit beiden IPv4 und ihnen zugewiesenen IPv6 Adressen, um im Dual-Stack-Modus betrieben werden zu können, und ein Subnetz nur für den Betrieb als IPv6 -only. IPv6

Geben Sie beim Erstellen einer neuen Subnetzgruppe die VPC ID (VPC-ID) ein, zu der sie gehört.

Weitere Informationen finden Sie unter:

- [Auswahl eines Netzwerktyps in ElastiCache.](#)
- [Erstellen eines Subnetzes in der VPC](#)

Wenn Sie [Verwenden von lokalen Zonen mit ElastiCache](#) sind, müssen Sie ein Subnetz erstellen oder auswählen, das in der lokalen Zone liegt.

Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).

4. Für Availability zone placements (Availability-Zone-Platzierungen) haben Sie zwei Optionen:

- Keine Präferenz — wählt die Availability Zone. ElastiCache
- Availability Zones angeben – Sie geben die Availability Zone für jeden Cluster an.

Wenn Sie die Availability Zones angeben, wählen Sie für jeden Cluster in jedem Shard die Availability Zone aus der Liste aus.

Weitere Informationen finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

5. Wählen Sie Next (Weiter).
6. Unter Erweiterte Redis OSS-Einstellungen

- Für Security (Sicherheit):
 - i. Zur Verschlüsselung Ihrer Daten haben Sie die folgenden Optionen:
 - Verschlüsselung im Ruhezustand – Ermöglicht die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

 Note

Sie haben die Möglichkeit, einen anderen Verschlüsselungsschlüssel anzugeben, indem Sie Customer Managed AWS KMS Key und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwendung von kundenverwalteten Schlüsseln aus AWS-KMS](#).

- Verschlüsselung während der Übertragung – Ermöglicht die Verschlüsselung von Daten während der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für ElastiCache Engine-Version 6.0 für Redis OSS und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:
 - Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
 - Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern aus, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
 - AUTH Default User — Ein Authentifizierungsmechanismus für Valkey- und Redis OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).
- AUTH — Ein Authentifizierungsmechanismus für den Redis OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).

 Note

Für Redis OSS-Versionen zwischen 3.2.6 und höher, mit Ausnahme von Version 3.2.10, ist Redis OSS AUTH die einzige Option.

- ii. Wählen Sie für Security groups (Sicherheitsgruppen) die gewünschten Sicherheitsgruppen für diesen Cluster aus. Eine security group (Sicherheitsgruppe) fungiert als Firewall, um den Netzwerkzugriff auf Ihren Cluster zu steuern. Sie können die Standardsicherheitsgruppe für Ihre VPC verwenden oder eine neue erstellen.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Sicherheitsgruppen für Ihre VPC](#) im Benutzerhandbuch zu Amazon VPC.

7. Wenn Sie regelmäßig geplante automatische Sicherungen möchten, aktivieren Sie Enable automatic backups und geben Sie ein, wie viele Tage lang jede automatische Sicherung beibehalten werden soll, bevor sie automatisch gelöscht wird. Wenn Sie keine regelmäßig geplanten automatischen Sicherungen möchten, deaktivieren Sie das Kontrollkästchen Enable automatic backups. In beiden Fällen haben Sie jederzeit die Option, manuelle Sicherungen zu erstellen.

Weitere Informationen zu Sicherung und Wiederherstellung finden Sie unter [Snapshot und Wiederherstellung](#).

8. (Optional) Geben Sie ein Wartungsfenster an. Das Maintenance window (Wartungsfenster) ist der (in der Regel eine Stunde lange) allwöchentliche Zeitraum, für den ElastiCache die Systemwartung Ihres Clusters plant. Sie können es ElastiCache überlassen, einen Tag und eine Uhrzeit für das Wartungsfenster auszuwählen (No preference (Keine Präferenz)) oder Sie können den Tag, die Uhrzeit und die Dauer selbst wählen (Specify maintenance window (Wartungsfenster angeben)). Treffen Sie bei Wahl von Specify maintenance window eine Auswahl in den Listen Start day, Start time und Duration (in Stunden) für Ihr Wartungsfenster. Alle Zeiten sind UCT-Zeiten.

Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

9. (Optional) Für Protokolle:
 - Unter Protokollformat wählen Sie entweder Text oder JSON aus.
 - Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen aus und geben Sie entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und wählen Sie dann entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Streamnamen aus.
10. Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen unter Tags zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).
11. Wählen Sie Weiter aus.

12. Überprüfen Sie alle Ihre Einträge und ausgewählten Optionen und machen Sie dann evtl. erforderliche Korrekturen. Sobald Sie bereit sind, klicken Sie auf Create (Erstellen).

On premises

1. Für On premises (On-Premises) empfehlen wir Ihnen, Auto-failover (Automatisches Failover) aktiviert zu lassen. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache für Redis OSS](#) mit Multi-AZ
2. Um die Erstellung des Clusters abzuschließen, befolgen Sie die Schritte unter [Verwenden von Outposts](#).

Sobald der Status Ihres Clusters verfügbar ist, können Sie Amazon EC2 Zugriff darauf gewähren, eine Verbindung herstellen und ihn verwenden. Weitere Informationen erhalten Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Redis OSS-Clusters (Cluster-Modus deaktiviert) (AWS CLI)

Example

Der folgende CLI-Code erstellt einen Redis OSS-Cluster (Clustermodus deaktiviert) ohne Replikate.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine redis \  
--num-cache-nodes 1 \  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Für Windows:

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine redis ^  
--num-cache-nodes 1 ^  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Informationen zum Arbeiten mit aktiviertem Cluster-Modus finden Sie in den folgenden Themen:

- Wenn Sie die Konsole verwenden möchten, lesen Sie [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#).
- Informationen zur Verwendung von finden Sie unter AWS CLI. [Erstellen eines Valkey- oder Redis OSS-Clusters \(Clustermodus aktiviert\) \(AWS CLI\)](#)

Schritt 3: Zugriff auf den Cluster autorisieren

In diesem Abschnitt wird davon ausgegangen, dass Sie mit dem Starten und Herstellen einer Verbindung zu EC2 Amazon-Instances vertraut sind. Weitere Informationen finden Sie im [Amazon EC2 Getting Started Guide](#).

Alle ElastiCache Cluster sind für den Zugriff von einer EC2 Amazon-Instance aus konzipiert. Das gängigste Szenario ist der Zugriff auf einen ElastiCache Cluster von einer EC2 Amazon-Instance in derselben Amazon Virtual Private Cloud (Amazon VPC), was bei dieser Übung der Fall sein wird.

Standardmäßig ist der Netzwerkzugriff auf Ihren Cluster auf das Konto beschränkt, über das er erstellt wurde. Bevor Sie von einer EC2 Instance aus eine Verbindung zu einem Cluster herstellen können, müssen Sie die EC2 Instance für den Zugriff auf den Cluster autorisieren. Die erforderlichen Schritte hängen davon ab, ob Sie Ihren Cluster in EC2 VPC oder EC2 -Classic gestartet haben.

Der häufigste Anwendungsfall ist, wenn eine auf einer EC2 Instance bereitgestellte Anwendung eine Verbindung zu einem Cluster in derselben VPC herstellen muss. Der einfachste Weg, den Zugriff zwischen EC2 Instances und Clustern in derselben VPC zu verwalten, ist wie folgt:

1. Erstellen Sie eine VPC-Sicherheitsgruppe für Ihren Cluster. Diese Sicherheitsgruppe kann verwendet werden, um den Zugriff auf die Cluster-Instances zu beschränken. Sie können für diese Sicherheitsgruppe beispielsweise eine benutzerdefinierte Regel erstellen, die TCP-Zugriff über den Port, den Sie dem Cluster bei seiner Erstellung zugeordnet haben, und eine IP-Adresse gewährt, mit der Sie auf den Cluster zugreifen.

Der Standardport für Redis OSS-Cluster und Replikationsgruppen ist 6379

Important

ElastiCache Amazon-Sicherheitsgruppen gelten nur für Cluster, die nicht in einer Amazon Virtual Private Cloud-Umgebung (VPC) ausgeführt werden. Wenn Sie ein Cluster in einer Amazon Virtual Private Cloud betreiben, ist die Option Sicherheitsgruppen im Navigationsbereich der Konsole nicht verfügbar.

Wenn Sie Ihre ElastiCache Knoten in einer Amazon VPC ausführen, kontrollieren Sie den Zugriff auf Ihre Cluster mit Amazon VPC-Sicherheitsgruppen, die sich von ElastiCache Sicherheitsgruppen unterscheiden. Weitere Informationen zur Verwendung ElastiCache in einer Amazon VPC finden Sie unter [Amazon VPCs und ElastiCache Sicherheit](#)

2. Erstellen Sie eine VPC-Sicherheitsgruppe für Ihre EC2 Instances (Web- und Anwendungsserver). Diese Sicherheitsgruppe kann bei Bedarf den Zugriff auf die EC2 Instance aus dem Internet über die Routingtabelle der VPC ermöglichen. Sie können beispielsweise Regeln für diese Sicherheitsgruppe festlegen, um den TCP-Zugriff auf die EC2 Instance über Port 22 zu ermöglichen.
3. Erstellen Sie in der Sicherheitsgruppe für Ihren Cluster benutzerdefinierte Regeln, die Verbindungen von der Sicherheitsgruppe aus zulassen, die Sie für Ihre EC2 Instances erstellt haben. Damit wird jedem Mitglied der Sicherheitsgruppe der Zugriff auf die DB-Instances gestattet.

 Note

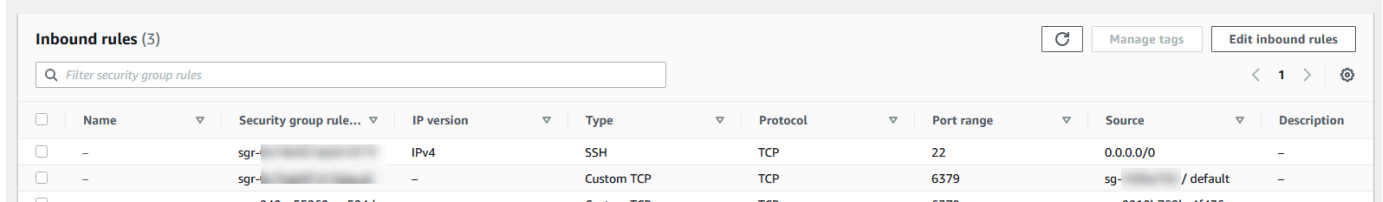
Wenn Sie planen, sie zu verwenden [Verwenden von lokalen Zonen mit ElastiCache](#), stellen Sie sicher, dass Sie sie aktiviert haben. Wenn Sie eine Subnetzgruppe in dieser Local Zone erstellen, wird Ihre VPC auf diese Local Zone erweitert, und Ihre VPC behandelt das Subnetz wie jedes Subnetz in einer anderen Availability Zone. Alle relevanten Gateways und Routing-Tabellen werden automatisch angepasst.

So erstellen Sie eine Regel in einer VPC-Sicherheitsgruppe, die Verbindungen über eine andere Sicherheitsgruppe zulässt

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc>.
2. Wählen Sie im Navigationsbereich Security Groups (Sicherheitsgruppen) aus.
3. Wählen Sie eine Sicherheitsgruppe aus oder erstellen Sie eine, die Sie für Ihre Cluster-Instances verwenden werden. Wählen Sie unter Inbound Rules (Eingangsregeln) die Option Edit Inbound Rules (Eingangsregeln bearbeiten) und dann Add Rule (Regeln hinzufügen). Diese Sicherheitsgruppe gewährt Mitgliedern einer anderen Sicherheitsgruppe Zugriff.
4. Wählen Sie für Type die Option Custom TCP Rule aus.
 - a. Geben Sie für Port Range den Port an, den Sie beim Erstellen des Clusters verwendet haben.

Der Standardport für Redis OSS-Cluster und Replikationsgruppen ist. 6379

- b. Geben Sie in das Feld Source die ersten Zeichen der ID der Sicherheitsgruppe ein. Wählen Sie aus der Liste die Sicherheitsgruppe aus, die Sie für Ihre EC2 Amazon-Instances verwenden möchten.
5. Wählen Sie Save, wenn Sie fertig sind.



Wenn Sie den Zugriff aktiviert haben, können Sie eine Verbindung zum Knoten herstellen, wie im nächsten Abschnitt beschrieben.

Informationen zum Zugriff auf Ihren ElastiCache Cluster von einer anderen Amazon VPC, einer anderen AWS Region oder sogar von Ihrem Unternehmensnetzwerk aus finden Sie im Folgenden:

- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#)
- [Zugriff auf ElastiCache Ressourcen von außen AWS](#)

Schritt 4: Herstellen einer Verbindung zu einem Cluster-Knoten

Bevor Sie fortfahren, müssen Sie [Schritt 3: Zugriff auf den Cluster autorisieren](#) abschließen.

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon EC2 Getting Started Guide](#).

Eine EC2 Amazon-Instance kann nur dann eine Verbindung zu einem Clusterknoten herstellen, wenn Sie sie dazu autorisiert haben.

Finden Sie Ihre Knotenendpunkte

Wenn sich Ihr Cluster im Status „Verfügbar“ befindet und Sie den Zugriff darauf autorisiert haben, können Sie sich bei einer EC2 Amazon-Instance anmelden und eine Verbindung zum Cluster herstellen. Hierzu müssen Sie zuerst den Endpunkt bestimmen.

Finden Sie die Endpunkte (Konsole) eines Valkey- oder Redis OSS-Clusters (Cluster-Modus deaktiviert)

Wenn ein Redis OSS-Cluster (Clustermodus deaktiviert) nur einen Knoten hat, wird der Endpunkt des Knotens sowohl für Lese- als auch für Schreibvorgänge verwendet. Wenn ein Cluster mehrere Knoten enthält, gibt es drei Arten von Endpunkten: den primären Endpunkt, den Reader-Endpunkt und die Knotenendpunkte.

Der primäre Endpunkt ist ein DNS-Name, der immer auf den primären Knoten im Cluster aufgelöst wird. Der primäre Endpunkt ist Änderungen an Ihrem Cluster, wie Heraufstufen einer Read Replica in die Rolle des primären Knotens, gegenüber immun. Für Schreibvorgänge empfehlen wir, dass Anwendungen sich ausschließlich mit dem primären Endpunkt verbinden.

Ein Reader-Endpunkt teilt eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Read Replicas in einem ElastiCache for Redis OSS-Cluster auf. Zusätzliche Faktoren, z. B. wenn die Anwendung die Verbindungen erstellt oder wie die Anwendung die Verbindungen (erneut) verwendet, bestimmen die Verteilung des Datenverkehrs. Reader-Endpunkte bleiben hinsichtlich der Cluster-Änderungen beim Hinzufügen oder Entfernen von Replicas in Echtzeit auf dem aktuellen Stand. Sie können die mehreren Read Replicas Ihres ElastiCache Redis OSS-Clusters in verschiedenen AWS Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Leser-Endpunkte sicherzustellen.

Note

Ein Leser-Endpoint ist kein Load Balancer. Es handelt sich um einen DNS-Eintrag, der nach dem Rotationsprinzip zu einer IP-Adresse eines der Replikationsknoten aufgelöst wird.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpoint werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren.

So finden Sie die Endpunkte eines Redis OSS-Clusters (Clustermodus deaktiviert)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Redis OSS-Caches aus.

Der Cluster-Bildschirm wird mit einer Liste angezeigt, die alle vorhandenen serverlosen Valkey- oder Redis OSS-Caches, Redis OSS-Cluster (Clustermodus deaktiviert) und Redis OSS-Cluster (Clustermodus aktiviert) enthält. Wählen Sie den Cluster, den Sie im Abschnitt [Erstellen eines Redis OSS-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#) erstellt haben.

3. Um die Primary and/or Reader-Endpunkte des Clusters zu finden, wählen Sie den Namen des Clusters (nicht das Optionsfeld).

▼ Cluster details			
Cluster name	Description	Node type cache.r6g.large	Status Available
Engine Redis OSS	Engine version 6.0.5	Global datastore -	Global datastore role -
Update status Update available	Cluster mode Off	Shards 1	Number of nodes 3
Data tiering Disabled	Multi-AZ Enabled	Auto-failover Enabled	Encryption in transit Disabled
Encryption at rest Disabled	Parameter group default.redis6.x	Outpost ARN -	Configuration endpoint -
Primary endpoint [icon] [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [icon] [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Primär- und Reader-Endpunkte für einen Redis OSS-Cluster (Clustermodus deaktiviert)

Wenn sich im Cluster nur ein Knoten befindet, ist kein primärer Endpunkt vorhanden. Sie können mit dem nächsten Schritt fortfahren.

4. Wenn der Redis OSS-Cluster (Cluster-Modus deaktiviert) über Replikatknoten verfügt, können Sie die Replikatknoten-Endpunkte des Clusters finden, indem Sie den Namen des Clusters und dann die Registerkarte Knoten auswählen.

Auf dem angezeigten Knotenbildschirm wird jeder Knoten im Cluster, –primärer Knoten und Replikate, – mit seinem Endpunkt aufgelistet.

☐	Node Name	Status	Current Role	Port	Endpoint
☐	test-no-001	available	primary	6379	10.0.0.1:6379:amazonaws.com
☐	test-no-002	available	replica	6379	10.0.0.2:6379:amazonaws.com
☐	test-no-003	available	replica	6379	10.0.0.3:6379:amazonaws.com

Knotenendpunkte für einen Redis OSS-Cluster (Clustermodus deaktiviert)

5. So kopieren Sie einen Endpunkt in Ihre Zwischenablage:
 - a. Suchen Sie einen Endpunkt nach dem anderen nach dem zu kopierenden Endpunkt.
 - b. Wählen Sie das Kopiersymbol direkt vor dem Endpunkt aus.

Der Endpunkt wird nun in Ihre Zwischenablage kopiert. Informationen zur Verwendung des Endpunkts zur Verbindung mit einem Knoten finden Sie unter [Mit Knoten verbinden](#).

Ein primärer Redis OSS-Endpunkt (Clustermodus deaktiviert) sieht etwa wie folgt aus. Abhängig davon, ob die Verschlüsselung während der Übertragung aktiviert ist oder nicht, unterscheidet sich dies.

Verschlüsselung während der Übertragung nicht aktiviert

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port
```

```
redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Verschlüsselung während der Übertragung aktiviert

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port
```



```
master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Informationen zu den Endpunkten finden Sie im entsprechenden Thema für die Engine und den Cluster-Typ, die von Ihnen ausgeführt werden.

- [Verbindungsendpunkte finden in ElastiCache](#)
- [Finden von Endpunkten für einen Valkey- oder Redis OSS-Cluster \(Clustermodus aktiviert\) \(Konsole\)](#)—Sie benötigen den Konfigurationsendpunkt des Clusters.
- [Suchen von Endpunkten \(AWS CLI\)](#)
- [Endpunkte finden \(API\) ElastiCache](#)

Stellen Sie eine Connect zu einem Valkey- oder Redis-OSS-Cluster oder einer Replikationsgruppe her (Linux)

Da Sie nun den benötigten Endpunkt haben, können Sie sich bei einer EC2 Instanz anmelden und eine Verbindung zum Cluster oder zur Replikationsgruppe herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Die neueste Version von `valkey-cli` unterstützt auch die Verbindung aktivierter Cluster. SSL/TLS encryption/authentication

Im folgenden Beispiel werden EC2 Amazon-Instances verwendet, auf denen Amazon Linux und Amazon Linux 2 ausgeführt werden. Einzelheiten zur Installation und Kompilierung von Valkey-Cli mit anderen Linux-Distributionen finden Sie in der Dokumentation für Ihr spezielles Betriebssystem.

Note

Dieser Prozess umfasst das Testen einer Verbindung mit dem Valkey-Cli-Hilfsprogramm nur für den ungeplanten Gebrauch. [Eine Liste der unterstützten Clients finden Sie in der Valkey-Dokumentation](#). Beispiele für die Verwendung von AWS SDKs with finden Sie ElastiCache unter [Tutorials: Erste Schritte mit Python und ElastiCache](#).

Herstellen einer Verbindung zu einem unverschlüsselten Cluster mit deaktiviertem Cluster-Modus

1. Führen Sie den folgenden Befehl aus, um eine Verbindung zum Cluster herzustellen *primary-endpoint* und *port number* durch den Endpunkt Ihres Clusters und Ihre Portnummer zu ersetzen. (Der Standardport für Valkey und Redis OSS ist 6379.)

```
src/valkey-cli -h primary-endpoint -p port number
```

Das Ergebnis in einer Befehlszeile sieht in etwa wie folgt aus:

```
primary-endpoint:port number
```

2. Sie können jetzt die OSS-Befehle Valkey und Redis ausführen.

```
set x Hello
OK

get x
"Hello"
```

Herstellen einer Verbindung mit einem unverschlüsselten Cluster mit aktiviertem Cluster-Modus

1. Führen Sie den folgenden Befehl aus, um eine Verbindung zum Cluster herzustellen *configuration-endpoint* und ihn *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer zu ersetzen. (Der Standardport für Valkey und Redis OSS ist 6379.)

```
src/valkey-cli -h configuration-endpoint -c -p port number
```

 Note

Im vorherigen Befehl aktiviert Option -c den Cluster-Modus nach den [-ASK und -MOVED-Umleitungen](#).

Das Ergebnis in einer Befehlszeile sieht in etwa wie folgt aus:

```
configuration-endpoint:port number
```

2. Sie können jetzt die OSS-Befehle Valkey und Redis ausführen. Beachten Sie, dass die Umleitung erfolgt, weil Sie diese mit der Option -c aktiviert haben. Wenn die Umleitung nicht aktiviert ist, gibt der Befehl die Fehlermeldung MOVED zurück. Weitere Informationen zum MOVED-Fehler finden Sie in der [Clusterspezifikation](#).

```
set x Hi
-> Redirected to slot [16287] located at 172.31.28.122:6379
OK
set y Hello
OK
get y
"Hello"
set z Bye
-> Redirected to slot [8157] located at 172.31.9.201:6379
OK
get z
"Bye"
get x
-> Redirected to slot [16287] located at 172.31.28.122:6379
"Hi"
```

Verbindung zu einem Encryption/Authentication aktivierten Cluster herstellen

Standardmäßig verwendet valkey-cli eine unverschlüsselte TCP-Verbindung, wenn eine Verbindung zu Valkey und Redis OSS hergestellt wird. Die Option wird zum SSL/TLS Zeitpunkt der Valkey-CLI-Kompilierung BUILD_TLS=yes aktiviert, wie im vorherigen Abschnitt gezeigt. [Laden Sie den Befehlszeilenzugriff herunter und richten Sie ihn ein](#) Die Aktivierung von AUTH ist optional. Sie müssen jedoch Verschlüsselung während der Übertragung aktivieren, um AUTH zu aktivieren. Weitere Informationen zur ElastiCache Verschlüsselung und Authentifizierung finden Sie unter [ElastiCache Verschlüsselung bei der Übertragung \(TLS\)](#)

Note

Sie können die Option `--tls` mit valkey-cli verwenden, um eine Verbindung zu verschlüsselten Clustern herzustellen, die im Clustermodus aktiviert oder deaktiviert sind. Wenn für einen Cluster ein AUTH-Token festgelegt ist, können Sie die Option `-a` verwenden, um ein AUTH-Passwort anzugeben.

Stellen Sie in den folgenden Beispielen sicher, dass Sie *cluster-endpoint* und *port number* durch den Endpunkt Ihres Clusters und Ihre Portnummer ersetzen. (Der Standardport für Redis OSS ist 6379.)

Verbindung zu verschlüsselten Clustern mit Cluster-Modus deaktiviert

Das folgende Beispiel verbindet zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung:

```
src/valkey-cli -h cluster-endpoint --tls -a your-password -p port number
```

Das folgende Beispiel verbindet zu einem Cluster nur mit aktivierter Verschlüsselung:

```
src/valkey-cli -h cluster-endpoint --tls -p port number
```

Verbindung zu verschlüsselten Clustern mit Cluster-Modus aktiviert

Das folgende Beispiel verbindet zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung:

```
src/valkey-cli -c -h cluster-endpoint --tls -a your-password -p port number
```

Das folgende Beispiel verbindet zu einem Cluster nur mit aktivierter Verschlüsselung:

```
src/valkey-cli -c -h cluster-endpoint --tls -p port number
```

Nachdem Sie eine Verbindung zum Cluster hergestellt haben, können Sie die Befehle Valkey oder Redis OSS ausführen, wie in den vorherigen Beispielen für unverschlüsselte Cluster gezeigt.

Eine Alternative zu valkey-cli oder Redis-CLI

Wenn für den Cluster der Clustermodus nicht aktiviert ist und Sie für einen kurzen Test eine Verbindung zum Cluster herstellen müssen, ohne die Valkey-CLI- oder Redis-CLI-Kompilierung durchzuführen, können Sie Telnet oder OpenSSL verwenden. Stellen Sie sicher, dass Sie in den folgenden Beispielbefehlen *cluster-endpoint* und durch den Endpunkt Ihres Clusters und *port number* Ihre Portnummer ersetzen. (Der Standardport für Redis OSS ist 6379.)

Das folgende Beispiel stellt eine Verbindung zu einem deaktivierten Cluster im Clustermodus mit aktivierter Verschlüsselung und and/or Authentifizierung her:

```
openssl s_client -connect cluster-endpoint:port number
```

Wenn für den Cluster ein Passwort festgelegt ist, stellen Sie zunächst eine Verbindung mit dem Cluster her. Nachdem Sie eine Verbindung hergestellt haben, authentifizieren Sie den Cluster mit

dem folgenden Befehl, und drücken Sie dann die Enter-Taste. Im folgenden Beispiel ersetzen Sie es *your-password* durch das Passwort für Ihren Cluster.

```
Auth your-password
```

Im folgenden Beispiel wird eine Verbindung zu einem deaktivierten Cluster im Cluster-Modus mit Verschlüsselung und / oder Authentifizierung hergestellt:

```
telnet cluster-endpoint port number
```

Stellen Sie eine Connect zu einem Valkey- oder Redis-OSS-Cluster oder einer Replikationsgruppe her (Windows)

Um von einer EC2 Windows-Instanz aus mithilfe der Valkey- oder Redis OSS-CLI eine Verbindung zum Cluster herzustellen, müssen Sie das valkey-cli-Paket herunterladen und valkey-cli.exe verwenden, um von einer Windows-Instanz aus eine Verbindung zum Valkey- oder Redis-OSS-Cluster herzustellen. EC2

Im folgenden Beispiel verwenden Sie das Valkey-CLI-Hilfsprogramm, um eine Verbindung zu einem Cluster herzustellen, der nicht verschlüsselungsfähig ist und auf dem Valkey oder Redis OSS ausgeführt wird. Weitere Informationen zu Valkey und verfügbaren Befehlen finden Sie unter Valkey-Befehle auf der [Valkey-Website](#).

So stellen Sie mithilfe von Valkey-CLI eine Verbindung zu einem Valkey- oder Redis-OSS-Cluster her, der nicht verschlüsselungsfähig ist

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her. Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2 Erste Schritte“](#).
2. Kopieren Sie den Link und fügen Sie ihn <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> in einen Internetbrowser ein, um die ZIP-Datei für den Redis OSS-Client aus der verfügbaren Version unter herunterzuladen GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Extrahieren Sie die ZIP-Datei in den gewünschten Ordner/Pfad.

Öffnen Sie die Befehlszeile, wechseln Sie in das Valkey-Verzeichnis und führen Sie den Befehl aus. c:\Valkey>valkey-cli -h *Valkey_Cluster_Endpoint* -p 6379

Beispiel:

```
c:\Valkey>valkey-cli -h cmd.xxxxxxx.ng.0001.usw2.cache.amazonaws.com -p 6379
```

3. Führen Sie die Befehle Valkey oder Redis OSS aus.

Sie sind jetzt mit dem Cluster verbunden und können Valkey- oder Redis OSS-Befehle wie die folgenden ausführen.

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5   // Set key "b" with a string value and a 5 second expiration
"Good-bye"
get b                   // Get value for key "b"
"Good-bye"

                        // wait >= 5 seconds

get b
(nil)                   // key has expired, nothing returned
quit                   // Exit from valkey-cli
```

Löschen eines Clusters

Solange sich ein Cluster im Zustand `available` befindet, werden Ihnen dafür Gebühren berechnet. Dabei spielt es keine Rolle, ob Sie ihn aktiv nutzen oder ob nicht. Löschen Sie den Cluster, damit keine Gebühren mehr anfallen.

Warning

- Wenn Sie einen ElastiCache Cluster löschen, werden Ihre manuellen Snapshots beibehalten. Sie können auch einen letzten Snapshot erstellen, bevor der Cluster gelöscht wird. Automatisch erstellte Cache-Snapshots werden nicht aufbewahrt. Weitere Informationen finden Sie unter [Snapshot und Wiederherstellung](#).

- **CreateSnapshot**Für die Erstellung eines endgültigen Snapshots ist eine Genehmigung erforderlich. Ohne diese Genehmigung schlägt der API-Aufruf mit einer Access Denied Ausnahme fehl.

Mit dem AWS-Managementkonsole

Mit dem folgenden Verfahren wird ein einzelner Cluster aus Ihrer Bereitstellung gelöscht. Um mehrere Cache-Cluster zu löschen, wiederholen Sie das Verfahren für jeden Cluster, den Sie löschen möchten. Sie brauchen nicht zu warten, bis ein Cluster fertig gelöscht ist, bevor Sie den Vorgang zum Löschen eines anderen Clusters starten.

So löschen Sie einen Cluster

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Engine-Dashboard Valkey oder Redis OSS aus.

Eine Liste aller Caches, die auf dieser Engine laufen, wird angezeigt.

3. Wählen Sie aus der Liste den Namen des zu löschenden Clusters aus. In diesem Fall der Name des von Ihnen unter [Schritt 2: Erstellen eines Clusters](#) erstellten Clusters.

Important

Sie können jeweils nur einen Cluster von der ElastiCache Konsole löschen. Werden mehrere Cluster ausgewählt, wird die Löschoperation deaktiviert.

4. Klicken Sie bei Actions auf Delete.
5. Geben Sie im Bestätigungsbildschirm Cluster löschen den Namen des Clusters ein und wählen Sie Endgültiges Backup aus. Wählen Sie dann Löschen, um den Cluster zu löschen, oder Abbrechen aus, wenn Sie den Cluster beibehalten möchten.

Wenn Sie Delete auswählen, ändert sich der Status des Clusters zu deleting.

Sobald dieser Cluster nicht mehr in der Cluster-Liste erscheint, fallen dafür keine Gebühren mehr an.

Mit dem AWS CLI

Der folgende Code löscht den Cluster `my-cluster`. In diesem Fall ersetzen Sie `my-cluster` mit dem Namen des Clusters, den Sie unter [Schritt 2: Erstellen eines Clusters](#) erstellt haben.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

Die `delete-cache-cluster` CLI-Aktion löscht nur einen Cluster. Um mehrere Cluster zu löschen, rufen Sie `delete-cache-cluster` jeden Cache-Cluster auf, den Sie löschen möchten. Sie müssen nicht warten, bis ein Cluster vollständig gelöscht ist, bevor Sie einen anderen löschen.

Für Linux, macOS oder Unix:

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --region us-east-2
```

Für Windows:

```
aws elasticache delete-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --region us-east-2
```

Weitere Informationen finden Sie im AWS CLI ElastiCache For-Thema [delete-cache-cluster](#).

Weitere ElastiCache Tutorials und Videos

Die folgenden Tutorials befassen sich mit Aufgaben, die für den ElastiCache Amazon-Benutzer von Interesse sind.

- [ElastiCache Videos](#)
- [Tutorial: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon VPC](#)

ElastiCache Videos

Hier finden Sie Videos, die Ihnen helfen, grundlegende und fortgeschrittene ElastiCache Amazon-Konzepte zu erlernen. Informationen zu AWS Schulungen finden Sie unter [AWS Schulung und Zertifizierung](#).

Themen

- [Einführungsvideos](#)
- [Weiterführende Videos](#)

Einführungsvideos

Die folgenden Videos stellen Ihnen Amazon vor ElastiCache.

Themen

- [AWS re:Invent 2020: Was ist neu bei Amazon ElastiCache](#)
- [AWS re:Invent 2019: Was ist neu bei Amazon ElastiCache](#)
- [AWS re:Invent 2017: Was ist neu bei Amazon ElastiCache](#)
- [DAT204 — Entwicklung skalierbarer Anwendungen auf AWS NoSQL-Diensten \(re:Invent 2015\)](#)
- [DAT207 — Schnellere Anwendungsleistung mit Amazon ElastiCache \(re:Invent 2013\)AWS](#)

AWS re:Invent 2020: Was ist neu bei Amazon ElastiCache

[AWS re:Invent 2020: Was ist neu bei Amazon ElastiCache](#)

AWS re:Invent 2019: Was ist neu bei Amazon ElastiCache

[AWS re:Invent 2019: Was ist neu bei Amazon ElastiCache](#)

AWS re:Invent 2017: Was ist neu bei Amazon ElastiCache

[AWS re:Invent 2017: Was ist neu bei Amazon ElastiCache](#)

DAT204 — Entwicklung skalierbarer Anwendungen auf AWS NoSQL-Diensten (re:Invent 2015)

In dieser Sitzung besprechen wir die Vorteile von NoSQL-Datenbanken und geben einen Überblick über die wichtigsten NoSQL-Services, die von AWS Amazon DynamoDB und Amazon angeboten

werden. ElastiCache Dann hören wir von zwei führenden Kunden, Expedia und Mapbox, über ihre Anwendungsfälle und architektonischen Herausforderungen und darüber, wie sie diese mithilfe von AWS NoSQL-Diensten bewältigt haben, einschließlich Entwurfsmustern und Best Practices. Am Ende der Sitzung werden Sie NoSQL und die leistungsstarken Fähigkeiten besser verstehen. Sie werden sich den Herausforderungen, die eine Datenbank darstellen kann, zuversichtlich gewachsen fühlen.

[DAT204 — Entwicklung skalierbarer Anwendungen auf AWS NoSQL-Diensten \(re:Invent 2015\)](#)

[DAT207 — Schnellere Anwendungsleistung mit Amazon ElastiCache \(re:Invent 2013\)](#)

In diesem Video erfahren Sie, wie Sie Amazon verwenden können, ElastiCache um auf einfache Weise ein In-Memory-Caching-System bereitzustellen, um die Leistung Ihrer Anwendung zu beschleunigen. Wir zeigen Ihnen, wie Sie Amazon verwenden können ElastiCache , um die Latenz Ihrer Anwendungen zu verbessern und die Belastung Ihrer Datenbankserver zu reduzieren. Wir zeigen Ihnen auch, wie Sie eine Caching-Schicht aufbauen, die einfach zu verwalten und zu skalieren ist, wenn Ihre Anwendung wächst. In dieser Sitzung gehen wir auf verschiedene Szenarien und Anwendungsfälle ein, die von der Aktivierung von Caching profitieren können, und besprechen die von Amazon ElastiCache bereitgestellten Funktionen.

[DAT207 — Schnellere Anwendungsleistung mit Amazon ElastiCache \(re:Invent 2013\)](#)

Weiterführende Videos

Die folgenden Videos behandeln fortgeschrittenere ElastiCache Amazon-Themen.

Themen

- [Erfolgreiches Design mit den ElastiCache Best Practices von Amazon \(re:Invent 2020\)](#)
- [Optimieren Sie Ihre Echtzeit-Apps mit Amazon ElastiCache \(re:Invent 2019\)](#)
- [Bewährte Methoden: Migration von Redis OSS-Clustern von Amazon EC2 zu ElastiCache \(re:Invent 2019\)](#)
- [Skalierung einer Fantasy-Sport-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 \(re:Invent 2018\)](#)
- [Zuverlässiges und skalierbares Redis OSS in der Cloud mit Amazon ElastiCache \(re:Invent 2018\)](#)
- [ElastiCache Deep Dive: Entwurfsmuster für In-Memory-Datenspeicher \(re:Invent 2018\)](#)
- [DAT305 — Amazon ElastiCache Deep Dive \(re:Invent 2017\)](#)
- [DAT306 — Amazon ElastiCache Deep Dive \(re:Invent 2016\)](#)

- [DAT317—Wie IFTTT Redis OSS ElastiCache zur Vorhersage von Ereignissen nutzt \(re:Invent 2016\)](#)
- [DAT407 — Amazon ElastiCache Deep Dive \(re:Invent 2015\)](#)
- [SDD402 — Amazon ElastiCache Deep Dive \(re:Invent 2014\)](#)
- [DAT307 — Tiefer Einblick in die ElastiCache Architektur und Entwurfsmuster von Amazon \(re:Invent 2013\)](#)

Erfolgreiches Design mit den ElastiCache Best Practices von Amazon (re:Invent 2020)

Angesichts des explosionsartigen Wachstums geschäftskritischer Echtzeitanwendungen, die auf Redis OSS basieren, sind Verfügbarkeit, Skalierbarkeit und Sicherheit zu den wichtigsten Überlegungen geworden. Lernen Sie bewährte Methoden kennen, um Amazon ElastiCache mit Online-Skalierung, Hochverfügbarkeit in Multi-AZ-Bereitstellungen und Sicherheitskonfigurationen auf Erfolgskurs zu bringen.

[Erfolgreiches Design mit den ElastiCache Best Practices von Amazon \(re:Invent 2020\)](#)

Optimieren Sie Ihre Echtzeit-Apps mit Amazon ElastiCache (re:Invent 2019)

Angesichts der rasanten Zunahme der Cloud-Nutzung und der damit verbundenen neuen Szenarien benötigen Anwendungen Latenzzeiten im Mikrosekundenbereich und einen hohen Durchsatz, um Millionen von Anfragen pro Sekunde zu unterstützen. Für die Verwaltung von Daten für Echtzeitanwendungen haben sich die Entwickler bisher auf spezielle Hardware und Workarounds verlassen, z. B. auf plattenbasierte Datenbanken in Kombination mit Datenreduzierungstechniken. Diese Vorgehensweisen können kostspielig und nicht erweiterungsfähig sein. Erfahren Sie, wie Sie die Leistung von Echtzeitanwendungen steigern können, indem Sie das vollständig verwaltete In-Memory-Amaon ElastiCache für extreme Leistung, hohe Skalierbarkeit, Verfügbarkeit und Sicherheit nutzen.

[Optimieren Sie Ihre Echtzeit-Apps mit Amazon ElastiCache \(re:Invent 2019\)](#)

Bewährte Methoden: Migration von Redis OSS-Clustern von Amazon EC2 zu ElastiCache (re:Invent 2019)

Es kann schwierig sein, Redis OSS-Cluster selbst zu verwalten. Sie müssen Hardware bereitstellen, Software patchen, Daten sichern und Workloads kontinuierlich überwachen. Mit der neu veröffentlichten Online-Migrationsfunktion für Amazon ElastiCache können Sie Ihre Daten jetzt ganz einfach von selbst gehostetem Redis OSS auf Amazon auf vollständig verwaltetes Amazon EC2

verschieben ElastiCache, wobei der Clustermodus deaktiviert ist. In dieser Sitzung erfahren Sie mehr über das neue Online-Migrationstool, sehen sich eine Demo an und, was noch wichtiger ist, lernen Sie praktische Best Practices für eine reibungslose Migration zu Amazon ElastiCache kennen.

[Bewährte Methoden: Migration von Redis OSS-Clustern von Amazon EC2 zu ElastiCache \(re:Invent 2019\)](#)

Skalierung einer Fantasy-Sport-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 (re:Invent 2018)

Dream11 ist Indiens führendes Sport-Tech-Startup. Es hat eine wachsende Basis von mehr als 40 Millionen Nutzern, die mehrere Sportarten spielen, darunter Fantasy Cricket, Fußball und Basketball, und bedient derzeit eine Million gleichzeitige Nutzer, die drei Millionen Anfragen pro Minute mit einer Reaktionszeit von 50 Millisekunden stellen. In diesem Vortrag erklärt Amit Sharma, CTO von Dream11, wie das Unternehmen Amazon Aurora und Amazon für den Umgang mit Flash-Verkehr verwendet, der ElastiCache sich innerhalb eines Antwortfensters von 30 Sekunden verdreifachen kann. Sharma spricht auch über die Skalierung von Transaktionen ohne Sperren und erklärt, wie man den Flash-Verkehr bewältigt und dabei fünf Millionen täglich aktive Nutzer bedient. Vollständiger Titel: AWS re:Invent 2018: Skalierung einer Fantasy-Sportplattform mit Amazon ElastiCache und Amazon Aurora () STP11

[Skalierung einer Fantasy-Sport-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 \(re:Invent 2018\)](#)

Zuverlässiges und skalierbares Redis OSS in der Cloud mit Amazon ElastiCache (re:Invent 2018)

In dieser Sitzung werden die Funktionen und Verbesserungen unseres Redis OSS-kompatiblen Service Amazon ElastiCache for Redis OSS behandelt. Wir behandeln wichtige Funktionen wie Redis OSS 5, Skalierbarkeits- und Leistungsverbesserungen, Sicherheit und Compliance und vieles mehr. Wir besprechen auch zukünftige Funktionen und Kundenfallstudien.

[Zuverlässiges und skalierbares Redis OSS in der Cloud mit Amazon ElastiCache \(re:Invent 2018\)](#)

ElastiCache Deep Dive: Entwurfsmuster für In-Memory-Datenspeicher (re:Invent 2018)

In dieser Sitzung bieten wir einen Blick hinter die Kulissen, um mehr über das Design und die Architektur von Amazon zu erfahren ElastiCache. Sehen Sie sich die gängigen Entwurfsmuster unserer Redis OSS- und Memcached-Angebote an und erfahren Sie, wie Kunden sie für die In-Memory-Datenverarbeitung verwenden, um die Latenz zu reduzieren und den Anwendungsdurchsatz zu verbessern. Wir überprüfen ElastiCache bewährte Verfahren, Entwurfsmuster und Anti-Patterns.

[ElastiCache Deep Dive: Entwurfsmuster für speicherinterne Datenspeicher \(re:Invent 2018\)](#)

DAT305 — Amazon ElastiCache Deep Dive (re:Invent 2017)

Schauen Sie hinter die Kulissen, um mehr über ElastiCache das Design und die Architektur von Amazon zu erfahren. Sehen Sie sich gängige Entwurfsmuster unserer Memcached- und Redis OSS-Angebote an und erfahren Sie, wie Kunden sie für In-Memory-Operationen eingesetzt haben, um die Latenz zu reduzieren und den Anwendungsdurchsatz zu verbessern. In diesem Video gehen wir auf ElastiCache bewährte Verfahren, Entwurfsmuster und Anti-Patterns ein.

Das Video zeigt Folgendes:

- ElastiCache für Redis OSS Online-Resharding
- ElastiCache Sicherheit und Verschlüsselung
- ElastiCache für Redis OSS Version 3.2.10

[DAT305 — Amazon ElastiCache Deep Dive \(re:Invent 2017\)](#)

DAT306 — Amazon ElastiCache Deep Dive (re:Invent 2016)

Schauen Sie hinter die Kulissen, um mehr über ElastiCache das Design und die Architektur von Amazon zu erfahren. Sehen Sie sich gängige Entwurfsmuster unserer Memcached- und Redis OSS-Angebote an und erfahren Sie, wie Kunden sie für In-Memory-Operationen eingesetzt haben, um die Latenz zu reduzieren und den Anwendungsdurchsatz zu verbessern. In dieser Sitzung besprechen wir ElastiCache bewährte Verfahren, Entwurfsmuster und Anti-Patterns.

[DAT306 — Amazon ElastiCache Deep Dive \(re:Invent 2016\)](#)

DAT317—Wie IFTTT Redis OSS ElastiCache zur Vorhersage von Ereignissen nutzt (re:Invent 2016)

IFTTT ist ein kostenloser Service, mit dem Kunden die Services, die sie am liebsten verwenden, noch besser nutzen können. Das reicht von der Automatisierung einfacher Aufgaben bis hin zur Interaktion mit und Steuerung von Wohnungen und Häusern. IFTTT verwendet ElastiCache für Redis OSS, um den Verlauf von Transaktionsläufen und Zeitplanprognosen sowie Indizes für Protokolldokumente auf Amazon S3 zu speichern. Sehen Sie sich diese Sitzung an, um zu erfahren, wie die Skripting-Leistung von Lua und die Datentypen von Redis OSS es Menschen ermöglicht haben, etwas zu erreichen, das sie anderswo nicht hätten erreichen können.

[DAT317—Wie IFTTT Redis OSS ElastiCache zur Vorhersage von Ereignissen einsetzt \(re:Invent 2016\)](#)

DAT407 — Amazon ElastiCache Deep Dive (re:Invent 2015)

Werfen Sie einen Blick hinter die Kulissen, um mehr über ElastiCache das Design und die Architektur von Amazon zu erfahren. Sehen Sie sich die gängigen Entwurfsmuster unserer Memcached- und Redis OSS-Angebote an und erfahren Sie, wie Kunden sie für In-Memory-Operationen eingesetzt und eine verbesserte Latenz und einen verbesserten Durchsatz für Anwendungen erzielt haben. In dieser Sitzung besprechen wir bewährte Verfahren, Entwurfsmuster und Anti-Pattern-Methoden im Zusammenhang mit Amazon ElastiCache.

[DAT407 — Amazon ElastiCache Deep Dive \(re:Invent 2015\)](#)

SDD402 — Amazon ElastiCache Deep Dive (re:Invent 2014)

In diesem Video untersuchen wir gängige Caching-Anwendungsfälle, die ElastiCache für Memcached- und Redis OSS-Engines, Muster, anhand derer Sie ermitteln können, welche Engine ElastiCache für Ihre Bedürfnisse besser geeignet ist, konsistentes Hashing und vieles mehr, um schnelle, skalierbare Anwendungen zu entwickeln. Frank Wiebe, Principal Scientist bei Adobe, beschreibt, wie Adobe Amazon nutzt, ElastiCache um das Kundenerlebnis zu verbessern und sein Geschäft zu skalieren.

[DAT402 — Amazon ElastiCache Deep Dive \(re:Invent 2014\)](#)

DAT307 — Tiefer Einblick in die ElastiCache Architektur und Entwurfsmuster von Amazon (re:Invent 2013)

In diesem Video untersuchen wir Caching, Caching-Strategien, Skalierung und Überwachung. Wir vergleichen auch die OSS-Engines ElastiCache für Memcached und für Redis. ElastiCache Während dieser Sitzung besprechen wir auch Best Practices und Designmuster im Zusammenhang mit Amazon ElastiCache.

[DAT307 — Tiefer Einblick in die ElastiCache Architektur und Designmuster von Amazon \(AWS re:Invent 2013\).](#)

Knoten verwalten in ElastiCache

Ein Knoten ist der kleinste Baustein einer ElastiCache Amazon-Bereitstellung. Er ist ein RAM-Bereich fester Größe, der sicher ins Netzwerk eingebunden ist. Auf jedem Knoten wird die Engine ausgeführt, je nachdem, welche Option beim Erstellen oder der letzten Änderung des Clusters ausgewählt wurde. Jeder Knoten besitzt einen eigenen Domain Name Service (DNS)-Namen und

Port. Es werden mehrere ElastiCache Knotentypen unterstützt, die jeweils unterschiedliche Mengen an Arbeitsspeicher und Rechenleistung aufweisen.

Eine ausführliche Erörterung der zu verwendenden Knotengröße finden Sie unter [Auswahl der Knotengröße](#).

Im Allgemeinen verfügen Valkey- oder Redis OSS-Bereitstellungen (Cluster-Modus aktiviert) aufgrund der Unterstützung von Sharding über eine Reihe kleinerer Knoten. Im Gegensatz dazu haben Valkey- oder Redis OSS-Bereitstellungen (Clustermodus deaktiviert) weniger, dafür aber größere Knoten in einem Cluster. Eine ausführliche Erörterung der zu verwendenden Knotengröße finden Sie unter [Auswahl der Knotengröße](#).

Themen

- [ElastiCache Knotenstatus anzeigen](#)
- [Valkey- oder Redis-OSS-Knoten und -Shards](#)
- [Mit Knoten verbinden](#)
- [Unterstützte Knotentypen](#)
- [Knoten neu starten](#)
- [Knoten ersetzen \(Valkey und Redis OSS\)](#)
- [Knoten ersetzen \(Memcached\)](#)
- [Reservierte Knoten](#)
- [Migration von Knoten der vorherigen Generation](#)

Einige wichtige Operationen, die Knoten umfassen, sind folgende:

- [Hinzufügen von Knoten zu einem ElastiCache Cluster](#)
- [Knoten aus einem ElastiCache Cluster entfernen](#)
- [Skalierung ElastiCache](#)
- [Verbindungsendpunkte finden in ElastiCache](#)
- [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#)

ElastiCache Knotenstatus anzeigen

Über die [ElastiCache Konsole](#) können Sie schnell auf den Status Ihres ElastiCache Knotens zugreifen. Der Status eines ElastiCache Knotens gibt den Zustand des Knotens an. Sie können die

folgenden Verfahren verwenden, um den ElastiCache Knotenstatus in der ElastiCache Amazon-Konsole, im AWS CLI Befehl oder im API-Vorgang anzuzeigen.

Die möglichen Statuswerte für ElastiCache Knoten sind in der folgenden Tabelle aufgeführt. In dieser Tabelle ist auch angegeben, ob Ihnen der ElastiCache Knoten in Rechnung gestellt wird.

Typ	Berechnet	Beschreibung
available	Berechnet	Der ElastiCache Knoten ist fehlerfrei und verfügbar.
creating	Nicht berechnet	Der ElastiCache Knoten wird gerade erstellt. Während der Knoten erstellt wird, kann nicht auf ihn zugegriffen werden.
deleting	Nicht berechnet	Der ElastiCache Knoten wird gelöscht.
modifying	Berechnet	Der ElastiCache Knoten wird aufgrund einer Kundenanfrage zur Änderung des Knotens geändert.
updating	Berechnet	Ein Aktualisierungsstatus gibt an, dass eine oder mehrere der folgenden Bedingungen für den ElastiCache Amazon-Knoten zutreffen: • Der ElastiCache Knoten wird im Rahmen des Service-Updates gepatcht. Weitere Informationen zu den Service-Updates finden Sie auf der Hilfeseite Amazon ElastiCache Managed Maintenance and Service Updates.

Typ	Berechnet	Beschreibung
		• Die VPC-Sicherheitsgruppen werden für den ElastiCache Cluster aktualisiert. • Der ElastiCache Cluster wird hoch- oder herunterskaliert. • Die Konfigurationen für die Protokollzustellung werden für den ElastiCache Cluster geändert. • Ein Löschvorgang für den ElastiCache Knoten steht noch aus. • Das ElastiCache Passwort wird updated/rotated verwendet AWS Secrets Manager.
rebooting cluster nodes	Berechnet	Der ElastiCache Knoten wird aufgrund einer Kundenanfrage oder eines ElastiCache Amazon-Prozesses, der einen Neustart des Knotens erfordert , neu gestartet.

Typ	Berechnet	Beschreibung
<code>incompatible_parameters</code>	Nicht berechnet	Amazon ElastiCache kann den Knoten nicht starten, da die in der Parametergruppe des Knotens angegebenen Parameter nicht mit dem Knoten kompatibel sind. Machen Sie entweder die letzten Änderungen an den Parametern rückgängig oder stellen Sie ihre Kompatibilität mit dem Knoten her, um wieder auf den Knoten zugreifen zu können. Weitere Informationen zu den inkompatiblen Parametern finden Sie in der Ereignisliste für den ElastiCache Knoten.
<code>incompatible_network</code>	Nicht berechnet	Ein inkompatibler Netzwerkstatus bedeutet, dass eine oder mehrere der folgenden Bedingungen auf den Amazon-Knoten zutreffen: ElastiCache • In dem Subnetz, in dem der ElastiCache Knoten gestartet wurde, sind keine IP-Adressen verfügbar. • Das in der Subnetzgruppe erwähnte ElastiCache Subnetz ist in der Amazon Virtual Private Cloud (Amazon VPC) nicht mehr vorhanden.

Typ	Berechnet	Beschreibung
restore_failed	Nicht berechnet	Ein Status „Wiederherstellung fehlgeschlagen“ bedeutet, dass einer der folgenden Punkte auf den Amazon-Knoten zutrifft: ElastiCache • Der Austausch des Knotens schlug wiederholt fehl, da die Instance-Kapazität nicht ausreichte. Dies ist in der Regel der Fall, wenn Knoten der vorherigen Generation ausgeführt werden. end-of-life Dies kann jedoch auch passieren, wenn Knoten der aktuellen Generation ersetzt werden, wenn AWS nicht genügend On-Demand-Kapazität zur Erfüllung Ihrer Anfrage in der angegebenen Availability Zone zur Verfügung steht. Weitere Informationen zum Reparieren oder Entfernen dieser Knoten finden Sie unter Migration von Knoten der vorherigen Generation. • Der angegebene RDB-Snapshot konnte nicht wiederhergestellt werden. • Das AWS Konto für den ElastiCache Cluster wurde gesperrt.

Typ	Berechnet	Beschreibung
		Der Knoten ist ausgefallen und konnte nicht wiederhergestellt werden.
snapshotting	Berechnet	ElastiCache erstellt einen Snapshot des Valkey- oder Redis-OSS-Knotens.

Den ElastiCache Knotenstatus mit der Konsole anzeigen

So zeigen Sie den Status eines ElastiCache Nodes mit der Konsole an:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis OSS Clusters oder Memcached Clusters aus. Die Seite Caches mit der Liste der Knoten wird angezeigt. ElastiCache Für die einzelnen Knoten wird der Statuswert angezeigt.
3. Sie können dann zur Registerkarte Service Updates für den Cache navigieren, um die Liste der für den Cache geltenden Service-Updates anzuzeigen.

Den ElastiCache Knotenstatus anzeigen mit dem AWS CLI

Verwenden Sie den `describe-cache-cluster` Befehl, um den ElastiCache Knoten und seine Statusinformationen mit dem AWS CLI anzuzeigen. Der folgende AWS CLI Befehl zeigt beispielsweise jeden ElastiCache Knoten an.

```
aws elasticache describe-cache-clusters
```

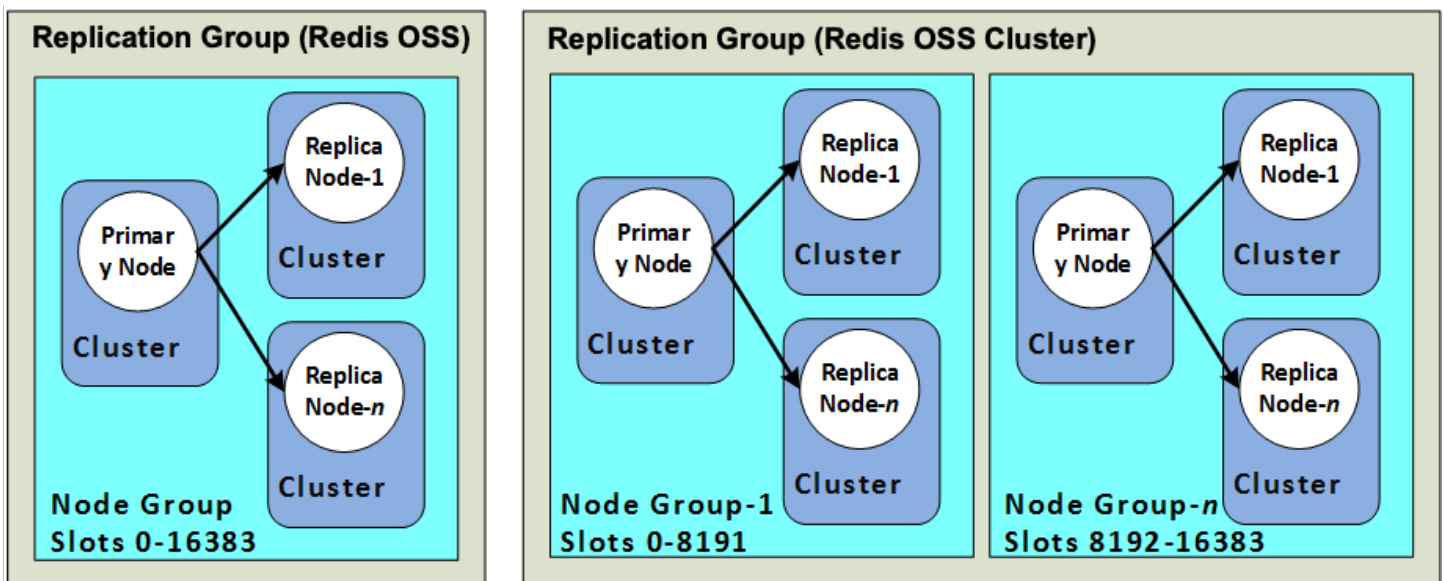
Den ElastiCache Knotenstatus über die API anzeigen

Um den Status des ElastiCache Nodes mithilfe der ElastiCache Amazon-API einzusehen, rufen Sie die `DescribeCacheClusteroperation` mit dem `ShowCacheNodeInfo` Flag auf, um Informationen über die einzelnen Cache-Knoten abzurufen.

Valkey- oder Redis-OSS-Knoten und -Shards

Ein Shard (in der API und CLI: Knotengruppe) ist eine hierarchische Anordnung von Knoten, die jeweils in einem Cluster eingebunden sind. Shards unterstützen die Replikation. Innerhalb eines Shards fungiert ein Knoten als primärer Knoten. Alle anderen Knoten in einem Shard dienen als schreibgeschützte Replikate des Primärknotens. Valkey oder Redis OSS Version 3.2 und höher unterstützen mehrere Shards innerhalb eines Clusters (in der API und CLI, einer Replikationsgruppe). Diese Unterstützung ermöglicht die Partitionierung Ihrer Daten in einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert).

Das folgende Diagramm veranschaulicht die Unterschiede zwischen einem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) und einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert).



Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) unterstützen die Replikation über Shards. Die API-Operation [DescribeReplicationGroups](#) (CLI: [describe-replication-groups](#)) listet die Knotengruppen mit den Mitglieds-knoten, der Rolle des Knotens innerhalb der Knotengruppe und anderen Informationen auf.

Wenn Sie einen Valkey- oder Redis OSS-Cluster erstellen, geben Sie an, ob Sie einen Cluster mit aktiviertem Clustering erstellen möchten. Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) haben nie mehr als einen Shard. Dieser kann horizontal skaliert werden, indem Read Replica-Knoten hinzugefügt (bis zu insgesamt fünf) oder gelöscht werden. Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#), [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#) oder [Löschen einer Read Replica für Valkey oder Redis OSS](#).

[\(Cluster-Modus deaktiviert\)](#). Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) können auch vertikal skaliert werden, indem sie die Knotentypen ändern. Weitere Informationen finden Sie unter [Skalierung von Replikatknoten für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#).

Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn es sich bei der Engine um Valkey oder Redis OSS Version 5.0.6 oder höher handelt. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

Nachdem ein Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) erstellt wurde, kann er geändert (nach innen oder außen skaliert) werden. Weitere Informationen erhalten Sie unter [Skalierung ElastiCache](#) und [Knoten ersetzen \(Valkey und Redis OSS\)](#).

Wenn Sie einen neuen Cluster erstellen, können Sie ihn mit Daten aus dem alten Cluster bestücken, damit er nicht von Anfang an leer ist. Dieser Ansatz funktioniert nur, wenn die Cluster-Gruppe die gleiche Anzahl an Shards hat wie der alte Cluster. Dies kann hilfreich sein, wenn Sie Ihren Knotentyp oder die Engine-Version ändern müssen. Weitere Informationen erhalten Sie unter [Erstellen manueller Backups](#) und [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Mit Knoten verbinden

Verbindung zu Valkey- oder Redis OSS-Knoten herstellen

Bevor Sie versuchen, eine Verbindung zu den Valkey- oder Redis OSS-Knoten in Ihrem Cluster herzustellen, müssen Sie über die Endpunkte für die Knoten verfügen. Informationen zum Ermitteln der Endpunkte finden Sie unter:

- [Finden Sie die Endpunkte eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden von Endpunkten für einen Valkey- oder Redis OSS-Cluster \(Clustermodus aktiviert\) \(Konsole\)](#)
- [Suchen von Endpunkten \(AWS CLI\)](#)
- [Endpunkte finden \(API\) ElastiCache](#)

Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen, auf dem Valkey oder Redis OSS ausgeführt wird.

Note

[Weitere Informationen zu verfügbaren Befehlen finden Sie auf der Commands-Webseite.](#)

So stellen Sie mithilfe der Valkey-CLI eine Verbindung zu einem Valkey- oder Redis-OSS-Cluster her

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her.

Note

Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2 Erste Schritte“](#).

2. Laden Sie zum Entwickeln von `valkey-cli` die GNU Compiler Collection (`gcc`) herunter und installieren Sie diese. Geben Sie an der Befehlszeile Ihrer EC2 Instance den folgenden Befehl ein und geben Sie ihn `y` an der Bestätigungsaufforderung ein.

```
sudo yum install gcc
```

Die Ausgabe sieht in etwa folgendermaßen aus.

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 27 M
Installed size: 53 M
Is this ok [y/N]: y
Downloading Packages:
(1/11): binutils-2.22.52.0.1-10.36.amzn1.x86_64.rpm      | 5.2 MB    00:00
(2/11): cpp46-4.6.3-2.67.amzn1.x86_64.rpm              | 4.8 MB    00:00
(3/11): gcc-4.6.3-3.10.amzn1.noarch.rpm                | 2.8 kB    00:00

...(output omitted)...

Complete!
```

3. Laden Sie das Valkey-CLI-Hilfsprogramm herunter und kompilieren Sie es. Dieses Hilfsprogramm ist in der Valkey-Softwaredistribution enthalten. Geben Sie an der Befehlszeile Ihrer EC2 Instanz die folgenden Befehle ein:

 Note

Führen Sie bei Ubuntu-Systemen `make distclean` aus, bevor Sie `make` ausführen.

```
wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
tar xvzf valkey-8.0.0.tar.gz
cd valkey-8.0.0
make distclean      # ubuntu systems only
make
```


4. Geben Sie an der Befehlszeile Ihrer EC2 Instance den folgenden Befehl ein.

```
src/valkey-cli -c -h mycachecuster.eaogs8.0001.usw2.cache.amazonaws.com -p 6379
```

Eine Valkey- oder Redis OSS-Eingabeaufforderung ähnlich der folgenden wird angezeigt.

```
redis mycachecuster.eaogs8.0001.usw2.cache.amazonaws.com 6379>
```

5. Testen Sie die Verbindung, indem Sie die Befehle Valkey oder Redis OSS ausführen.

Sie sind jetzt mit dem Cluster verbunden und können Valkey- oder Redis OSS-Befehle ausführen. Im Folgenden finden Sie einige Beispielfehle mit ihren Valkey- oder Redis OSS-Antworten.

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5   // Set key "b" with a string value and a 5 second expiration
get b
"Good-bye"

                        // wait 5 seconds
get b
(nil)                   // key has expired, nothing returned
quit                   // Exit from valkey-cli
```

Zum Herstellen der Verbindung zu Knoten oder Clustern mit SSL (Secure Sockets Layer)-Verschlüsselung (Verschlüsselung während der Übertragung aktiviert) siehe [ElastiCache Verschlüsselung bei der Übertragung \(TLS\)](#).

Verbindung zu Memcached-Knoten herstellen

Bevor Sie versuchen, eine Verbindung mit Ihrem Memcached-Cluster herzustellen, benötigen Sie die Endpunkte für die Knoten. Informationen zum Ermitteln der Endpunkte finden Sie unter:

- [Die Endpunkte eines Clusters finden \(Konsole\) \(Memcached\)](#)
- [Suchen von Endpunkten \(AWS CLI\)](#)

- [Endpunkte finden \(API\) ElastiCache](#)

Im folgenden Beispiel verwenden Sie das telnet-Dienstprogramm, um eine Verbindung mit einem Knoten herzustellen, der Memcached ausführt.

 Note

Weitere Informationen über Memcached und verfügbare Memcached-Befehle finden Sie auf der [Memcached](#)-Website.

So stellen Sie eine Verbindung mit einem Knoten mithilfe von telnet her

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her.

 Note

Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2 Erste Schritte“](#).

2. Laden Sie das Telnet-Hilfsprogramm herunter und installieren Sie es auf Ihrer EC2 Amazon-Instance. Geben Sie an der Befehlszeile Ihrer EC2 Amazon-Instance den folgenden Befehl und an der Befehlszeile y ein.

```
sudo yum install telnet
```

Die Ausgabe sieht in etwa folgendermaßen aus.

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
```

```
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                | 63 kB      00:00

...(output omitted)...

Complete!
```

3. Geben Sie an der Befehlszeile Ihrer EC2 Amazon-Instance den folgenden Befehl ein und ersetzen Sie den in diesem Beispiel gezeigten durch den Endpunkt Ihres Knotens.

```
telnet mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 11211
```

Die Ausgabe sieht in etwa folgendermaßen aus.

```
Trying 128.0.0.1...
Connected to mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com.
Escape character is '^]'.
>
```

4. Prüfen Sie die Verbindung mithilfe der Memcached-Befehle.

Sie sind nun mit einem Knoten verbunden und können Memcached-Befehle ausführen. Im Folgenden wird ein Beispiel gezeigt.

```
set a 0 0 5      // Set key "a" with no expiration and 5 byte value
hello           // Set value as "hello"
STORED
get a           // Get value for key "a"
VALUE a 0 5
hello
END
get b           // Get value for key "b" results in miss
END
>
```

Unterstützte Knotentypen

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2 Amazon-Instanztypen](#).

Note

Amazon ElastiCache stellt T2-Instances auf den Status der vorherigen Generation um. Sie können keine neuen ElastiCache Cluster mehr mithilfe von T2-Instances erstellen oder neue reservierte T2-Knoten erwerben. Es hat keine Auswirkungen auf bestehende T2-Cluster oder Reservierungen. Wir empfehlen ein Upgrade auf neuere Instance-Typen wie T3- oder T4G-Instances, um eine bessere Leistung und Kosteneffizienz zu erzielen.

Note

Die folgenden Instance-Typen werden in den Regionen AWS Asien-Pazifik (Thailand) und Mexiko (Zentral) unterstützt:

- m7g/r7g: groß, xl, 2xl, 4xl, 8xl, 12xl und 16xl.
- t3/t4g: Mikro, klein und mittel.

Informationen darüber, welche Knotengröße zu verwenden ist, finden Sie unter [Auswahl der Knotengröße](#).

Themen

- [Aktuelle Generation \(Memcached\)](#)
- [Aktuelle Generation \(Valkey oder Redis OSS\)](#)
- [Unterstützte Knotentypen nach AWS-Region](#)
- [Instances mit Spitzenlastleistung](#)
- [Verwandte Informationen](#)

Aktuelle Generation (Memcached)

Die folgenden Tabellen zeigen die Baseline- und Burst-Bandbreite für Instance-Typen, die den I/O Netzwerk-Credit-Mechanismus verwenden, um über ihre Basisbandbreite hinauszugehen.

Note

Instance-Typen mit burststabiler Netzwerkleistung verwenden einen I/O Netzwerk-Credit-Mechanismus, um ihre Basisbandbreite nach bestem Bemühen zu überschreiten.

General

Instance-Typ	Unterstützte Memcached-Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.m7g.large		0,937	12,5
cache.m7g.xlarge		1,876	12,5
cache.m7g.2xlarge		3,75	15
cache.m7g.4xlarge		7,5	15
cache.m7g.8xlarge		15	–
cache.m7g.12xlarge		22,5	–
cache.m7g.16xlarge		30	–
cache.m6g.large	1,5,16	0.75	10.0
cache.m6g.xlarge	1.5,16	1,25	10.0
cache.m6g.2xlarge	1,5,16	2.5	10.0
cache.m6g.4xlarge	1.5,16	5.0	10.0
cache.m6g.8xlarge	1.5,16	12	–

Instance-Typ	Unterstützte Memcached-Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
dache.m6g.12xlarge	1.5,16	20	–
dache.m6g.16xlarge	1.5,16	25	–
cache.m5.large	1.5,16	0.75	10.0
cache.m5.xlarge	1.5,16	1,25	10.0
cache.m5.2xlarge	1,5,16	2.5	10.0
cache.m5.4xlarge	1.5,16	5.0	10.0
cache.m5.12xlarge	1.5,16	–	–
cache.m5.24xlarge	1.5,16	–	–
cache.m4.large	1.5,16	0,45	1.2
cache.m4.xlarge	1,5,16	0.75	2.8
cache.m4.2xlarge	1.5,16	1,0	10.0
cache.m4.4xlarge	1.5,16	2.0	10.0
cache.m4.10xlarge	1.5,16	5.0	10.0
cache.t4g.micro	1.5,16	0,064	5.0
cache.t4g.small	1,5,16	0,128	5.0
cache.t4g.medium	1,5,16	0,256	5.0
cache.t3.micro	1,5,16	0,064	5.0
cache.t3.small	1,5,16	0,128	5.0
cache.t3.medium	1,5,16	0,256	5.0

Instance-Typ	Unterstützte Memcached-Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.t2.micro	1,5,16	0,064	1,024
cache.t2.small	1,5,16	0,128	1,024
cache.t2.medium	1,5,16	0,256	1,024

Für Memcached optimierter Speicher

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r7g.large		0,937	12,5
cache.r7g.xlarge		1,876	12,5
cache.r7g.2xlarge		3,75	15
cache.r7g.4xlarge		7,5	15
cache.r7g.8xlarge		15	–
cache.r7g.12xlarge		22,5	–
cache.r7g.16xlarge		30	–
cache.r6g.large	1,5,16	0.75	10.0
cache.r6g.xlarge	1.5,16	1,25	10.0
cache.r6g.2xlarge	1,5,16	2.5	10.0
cache.r6g.4xlarge	1.5,16	5.0	10.0
cache.r6g.8xlarge	1.5,16	12	–
cache.r6g.12xlarge	1.5,16	20	–

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r6g.16xlarge	1.5,16	25	–
cache.r5.large	1.5,16	0.75	10.0
cache.r5.xlarge	1.5,16	1,25	10.0
cache.r5.2xlarge	1,5,16	2.5	10.0
cache.r5.4xlarge	1.5,16	5.0	10.0
cache.r5.12xlarge	1.5,16	20	–
cache.r5.24xlarge	1.5,16	25	–
cache.r4.large	1.5,16	0.75	10.0
cache.r4.xlarge	1.5,16	1,25	10.0
cache.r4.2xlarge	1,5,16	2.5	10.0
cache.r4.4xlarge	1.5,16	5.0	10.0
cache.r4.8xlarge	1.5,16	12	–
cache.r4.16xlarge	1.5,16	25	–

Für Memcached optimiertes Netzwerk

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.c7gn.large	1.6.6	6,25	30
cache.c7gn.xlarge	1,6.6	12,5	40
cache.c7gn.2xlarge	1.6.6	25	50

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.c7gn.4xlarge	1.6.6	50	–
cache.c7gn.8xlarge	1.6.6	100	–
cache.c7gn.12xlarge	1.6.6	150	–
cache.c7gn.16xlarge	1.6.6	200	–

Aktuelle Generation (Valkey oder Redis OSS)

Weitere Informationen zur vorherigen Generation finden Sie unter [Knoten der vorherigen Generation](#).

 Note

Instance-Typen mit burststabiler Netzwerkleistung verwenden einen I/O Netzwerk-Kreditmechanismus, um ihre Basisbandbreite nach bestem Bemühen zu überschreiten.

General

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/O mit Redis OSS 5.0.6+	TLS-Offloading mit Redis OSS 6.2.5+	Verbessertes Bandbreiten I/O Multiple mit Redis OSS 7.0.4+	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.m7g.large	6.2	N	N	N	0,937	12,5
cache.m7g.xlarge	6.2	Y	Y	Y	1,876	12,5

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/ O mit Redis OSS 5.0.6+	TLS-Offlo ading mit Redis OSS 6.2.5+	Verbes tes Bandw I/O Multiple ing mit Redis OSS 7.0.4+	Baselin e (Gbit/ s)	Maximale Bandbreit e (Gbit/ s)
cache.m7g .2xlarge	6.2	Y	Y	Y	3,75	15
cache.m7g .4xlarge	6.2	Y	Y	Y	7,5	15
cache.m7g .8xlarge	6.2	Y	Y	Y	15	–
cache.m7g .12xlarge	6.2	Y	Y	Y	22,5	–
cache.m7g .16xlarge	6.2	Y	Y	Y	30	–
cache.m6g.large	5,0,6	N	N	N	0.75	10.0
cache.m6g .xlarge	5.0.6	Y	Y	Y	1,25	10.0
dache.m6g .2xlarge	5,0,6	Y	Y	Y	2.5	10.0
dache.m6g .4xlarge	5.0.6	Y	Y	Y	5.0	10.0
dache.m6g .8xlarge	5.0.6	Y	Y	Y	12	–

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/ O mit Redis OSS 5.0.6+	TLS-Offlo ading mit Redis OSS 6.2.5+	Verbes tes Bandw I/O Multiple ing mit Redis OSS 7.0.4+	Baselin e (Gbit/ s)	Maximale Bandbreit e (Gbit/ s)
dache.m6g .12xlarge	5.0.6	Y	Y	Y	20	–
dache.m6g .16xlarge	5.0.6	Y	Y	Y	25	–
cache.m5.large	3.2.4	N	N	N	0.75	10.0
cache.m5.xlarge	3.2.4	Y	N	N	1,25	10.0
cache.m5. 2xlarge	3.2.4	Y	Y	Y	2.5	10.0
cache.m5. 4xlarge	3.2.4	Y	Y	Y	5.0	10.0
cache.m5. 12xlarge	3.2.4	Y	Y	Y	12	–
cache.m5. 24xlarge	3.2.4	Y	Y	Y	25	–
cache.m4.large	3.2.4	N	N	N	0,45	1.2
cache.m4.xlarge	3.2.4	Y	N	N	0.75	2,8
cache.m4. 2xlarge	3.2.4	Y	Y	Y	1,0	10.0

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/ O mit Redis OSS 5.0.6+	TLS-Offlo ading mit Redis OSS 6.2.5+	Verbes tes Bandw I/O Multiple ing mit Redis OSS 7.0.4+	Baselin e (Gbit/ s)	Maximale Bandbreit (Gbit/ s)
cache.m4. 4xlarge	3.2.4	Y	Y	Y	2.0	10.0
cache.m4. 10xlarge	3.2.4	Y	Y	Y	5.0	10.0
cache.t4g.micro	3.2.4	N	N	N	0,064	5.0
cache.t4g.small	5,0,6	N	N	N	0,128	5.0
cache.t4g .medium	5,0,6	N	N	N	0,256	5.0
cache.t3.micro	3.2.4	N	N	N	0,064	5.0
cache.t3.small	3.2.4	N	N	N	0,128	5.0
cache.t3.medium	3.2.4	N	N	N	0,256	5.0
cache.t2.micro	3.2.4	N	N	N	0,064	1,024
cache.t2.small	3.2.4	N	N	N	0,128	1,024
cache.t2.medium	3.2.4	N	N	N	0,256	1,024

Arbeitsspeicher optimiert

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/ O mit Redis OSS 5.0.6+	TLS-Offlo ading mit Redis OSS 6.2.5+	Verbess tes Bandw I/O Multiple ing mit Redis OSS 7.0.4+	Baselin e (Gbit/ s)	Maximale Bandbreit e (Gbit/ s)
cache.r7g.large	6.2	N	N	N	0.937	12,5
cache.r7g.xlarge	6.2	Y	Y	Y	1,876	12,5
cache.r7g .2xlarge	6.2	Y	Y	Y	3,75	15
cache.r7g .4xlarge	6.2	Y	Y	Y	7,5	15
cache.r7g .8xlarge	6.2	Y	Y	Y	15	–
cache.r7g .12xlarge	6.2	Y	Y	Y	22,5	–
cache.r7g .16xlarge	6.2	Y	Y	Y	30	–
cache.r6g.large	5,0,6	N	N	N	0.75	10.0
cache.r6g.xlarge	5.0.6	Y	Y	Y	1,25	10.0
dache.r6g .2xlarge	5,0,6	Y	Y	Y	2.5	10.0
dache.r6g .4xlarge	5.0.6	Y	Y	Y	5.0	10.0

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/ O mit Redis OSS 5.0.6+	TLS-Offlo ading mit Redis OSS 6.2.5+	Verbes tes Bandw I/O Multiple ing mit Redis OSS 7.0.4+	Baselin e (Gbit/ s)	Maximale Bandbreit (Gbit/ s)
dache.r6g .8xlarge	5.0.6	Y	Y	Y	12	–
dache.r6g .12xlarge	5.0.6	Y	Y	Y	20	–
cache.r6g .16xlarge	5.0.6	Y	Y	Y	25	–
cache.r5.large	3.2.4	N	N	N	0.75	10.0
cache.r5.xlarge	3.2.4	Y	N	N	1,25	10.0
cache.r5.2xlarge	3.2.4	Y	Y	Y	2.5	10.0
cache.r5.4xlarge	3.2.4	Y	Y	Y	5.0	10.0
cache.r5. 12xlarge	3.2.4	Y	Y	Y	12	–
cache.r5. 24xlarge	3.2.4	Y	Y	Y	25	–
cache.r4.large	3.2.4	N	N	N	0.75	10.0
cache.r4.xlarge	3.2.4	Y	N	N	1,25	10.0
cache.r4.2xlarge	3.2.4	Y	Y	Y	2.5	10.0
cache.r4.4xlarge	3.2.4	Y	Y	Y	5.0	10.0

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/O mit Redis OSS 5.0.6+	TLS-Offloading mit Redis OSS 6.2.5+	Verbessertes Bandbreiten I/O Multipleing mit Redis OSS 7.0.4+	Baselinne (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r4.8xlarge	3.2.4	Y	Y	Y	12	–
cache.r4.16xlarge	3.2.4	Y	Y	Y	25	–

Mit Daten-Tiering optimierter Speicher

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/O mit Redis OSS 5.0.6+	TLS-Offloading mit Redis OSS 6.2.5+	Verbessertes Bandbreiten I/O Multipleing mit Redis OSS 7.0.4+	Baselinne (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r6gd.xlarge	6.2.0	Y	N	N	1,25	10
cache.r6gd.2xlarge	6.2.0	Y	Y	Y	2.5	10
cache.r6gd.4xlarge	6.2.0	Y	Y	Y	5.0	10

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/O mit Redis OSS 5.0.6+	TLS-Offloading mit Redis OSS 6.2.5+	Verbessertes Bandbreiten I/O Multipleing mit Redis OSS 7.0.4+	Baselinbandbreit (Gbit/s)	Maximale Bandbreit (Gbit/s)
cache.r6gd.8xlarge	6.2.0	Y	Y	Y	12	–
cache.r6gd.12xlarge	6.2.0	Y	Y	Y	20	–
cache.r6gd.16xlarge	6.2.0	Y	Y	Y	25	–

Netzwerkoptimiert

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/O mit Redis OSS 5.0.6+	TLS-Offloading mit Redis OSS 6.2.5+	Verbessertes Bandbreiten I/O Multipleing mit Redis OSS 7.0.4+	Baselinbandbreit (Gbit/s)	Maximale Bandbreit (Gbit/s)
cache.c7gn.large	6.2	N	N	N	6.25	30
cache.c7gn.xlarge	6.2	Y	Y	Y	12,5	40

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbessert I/O mit Redis OSS 5.0.6+	TLS-Offloading mit Redis OSS 6.2.5+	Verbessertes Bandbreiten I/O Multipleing mit Redis OSS 7.0.4+	Baseline Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.c7gn.2xlarge	6.2	Y	Y	Y	25	50
cache.c7gn.4xlarge	6.2	Y	Y	Y	50	–
cache.c7gn.8xlarge	6.2	Y	Y	Y	100	–
cache.c7gn.12xlarge	6.2	Y	Y	Y	150	–
cache.c7gn.16xlarge	6.2	Y	Y	Y	200	–

Unterstützte Knotentypen nach AWS-Region

Die unterstützten Knotentypen können je nach AWS Region variieren. Weitere Informationen finden Sie unter [ElastiCacheAmazon-Preise](#).

Instances mit Spitzenlastleistung

Sie können in Amazon Allzweck-Burstable T4g-, T3-Standard- und T2-Standard-Cache-Knoten starten. ElastiCache Diese Knoten bieten CPU-Baseline-Leistung und die Möglichkeit, diese Leistung jederzeit zu bursten, bis das angesammelte Guthaben aufgebraucht ist. Ein CPU-Guthaben stellt die Leistung eines gesamten CPU-Kerns für eine Minute zur Verfügung.

ElastiCacheDie T4g-, T3- und T2-Knoten von Amazon sind standardmäßig konfiguriert und eignen sich für Workloads mit einer durchschnittlichen CPU-Auslastung, die durchweg unter der

Basisleistung der Instance liegt. Um die Baseline-Leistung zu überschreiten, verwendet der Knoten Guthaben, die als CPU-Guthaben angefallen sind. Wenn das beim Knoten angefallene Guthaben sinkt, wird die Leistung schrittweise bis auf Baseline-Leistung gesenkt. Dank dieser schrittweisen Absenkung kommt es beim Knoten nicht zu einem drastischen Leistungsabfall, wenn das CPU-Guthaben ausgeschöpft ist. Weitere Informationen finden Sie unter [CPU-Guthaben und Basisleistung für Burstable Performance-Instances](#) im EC2 Amazon-Benutzerhandbuch.

Die folgende Tabelle listet die burst-fähigen Leistungsknotentypen und die Rate der verdienten CPU-Guthaben pro Stunde auf. Es zeigt auch die maximale Anzahl an verdienten CPU-Credits, die ein Knoten sammeln kann, und die Anzahl von v CPUs pro Knoten. Ergänzend wird die Baseline-Leistung als Prozentsatz der vollen Kernleistung (einer einzelnen vCPU) angegeben.

Pro Stunde erworbenes CPU-Guthaben	Maximal verdiente Guthaben, die angesammelt werden können*	v CPUs	Basisleistung pro vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung
12	288	2	10 %	0.5	Bis zu 5 Gigabit
24	576	2	20 %	1,37	Bis zu 5 Gigabit
24	576	2	20 %	3,09	Bis zu 5 Gigabit
12	288	2	10 %	0.5	Bis zu 5 Gigabit
24	576	2	20 %	1,37	Bis zu 5 Gigabit
24	576	2	20 %	3,09	Bis zu 5 Gigabit

Pro Stunde erworbenes CPU-Guthaben	Maximal verdiente Guthaben, die angesammelt werden können*	v CPUs	Basisleistung pro vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung
6	144	1	10 %	0.5	Gering bis mäßig
12	288	1	20 %	1,55	Gering bis mäßig
24	576	2	20 %	3,22	Gering bis mäßig

* Die Anzahl der Guthaben, die angesammelt werden können, entspricht dem Guthaben, das in einem 24-Stunden-Zeitraum verdient werden kann.

** Die Basisleistung in der Tabelle ist pro vCPU. Bei einigen Knotengrößen kommen mehrere vCPUs zum Einsatz. Für diese berechnen Sie die CPU-Basisauslastung für den Knoten, indem Sie den vCPU-Prozentsatz mit der Anzahl von v multiplizieren. CPUs

Die folgenden CPU-Kreditmetriken sind für T3- und T4g-Instances mit Spitzenleistung verfügbar:

 Note

Diese Metriken sind nicht für T2-Instances mit Spitzenleistung verfügbar.

- CPUCreditUsage
- CPUCreditBalance

Weitere Informationen zu diesen Metriken finden Sie unter [CPU-Guthaben-Metriken](#).

Darüber hinaus sollten Sie Folgendes beachten:

- Alle Knotentypen der aktuellen Generation werden standardmäßig in einer Virtual Private Cloud (VPC) auf der Grundlage von Amazon VPC erstellt.
- Redis OSS Append-Only Files (AOF) werden für T2-Instances nicht unterstützt. Redis OSS-Konfigurationsvariablen `appendonly` und werden in Redis OSS Version 2.8.22 und höher `appendfsync` nicht unterstützt.

Verwandte Informationen

- [ElastiCache Amazon-Produktmerkmale und -details](#)
- [Memcached-Knotentyp-spezifische Parameter für Memcached](#)
- [Valkey- und Redis OSS-Parameter](#)
- [Verschlüsselung während der Übertragung](#)

Knoten neu starten

Einige Änderungen erfordern einen Neustart eines Valkey-, Memcached- oder Redis OSS-Clusters, damit die Änderungen übernommen werden. Beispielsweise wird bei einigen Parametern eine Änderung am Parameterwert in einer Parametergruppe erst nach einem Neustart übernommen.

Themen

- [Redis OSS-Knoten neu starten \(nur Clustermodus deaktiviert\)](#)
- [Einen Cluster für Memcached neu starten](#)

Redis OSS-Knoten neu starten (nur Clustermodus deaktiviert)

Für Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) lauten die Parameter in Parametergruppen, die erst nach dem Neustart angewendet werden, wie folgt:

- `activeresharding`
- `databases`

Redis-Knoten können nur über die Konsole aktualisiert werden. ElastiCache Sie können immer nur einen einzelnen Knoten neu starten. Um mehrere Knoten neu zu starten, müssen Sie den Vorgang für jeden Knoten wiederholen.

Änderungen der Parameter Valkey oder Redis OSS (Cluster Mode Enabled)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
 - databases
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
 2. Löschen Sie den Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert). Siehe [Löschen eines Clusters in ElastiCache](#).
 3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

Unter Verwendung des AWS-Managementkonsole

Sie können einen Knoten mit der ElastiCache Konsole neu starten.

So starten Sie einen Knoten (Konsole) neu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die zutreffende AWS Region aus.
3. Wählen Sie im linken Navigationsbereich Redis OSS aus.

Eine Liste von Clustern, auf denen Redis OSS ausgeführt wird, wird angezeigt.

4. Wählen Sie den Cluster unter Cluster-Name aus.
5. Unter Knotenname wählen Sie das Optionsfeld neben dem Knoten aus, den Sie neu starten möchten.
6. Wählen Sie Aktionen und dann Neustart Knoten.

Um mehrere Knoten neu zu starten, wiederholen Sie die Schritte 2 bis 5 für jeden Knoten, den Sie neu starten möchten. Es ist nicht nötig zu warten, bis ein Knoten den Neustart beendet hat, um einen anderen neu zu starten.

Einen Cluster für Memcached neu starten

Wenn Sie einen Memcached-Cluster neu starten, leert der Cluster alle seine Daten und startet seine Engine neu. Während dieses Vorgangs können Sie nicht auf den Cluster zugreifen. Da der Cluster alle seine Daten bereinigt hat, starten Sie mit einem leeren Cluster, wenn der Cluster wieder verfügbar ist.

Sie können einen Cluster mithilfe der ElastiCache Konsole, der oder der API neu starten. AWS CLI ElastiCache Unabhängig davon, ob Sie die ElastiCache Konsole, die AWS CLI oder die ElastiCache API verwenden, können Sie nur den Neustart eines einzelnen Clusters initiieren. Um mehrere Cluster neu zu starten, müssen Sie den Prozess oder Vorgang iterieren.

Mit dem AWS-Managementkonsole

Sie können einen Cluster mit der ElastiCache Konsole neu starten.

So starten Sie einen Cluster neu (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, für die Sie sich interessieren.
3. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, den Sie neu starten möchten.

Es wird eine Liste der Cluster der ausgewählten Engine angezeigt.

4. Wählen Sie den neu zu startenden Cluster aus, indem Sie das Kontrollkästchen links neben dem Clusternamen aktivieren.

Die Schaltfläche Reboot (Neustart) wird aktiv.

Wenn Sie mehr als einen Cluster auswählen, ist das Kontrollkästchen Reboot deaktiviert.

5. Wählen Sie Reboot.

Der Bestätigungsbildschirm des Cluster-Neustarts wird angezeigt.

6. Wenn der Cluster neu gestartet werden soll, wählen Sie Reboot. Der Status des Clusters ändert sich in `rebooting cluster nodes` (Neustart von Clusterknoten).

Wenn der Cluster nicht neu gestartet werden soll, wählen Sie Cancel.

Wiederholen Sie zum Neustarten mehrerer Cluster die Schritte 2 bis 5 für jeden Cluster, den Sie neu starten möchten. Sie brauchen nicht zu warten, bis ein Cache-Cluster fertig neu gestartet wurde, bevor Sie einen anderen neu starten.

Um einen bestimmten Knoten neu zu starten, markieren Sie den Knoten und wählen Sie dann Neustart.

Verwenden Sie den AWS CLI

Um einen Cluster neu zu starten (AWS CLI), verwenden Sie den `reboot-cache-cluster`-CLI-Vorgang.

Um bestimmte Knoten im Cluster neu zu starten, listen Sie mit `--cache-node-ids-to-reboot` die spezifischen Cluster auf, die neu gestartet werden sollen. Der folgende Befehl startet die Knoten 0001, 0002 und 0004 von `my-cluster` neu.

Für Linux, macOS oder Unix:

```
aws elasticache reboot-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --cache-node-ids-to-reboot 0001 0002 0004
```

Für Windows:

```
aws elasticache reboot-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --cache-node-ids-to-reboot 0001 0002 0004
```

Um alle Knoten im Cluster neu zu starten, listen Sie mit dem Parameter `--cache-node-ids-to-reboot` die Knoten-IDs aller Cluster auf. Weitere Informationen finden Sie unter [reboot-cache-cluster](#).

Mithilfe der ElastiCache API

Verwenden Sie die `RebootCacheCluster` Aktion, um einen Cluster mithilfe der ElastiCache API neu zu starten.

Um bestimmte Knoten im Cluster neu zu starten, listen Sie mit `CacheNodeIdsToReboot` die spezifischen Cluster auf, die neu gestartet werden sollen. Der folgende Befehl startet die Knoten 0001, 0002 und 0004 von my-cluster neu.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RebootCacheCluster  
&CacheClusterId=my-cluster  
&CacheNodeIdsToReboot.member.1=0001  
&CacheNodeIdsToReboot.member.2=0002  
&CacheNodeIdsToReboot.member.3=0004  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Um alle Knoten im Cluster neu zu starten, listen Sie mit dem Parameter `CacheNodeIdsToReboot` die Knoten-IDs aller Cluster auf. Weitere Informationen finden Sie unter [RebootCacheCluster](#).

Knoten ersetzen (Valkey und Redis OSS)

Amazon aktualisiert seine Flotte ElastiCache häufig, indem Patches und Upgrades nahtlos auf Instances angewendet werden. Von Zeit zu Zeit müssen wir Ihre ElastiCache Knoten jedoch neu starten, um obligatorische Betriebssystemupdates auf den zugrunde liegenden Host anzuwenden. Diese Ersetzungen sind erforderlich, um Upgrades anzuwenden, die die Sicherheit, Zuverlässigkeit und Betriebsleistung erhöhen.

Sie haben die Möglichkeit, diese Ersetzungen vor dem geplanten Knotenersetzungsfenster jederzeit selbst zu verwalten. Wenn Sie eine Ersetzung selbst verwalten, erhält Ihre Instance das Betriebssystem-Update, wenn Sie den Knoten neu starten, und der geplante Austausch des Knotens wird abgebrochen. Sie erhalten möglicherweise weiterhin Benachrichtigungen darüber, dass der Knotenaustausch stattfinden wird. Wenn Sie die erforderliche Wartung bereits manuell vorgenommen haben, können Sie diese Benachrichtigungen ignorieren.

Note

Von Amazon automatisch generierte Ersatz-Cache-Knoten ElastiCache können unterschiedliche IP-Adressen haben. Sie sind dafür verantwortlich, Ihre Anwendungskonfiguration zu überprüfen, um sicherzustellen, dass Ihre Cache-Knoten den entsprechenden IP-Adressen zugeordnet sind.

In der folgenden Liste sind die Maßnahmen aufgeführt, die Sie ergreifen können, wenn ElastiCache Sie einen Ihrer Valkey- oder Redis-OSS-Knoten für den Austausch planen. Um die für Ihre Situation erforderlichen Informationen schneller zu finden, wählen Sie eine Option aus dem folgenden Menü aus.

- [Do nothing](#)— Lassen Sie Amazon den Knoten wie geplant ElastiCache ersetzen.
- [Change your maintenance window](#) – Ändern Sie Ihr Wartungsfenster und wählen Sie einen besser geeigneten Termin.
- Valkey- oder Redis OSS-Konfigurationen (Cluster-Modus aktiviert)
 - [Replace the only node in any Valkey or Redis OSS cluster](#)— Ein Verfahren zum Ersetzen eines Knotens in einem Valkey- oder Redis OSS-Cluster mithilfe von Backup und Restore.
 - [Replace a replica node in any Valkey or Redis OSS cluster](#)— Ein Verfahren zum Ersetzen einer Read-Replica in einem beliebigen Valkey- oder Redis-OSS-Cluster durch Erhöhen und Verringern der Replikatanzahl ohne Cluster-Ausfallzeiten.

- [Replace any node in a Valkey or Redis OSS \(cluster mode enabled\) shard](#)— Ein dynamisches Verfahren ohne Cluster-Ausfallzeiten, bei dem ein Knoten in einem Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) durch Aus- und In-Skalierung ersetzt wird.
- Valkey- oder Redis OSS-Konfigurationen (Clustermodus deaktiviert)
 - [Replace the only node in any Valkey or Redis OSS cluster](#)— Verfahren zum Ersetzen eines beliebigen Knotens in einem Valkey- oder Redis OSS-Cluster mithilfe von Backup und Restore.
 - [Replace a replica node in any Valkey or Redis OSS cluster](#)— Ein Verfahren zum Ersetzen einer Lesereplik in einem beliebigen Valkey- oder Redis-OSS-Cluster durch Erhöhen und Verringern der Replikatanzahl ohne Cluster-Ausfallzeiten.
 - [Replace a node in a Valkey or Redis OSS \(cluster mode disabled\) cluster](#)— Verfahren zum Ersetzen eines Knotens in einem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) mithilfe von Replikation.
 - [Replace a Valkey or Redis OSS \(cluster mode disabled\) read-replica](#)— Ein Verfahren zum manuellen Ersetzen einer Read-Replica in einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert).
 - [Replace a Valkey or Redis OSS \(cluster mode disabled\) primary node](#)— Ein Verfahren zum manuellen Ersetzen des Primärknotens in einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert).

Optionen für den Austausch von Valkey- und Redis OSS-Knoten

- Nichts tun — Wenn Sie nichts tun, wird der Knoten wie geplant ElastiCache ersetzt.

Bei Nicht-Cluster-Konfigurationen mit aktiviertem Autofailover werden Cluster auf Valkey 7.2 und höher und Redis OSS 5.0.6 und höher vollständig ersetzt, während der Cluster weiterhin online bleibt und eingehende Schreib Anforderungen bearbeitet. Bei Clustern mit aktiviertem Auto Failover auf Redis OSS 4.0.10 oder niedriger stellen Sie möglicherweise eine kurze Schreibunterbrechung von bis zu einigen Sekunden im Zusammenhang mit DNS-Updates fest.

Wenn der Knoten Mitglied eines Clusters mit auto Failover ist, bietet OSS ElastiCache für Valkey oder Redis eine verbesserte Verfügbarkeit bei Patches, Updates und anderen wartungsbezogenen Knotenersetzungen.

Bei ElastiCache Clusterkonfigurationen, die für die Verwendung ElastiCache für Valkey- oder Redis OSS-Clusterclients eingerichtet sind, wird der Austausch jetzt abgeschlossen, während der Cluster eingehende Schreibanforderungen bearbeitet.

Bei Nicht-Cluster-Konfigurationen mit aktiviertem Autofailover werden Cluster auf Valkey 7.2 und höher und Redis OSS 5.0.6 und höher vollständig ersetzt, während der Cluster weiterhin online bleibt und eingehende Schreibanforderungen bearbeitet. Bei Clustern mit aktiviertem Auto Failover auf Redis OSS 4.0.10 oder niedriger stellen Sie möglicherweise eine kurze Schreibunterbrechung von bis zu einigen Sekunden im Zusammenhang mit DNS-Updates fest.

Wenn der Knoten eigenständig ist, startet Amazon ElastiCache zuerst einen Ersatzknoten und synchronisiert dann vom vorhandenen Knoten aus. Der vorhandene Knoten ist während dieser Zeit nicht für Serviceanfragen verfügbar. Sobald die Synchronisierung abgeschlossen ist, wird der bestehende Knoten beendet und der neue Knoten nimmt seinen Platz ein. ElastiCache bemüht sich nach besten Kräften, Ihre Daten während dieses Vorgangs aufzubewahren.

- Ändern Sie Ihr Wartungsfenster — Bei geplanten Wartungsereignissen erhalten Sie eine E-Mail oder eine Benachrichtigung von ElastiCache. Wenn Sie in diesen Fällen Ihr Wartungsfenster vor der geplanten Ersatzzeit ändern, wird Ihr Knoten zur neuen Uhrzeit ersetzt. Weitere Informationen finden Sie hier:
 - [Einen ElastiCache Cluster ändern](#)
 - [Ändern einer Replikationsgruppe](#)

 Note

Die Möglichkeit, Ihr Ersatzfenster zu ändern, indem Sie Ihr Wartungsfenster verschieben, ist nur verfügbar, wenn die ElastiCache Benachrichtigung ein Wartungsfenster beinhaltet. Wenn die Benachrichtigung kein Wartungsfenster enthält, können Sie Ihr Ersatzfenster nicht ändern.

Es ist beispielsweise Donnerstag, der 9. November, um 15:00 Uhr und das nächste Wartungsfenster ist am Freitag, dem 10. November, um 17:00 Uhr. Es folgen drei Szenarien mit den jeweiligen Ergebnissen:

- Sie ändern Ihr Wartungsfenster auf Freitag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben, aber vor dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Freitag, dem 10. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Samstag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben und nach dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Samstag, dem 11. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Mittwoch, 16:00 Uhr, früher in der Woche, als die aktuellen Datums- und Uhrzeitangaben. Dieser Knoten wird am kommenden Mittwoch, dem 15. November, um 16:00 Uhr ersetzt.

Detaillierte Anweisungen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

- Ersetzen Sie den einzigen Knoten in einem Valkey- oder Redis OSS-Cluster — Wenn der Cluster keine Read Replicas hat, können Sie das folgende Verfahren verwenden, um den Knoten zu ersetzen.

So ersetzen Sie den einzigen Knoten mithilfe von Sicherung und Wiederherstellung

1. Erstellen Sie einen Snapshot des Clusters des Knotens. Detaillierte Anweisungen finden Sie unter [Erstellen manueller Backups](#).
 2. Erstellen Sie einen neuen Cluster und übernehmen Sie die Daten mithilfe des Snapshots. Detaillierte Anweisungen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).
 3. Löschen Sie den Cluster mit dem für das Ersetzen geplanten Knoten. Detaillierte Anweisungen finden Sie unter [Löschen eines Clusters in ElastiCache](#).
 4. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Knotens.
- Ersetzen Sie einen Replikatknoten in einem beliebigen Valkey- oder Redis OSS-Cluster — Um einen Replikatcluster zu ersetzen, erhöhen Sie die Anzahl der Replikate. Fügen Sie dafür ein

Replikat hinzu und verringern Sie die Replikatanzahl, indem Sie das Replikat entfernen, das Sie ersetzen möchten. Dieser Vorgang ist dynamisch und ist nicht mit Cluster-Ausfallzeiten verbunden.

 Note

Wenn Ihr Shard oder Ihre Replikationsgruppe bereits fünf Replikate hat, kehren Sie die Schritte 1 und 2 um.

Um ein Replikat in einem Valkey- oder Redis OSS-Cluster zu ersetzen

1. Erhöhen Sie die Replikatanzahl, indem Sie ein Replikat zum Shard oder zur Replikationsgruppe hinzufügen. Weitere Informationen finden Sie unter [Erhöhen der Anzahl der Replikate in einem Shard](#).
 2. Löschen Sie das Replikat, das Sie ersetzen möchten. Weitere Informationen finden Sie unter [Verringern der Anzahl der Replikate in einem Shard](#).
 3. Aktualisieren Sie die Endpunkte in Ihrer Anwendung.
- Ersetzen Sie einen beliebigen Knoten in einem Valkey- oder Redis OSS-Shard (Clustermodus aktiviert) — Verwenden Sie Online-Resharding, um den Knoten in einem Cluster ohne Ausfallzeiten zu ersetzen. Fügen Sie zunächst einen Shard durch Ausskalieren hinzu und löschen Sie den Shard mit dem Knoten, der durch Einkalieren ersetzt werden soll.

Um einen beliebigen Knoten in einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) zu ersetzen

1. Horizontales Skalieren: Fügen Sie einen zusätzlichen Shard mit derselben Konfiguration wie die des bestehenden Shards mit dem zu ersetzenden Knoten hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Shards mit Online-Resharding](#).
2. Vertikales Skalieren: Löschen Sie den Shard mit dem zu ersetzenden Knoten. Weitere Informationen finden Sie unter [Entfernen von Shards mit Online-Resharding](#).
3. Aktualisieren Sie die Endpunkte in Ihrer Anwendung.

- Ersetzen Sie einen Knoten in einem Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) — Wenn es sich bei dem Cluster um einen Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) ohne Read Replicas handelt, verwenden Sie das folgende Verfahren, um den Knoten zu ersetzen.

So ersetzen Sie den Knoten mithilfe der Replikation (nur mit deaktiviertem Cluster-Modus)

1. Fügen Sie dem Cluster die Replikation mit dem Knoten hinzu, der als primärer Knoten ersetzt werden soll. Aktivieren Sie Multi-AZ auf diesem Cluster nicht. Detaillierte Anweisungen finden Sie unter [Um die Replikation zu einem Valkey- oder Redis OSS-Cluster ohne Shards hinzuzufügen](#).
 2. Fügen Sie dem Cluster ein Lesereplikat hinzu. Detaillierte Anweisungen finden Sie unter [Um Knoten zu einem ElastiCache Cluster \(Konsole\) hinzuzufügen](#).
 3. Stufen Sie das soeben erstellte Lesereplikat auf Primär hoch. Detaillierte Anweisungen finden Sie unter [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#).
 4. Löschen Sie den für das Ersetzen geplanten Knoten. Detaillierte Anweisungen finden Sie unter [Knoten aus einem ElastiCache Cluster entfernen](#).
 5. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Knotens.
- Ersetzen Sie eine Valkey- oder Redis OSS-Read-Replica (Cluster-Modus deaktiviert) — Wenn es sich bei dem Knoten um eine Read-Replica handelt, ersetzen Sie den Knoten.

Wenn Ihr Cluster nur einen Replikatknoten enthält und Multi-AZ aktiviert ist, müssen Sie Multi-AZ deaktivieren, bevor Sie das Replikat löschen können. Detaillierte Anweisungen finden Sie unter [Ändern einer Replikationsgruppe](#).

Um ein Valkey- oder Redis OSS-Lese-Replikat (Cluster-Modus deaktiviert) zu ersetzen

1. Löschen Sie das für den Austausch geplante Replikat. Detaillierte Informationen finden Sie hier:
 - [Verringern der Anzahl der Replikate in einem Shard](#)
 - [Knoten aus einem ElastiCache Cluster entfernen](#)

2. Fügen Sie ein neues Replikat hinzu, das das auszutauschende Replikat ersetzen soll. Wenn Sie denselben Namen wie für das soeben gelöschte Replikat verwenden, können Sie Schritt 3 überspringen. Detaillierte Informationen finden Sie hier:
 - [Erhöhen der Anzahl der Replikate in einem Shard](#)
 - [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#)
 3. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Replikats.
 4. Wenn Sie die Multi-AZ-Funktion beim Start deaktiviert haben, aktivieren Sie sie nun wieder. Detaillierte Anweisungen finden Sie unter [Aktivieren von Multi-AZ](#).
- Ersetzen Sie einen primären Knoten von Valkey oder Redis OSS (Cluster-Modus deaktiviert)
 - Wenn der Knoten der primäre Knoten ist, stufen Sie zuerst eine Read-Replica zum primären Knoten herauf. Löschen Sie dann das Replikat, das vorher der primäre Knoten war.

Wenn Ihr Cluster nur ein Replikat enthält und Multi-AZ aktiviert ist, müssen Sie Multi-AZ deaktivieren, bevor Sie das Replikat in Schritt 2 löschen können. Detaillierte Anweisungen finden Sie unter [Ändern einer Replikationsgruppe](#).

Um einen Valkey- oder Redis OSS-Primärknoten (Cluster-Modus deaktiviert) zu ersetzen

1. Stufen Sie ein Lesereplikat auf Primär hoch. Detaillierte Anweisungen finden Sie unter [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#).
2. Löschen Sie den Knoten, der ausgetauscht werden soll (der alte primäre Knoten). Detaillierte Anweisungen finden Sie unter [Knoten aus einem ElastiCache Cluster entfernen](#).
3. Fügen Sie ein neues Replikat hinzu, das das auszutauschende Replikat ersetzen soll. Wenn Sie denselben Namen wie den des soeben gelöschten Knotens verwenden, können Sie das Ändern der Endpunkte in Ihrer Anwendung überspringen.

Detaillierte Anweisungen finden Sie unter [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

4. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Knotens.

5. Wenn Sie die Multi-AZ-Funktion beim Start deaktiviert haben, aktivieren Sie sie nun wieder. Detaillierte Anweisungen finden Sie unter [Aktivieren von Multi-AZ](#).

Knoten ersetzen (Memcached)

Amazon ElastiCache for Memcached aktualisiert seine Flotte regelmäßig, indem Patches und Upgrades nahtlos auf Instances angewendet werden. Von Zeit zu Zeit müssen wir jedoch Ihre ElastiCache for Memcached-Knoten neu starten, um obligatorische Betriebssystem-Updates auf den zugrunde liegenden Host anzuwenden. Diese Ersetzungen sind erforderlich, um Upgrades anzuwenden, die die Sicherheit, Zuverlässigkeit und Betriebsleistung erhöhen.

Sie haben die Möglichkeit, diese Ersetzungen vor dem geplanten Knotenersetzungsfenster jederzeit selbst zu verwalten. Wenn Sie eine Ersetzung selbst verwalten, erhält Ihre Instance das Betriebssystem-Update, wenn Sie den Knoten neu starten, und der geplante Austausch des Knotens wird abgebrochen. Sie erhalten möglicherweise weiterhin Benachrichtigungen darüber, dass der Knotenaustausch stattfindet. Wenn Sie die erforderliche Wartung bereits manuell vorgenommen haben, können Sie diese Benachrichtigungen ignorieren.

Note

Von Amazon automatisch generierte Ersatz-Cache-Knoten ElastiCache können unterschiedliche IP-Adressen haben. Sie sind dafür verantwortlich, Ihre Anwendungskonfiguration zu überprüfen, um sicherzustellen, dass Ihre Cache-Knoten den entsprechenden IP-Adressen zugeordnet sind.

In der folgenden Liste sind die Maßnahmen aufgeführt, die Sie ergreifen können, ElastiCache wenn Sie den Austausch eines Ihrer Memcached-Knoten planen.

- **Nichts tun** — Wenn Sie nichts tun, wird der Knoten wie geplant ElastiCache ersetzt. Wenn der Knoten ElastiCache automatisch durch einen neuen Knoten ersetzt wird, ist der neue Knoten zunächst leer.
- **Ändern Sie Ihr Wartungsfenster** — Bei geplanten Wartungsereignissen erhalten Sie eine E-Mail oder eine Benachrichtigung von ElastiCache. Wenn Sie in diesen Fällen Ihr Wartungsfenster vor der geplanten Ersatzzeit ändern, wird Ihr Knoten zur neuen Uhrzeit ersetzt. Weitere Informationen finden Sie unter [Einen ElastiCache Cluster ändern](#).

Note

Die Möglichkeit, Ihr Ersatzfenster zu ändern, indem Sie Ihr Wartungsfenster verschieben, ist nur verfügbar, wenn die ElastiCache Benachrichtigung ein Wartungsfenster beinhaltet.

Wenn die Benachrichtigung kein Wartungsfenster enthält, können Sie Ihr Ersatzfenster nicht ändern.

Es ist beispielsweise Donnerstag, der 9. November, um 15:00 Uhr und das nächste Wartungsfenster ist am Freitag, dem 10. November, um 17:00 Uhr. Es folgen drei Szenarien mit den jeweiligen Ergebnissen:

- Sie ändern Ihr Wartungsfenster auf Freitag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben, aber vor dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Freitag, dem 10. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Samstag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben und nach dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Samstag, dem 11. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Mittwoch, 16:00 Uhr, früher in der Woche, als die aktuellen Datums- und Uhrzeitangaben. Dieser Knoten wird am kommenden Mittwoch, dem 15. November, um 16:00 Uhr ersetzt.

Detaillierte Anweisungen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

- Manuelles Ersetzen des Knotens – Wenn Sie den Knoten vor dem nächsten Wartungsfenster ersetzen müssen, ersetzen Sie den Knoten manuell.

Wenn Sie den Knoten manuell ersetzen, werden Schlüssel umverteilt. Die Umverteilung verursacht Cache-Fehler.

So ersetzen Sie einen Memcached-Knoten manuell

1. Löschen Sie den für das Ersetzen geplanten Knoten. Detaillierte Anweisungen finden Sie unter [Knoten aus einem ElastiCache Cluster entfernen](#).
2. Fügen Sie dem Cluster einen neuen Knoten hinzu. Detaillierte Anweisungen finden Sie unter [Hinzufügen von Knoten zu einem ElastiCache Cluster](#).
3. Wenn Sie Auto Discovery in diesem Cluster nicht verwenden, rufen Sie Ihre Anwendung auf und tauschen Sie jede Instance des Endpunkts des alten mit dem Endpunkt des neuen Knotens aus.

Reservierte Knoten

Die Reservierung eines oder mehrerer ElastiCache Knoten kann eine Möglichkeit für Sie sein, die Kosten zu senken. Für reservierte Knoten wird eine Vorausgebühr erhoben, die von der Art des Knotens und der Dauer der Reservierung – ein oder drei Jahre – abhängt.

Um festzustellen, ob reservierte Knoten für Ihre Anwendungsfälle Kosten einsparen, ermitteln Sie zunächst die Knotengröße und wie viele Knoten Sie benötigen. Schätzen Sie dann die Verwendung des Knotens und vergleichen Sie die Gesamtkosten von On-Demand-Knoten im Gegensatz zu reservierten Knoten. Sie können die Nutzung von reservierten und On-Demand-Knoten in Ihren Clustern mischen und anpassen. Preisinformationen finden Sie unter [ElastiCache Amazon-Preise](#).

Themen

- [Preismanagement mit reservierten Knotenpunkten](#)
- [Standardangebote für reservierte Knoten](#)
- [Größe: flexible reservierte Knoten](#)
- [Löschen eines reservierten Knotens](#)
- [Ältere Angebote für reservierte Knoten](#)
- [Abrufen von Informationen über Angebote für reservierte Knoten](#)
- [Kauf eines reservierten Knotens](#)
- [Abrufen von Informationen über Ihre reservierten Knoten](#)

Preismanagement mit reservierten Knotenpunkten

Durch Reservieren eines oder mehrerer Knoten lassen sich die Kosten möglicherweise reduzieren. Für reservierte Knoten wird eine Vorausgebühr erhoben, die von der Art des Knotens und der Dauer der Reservierung – ein oder drei Jahre – abhängt. Diese Gebühr ist wesentlich geringer als die stündliche Nutzungsgebühr, die bei On-Demand-Knoten anfällt.

Um festzustellen, ob reservierte Knoten für Ihre Anwendungsfälle Kosten einsparen, ermitteln Sie zunächst die Knotengröße und wie viele Knoten Sie benötigen. Schätzen Sie dann die Nutzung des Knotens und vergleichen Sie die Gesamtpreise, die Ihnen durch die Nutzung von On-Demand-Knoten im Vergleich zu reservierten Knoten entstehen. Sie können die Nutzung von reservierten und On-Demand-Knoten in Ihren Clustern mischen und anpassen. Preisinformationen finden Sie unter [ElastiCache Amazon-Preise](#).

AWS Region, Knotentyp und Laufzeit müssen beim Kauf ausgewählt werden und können später nicht mehr geändert werden.

Sie können die AWS-Managementkonsole, oder die ElastiCache API verwenden AWS CLI, um verfügbare Angebote für reservierte Knoten aufzulisten und zu erwerben.

Weitere Informationen zu reservierten Knoten finden Sie unter [Amazon ElastiCache Reserved Nodes](#).

Standardangebote für reservierte Knoten

Wenn Sie eine Reserved Node Instance (RI) bei Amazon erwerben ElastiCache, können Sie sich verpflichten, für die Dauer der Reserved Node Instance einen vergünstigten Tarif für einen bestimmten Node-Instance-Typ und eine AWS Region zu erhalten. Um eine Amazon ElastiCache Reserved Node Instance zu verwenden, erstellen Sie eine neue ElastiCache Node-Instance, genau wie bei einer On-Demand-Instance.

Wenn die Spezifikationen der neuen Reserve-Node-Instance mit einer bestehenden Reserved Node-Instance für Ihr Konto übereinstimmen, wird Ihnen der für die Reserved Node-Instance angebotene vergünstigte Tarif in Rechnung gestellt. Andernfalls wird die Knoten-Instance auf Basis eines On-Demand-Preises abgerechnet. Diese Standardversionen RIs sind ab den Instance-Familien R5 und M5 verfügbar.

Note

Alle im Folgenden erörterten Angebotstypen sind mit Laufzeiten von einem Jahr und drei Jahren erhältlich.

Angebotstypen

No Upfront RI ermöglicht den Zugriff auf eine Reserved ElastiCache Instance, ohne dass eine Vorauszahlung erforderlich ist. Ihre No Upfront Reserved ElastiCache Instance berechnet für jede Stunde innerhalb der Laufzeit einen vergünstigten Stundensatz, unabhängig von der Nutzung.

Bei teilweiser Vorauszahlung muss ein Teil der Reserved Elasticache Instance im Voraus bezahlt werden. Die innerhalb der Laufzeit verbleibenden Stunden werden unabhängig von der Nutzung zu einem vergünstigten Stundensatz berechnet. Diese Option ersetzt die Legacy-Option Hohe Auslastung, die im nächsten Abschnitt erklärt wird.

Vorauszahlung RI muss der gesamte Betrag zu Beginn der Laufzeit der RI bezahlt werden. Für den Rest der Laufzeit entstehen Ihnen keine weiteren Preise, unabhängig von der Anzahl der genutzten Stunden.

Größe: flexible reservierte Knoten

Alle reservierten Knoten sind größenflexibel. Wenn Sie einen reservierten Knoten kaufen, geben Sie unter anderem den Knotentyp an, zum Beispiel `cache.r6g.xlarge`. Weitere Informationen zu Knotentypen finden Sie unter [ElastiCacheAmazon-Preise](#).

Wenn Sie über einen Node verfügen und diesen auf eine größere Kapazität skalieren müssen, wird Ihr reservierter Node automatisch auf Ihren skalierten Node angewendet. Das heißt, Ihre reservierten Knoten werden automatisch für die Nutzung beliebiger Größe in derselben Knotenfamilie verwendet. Größenflexible reservierte Knoten sind für Knoten mit derselben AWS Region verfügbar. Größenflexible reservierte Knoten können nur innerhalb ihrer Knotenfamilien skaliert werden. Beispielsweise kann ein reservierter Knoten für `cache.r6g.xlarge` für `cache.r6g.2xlarge` gelten, aber nicht für `cache.r6gd.large`, da `cache.r6g` und `cache.r6gd` unterschiedliche Knotenfamilien sind.

Größenflexibilität bedeutet, dass Sie sich frei zwischen Konfigurationen innerhalb derselben Knotenfamilie bewegen können. Sie können beispielsweise ohne zusätzliche Kosten von einem reservierten `r6g.xlarge`-Knoten (8 normalisierte Einheiten) zu zwei reservierten `r6g.large`-Knoten (8 normalisierte Einheiten) ($2 \times 4 = 8$ normalisierte Einheiten) in derselben Region wechseln. AWS

Aktualisieren von Knoten von Redis OSS auf Valkey

Mit dem Start von Valkey können Sie jetzt Ihren Redis OSS-Rabatt für reservierte Knoten auf die Valkey-Cache-Engine anwenden. ElastiCache Sie können ein Upgrade von Redis OSS auf Valkey durchführen und gleichzeitig von bestehenden Verträgen und Reservierungen profitieren. Sie können Ihre Vorteile nicht nur innerhalb der Cache-Node-Familie und -Engine nutzen, sondern auch einen größeren Mehrwert erzielen. Valkey ist im Vergleich zu Redis OSS mit einem discount von 20% erhältlich, und dank der Flexibilität für reservierte Knoten können Sie Ihre reservierten Redis OSS-Nodes verwenden, um 20% mehr laufende Valkey-Knoten abzudecken.

Zur Berechnung des ermäßigten Tarifs hat jede Kombination aus ElastiCache Knoten und Motor einen Normalisierungsfaktor, der in Einheiten gemessen wird. Reservierte Knoteneinheiten können auf jeden laufenden Knoten innerhalb der Instanzfamilie des reservierten Knotens für eine bestimmte Engine angewendet werden. Reservierte Redis OSS-Knoten können zusätzlich Engine-übergreifend eingesetzt werden, um laufende Valkey-Knoten abzudecken. Da Valkey im Vergleich zu Redis OSS und Memcached günstiger ist, sind die Einheiten für einen bestimmten Instance-Typ niedriger, sodass ein reservierter Redis OSS-Node mehr Valkey-Knoten abdecken kann.

Nehmen wir als Beispiel an, Sie haben einen reservierten Knoten für einen cache.r7g.4xlarge für die Redis OSS-Engine (32 Einheiten) gekauft und betreiben einen cache.r7g.4xlarge Redis OSS-Knoten (32 Einheiten). Wenn Sie den Knoten auf Valkey aktualisieren, sinkt der Normalisierungsfaktor des laufenden Knotens auf 25,6 Einheiten, und Ihr vorhandener reservierter Knoten bietet Ihnen zusätzliche 6,4 Einheiten, die Sie gegen jeden anderen laufenden Valkey- oder Redis-OSS-Knoten innerhalb der cache.r7g-Familie in der Region verwenden können. Sie könnten damit 25% eines anderen cache.r7g.4xlarge Valkey-Knotens im Konto (25,6 Einheiten) oder 100% eines cache.r7g.xlarge Valkey-Knotens (6,4 Einheiten) abdecken.

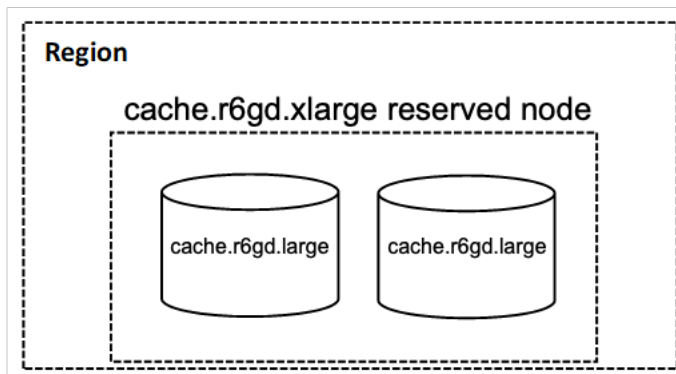
Vergleich der Nutzung mit normalisierten Einheiten

Sie können die Nutzung für verschiedene reservierte Knotengrößen vergleichen, indem Sie normalisierte Einheiten verwenden. Beispielsweise entspricht eine Nutzungsstunde auf zwei cache.r6g.4xlarge-Knoten 16 Stunden Nutzung auf einem cache.r6g.large. Die folgende Tabelle zeigt die Anzahl der normalisierten Einheiten für jede Knotengröße:

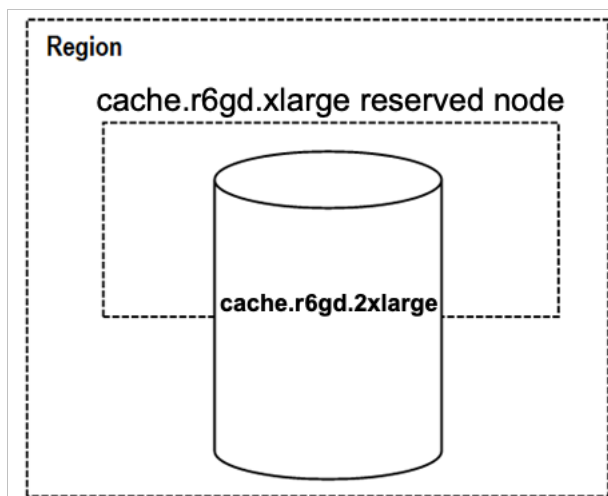
Knotengröße	Normalisierte Einheiten mit Redis OSS oder Memcached	Normalisierte Einheiten mit Valkey
Micro	0.5	0.4
small	1	8.
Medium	2	1,6
large	4	3.2
xlarge	8	6.4
2xlarge	16	12.8
4xlarge	32	25,6
6xlarge	48	38,4
8xlarge	64	51,2
10xlarge	80	64
12xlarge	96	76,8

Knotengröße	Normalisierte Einheiten mit Redis OSS oder Memcached	Normalisierte Einheiten mit Valkey
16xlarge	128	102,4
24xlarge	192	153,6

Beispiel: Sie kaufen einen reservierten Cache.r6gd.xlarge-Node und Sie haben zwei laufende reservierte Cache.r6gd.large-Nodes in Ihrem Konto in derselben Region. AWS In diesem Fall wird der Abrechnungsvorteil vollständig auf beide Knoten angewendet.



Wenn in Ihrem Konto in derselben AWS Region eine cache.r6gd.2xlarge-Instance ausgeführt wird, wird der Abrechnungsvorteil alternativ auf 50 Prozent der Nutzung des reservierten Knotens angerechnet.



Löschen eines reservierten Knotens

Die Bedingungen für einen reservierten Knoten beinhalten eine Laufzeit von einem oder drei Jahren. Sie können einen reservierten Knoten nicht kündigen. Sie können jedoch einen Knoten löschen, für den ein discount für reservierte Knoten gilt. Der Vorgang zum Löschen eines Knotens, für den ein discount für reservierte Knoten gilt, ist derselbe wie für jeden anderen Knoten.

Wenn Sie einen Knoten löschen, für den ein discount für reservierte Knoten gilt, können Sie einen anderen Knoten mit kompatiblen Spezifikationen starten. In diesem Fall erhalten Sie den Rabatt während des Reservierungszeitraums (ein Jahr oder drei Jahre).

Ältere Angebote für reservierte Knoten

Es gibt drei Stufen von Legacy-Knoten-Reservierungen – schwere Auslastung, mittlere Auslastung und leichte Auslastung. Knoten können in jeder Auslastung für ein oder drei Jahre reserviert werden. Der Knotentyp, der Auslastungsgrad und die Reservierungsdauer beeinflussen Ihre Gesamtpreise. Überprüfen Sie die Einsparungen, die reservierte Knoten Ihrem Unternehmen bringen können, indem Sie verschiedene Modelle vergleichen, bevor Sie reservierte Knoten kaufen.

Die Auslastungsstufe oder Reservierungsdauer eines erworbenen Knotens lässt sich nicht mehr ändern.

Auslastungsstufen

Reservierte Knoten mit starker Auslastung ermöglichen Workloads mit einer konsistenten Kapazitätsgrundlage oder konstante Workloads. Reservierte Knoten mit starker Auslastung erfordern zwar eine hohe Vorabzahlung, doch wenn Sie planen, mehr als 79 % der Reservierungsdauer zu nutzen, erzielen Sie die größten Einsparungen (bis zu 70 % weniger als der Preis von On-Demand-Knoten). Bei reservierten Knoten mit hoher Auslastung zahlen Sie eine einmalige Gebühr. Danach wird für die Dauer der Laufzeit eine niedrigere Stundengebühr fällig, unabhängig davon, ob Ihr Knoten läuft.

Reservierte Knoten mit mittlerer Auslastung sind die beste Option, wenn Sie vorhaben, Ihre reservierten Knoten einen Großteil der Zeit zu nutzen, und Sie entweder eine niedrigere Einmalgebühr wünschen oder nicht mehr für Ihren Knoten zahlen möchten, wenn Sie ihn abschalten. Reservierte Knoten mit mittlerer Auslastung stellen eine kosteneffiziente Option dar, wenn Sie planen, mehr als 40 % der Knotenreservierungsdauer in Anspruch zu nehmen. Mit dieser Option können Sie bis zu 64 % gegenüber dem Preis für On-Demand-Knoten sparen. Für reservierte Knoten mit mittlerer Auslastung zahlen Sie eine etwas höhere Einmalgebühr als mit reservierten Knoten mit

geringer Auslastung und profitieren von niedrigeren Nutzungstarifen pro Stunde, wenn Sie einen Knoten ausführen.

Reservierte Knoten mit geringer Auslastung sind ideal für periodische Workloads, die nur wenige Stunden pro Tag oder an wenigen Tagen pro Woche ausgeführt werden. Für reservierte Knoten mit geringer Auslastung zahlen Sie eine Einmalgebühr sowie eine ermäßigte Nutzungsgebühr pro Stunde, wenn Ihr Knoten ausgeführt wird. Sie können mit dem Sparen beginnen, wenn Ihr Knoten zu mehr als 17 % der reservierten Knotenlaufzeit läuft. Sie können während der gesamten Laufzeit Ihres reservierten Knotens bis zu 56 % gegenüber den On-Demand-Tarifen sparen.

Ältere Angebote für reservierte Knoten

Angebot	Vorabkosten	Nutzungsgebühr	Vorteil
Hohe Auslastung	Am höchsten	Niedrigste Gebühr pro Stunde. Gilt für die gesamte Dauer, unabhängig davon, ob Sie den reservierten Knoten in Anspruch nehmen.	Niedrigste Gesamtkosten, wenn Sie planen, Ihre reservierten Knoten mehr als 79 % eines 3-Jahres-Zeitraums auszuführen.
Mittlere Auslastung	Mittelschwer	Nutzungsgebühr pro Stunde für jede Stunde, die der Knoten ausgeführt wird. Keine Nutzungsgebühr pro Stunde, wenn der Knoten nicht ausgeführt wird.	Geeignet für elastische Workloads oder bei erwarteter mittlerer Nutzung, mehr als 40 % eines 3-Jahres-Zeitraums.
Leichte Auslastung	Am niedrigsten	Nutzungsgebühr pro Stunde für jede Stunde, die der Knoten ausgeführt wird. Keine	Die höchsten Gesamtpreise, wenn Sie planen, die ganze Zeit in Betrieb zu sein. Dies sind

Angebot	Vorabkosten	Nutzungsgebühr	Vorteil
		Nutzungsgebühr pro Stunde, wenn der Knoten nicht ausgeführt wird. Höchste Gebühren pro Stunde aller Angebotstarife, die Gebühren fallen jedoch nur an, wenn der reservierte Knoten ausgeführt wird.	jedoch die niedrigsten Gesamtpreise, wenn Sie planen, Ihren reservierten Knotenpunkt nur unregelmässig zu nutzen, d. h. mehr als etwa 15 % einer dreijährigen Laufzeit.
On-Demand-Verwendung (Keine reservierten Knoten)	Keine	Höchste Gebühr pro Stunde. Fällt nur immer dann an, wenn der Knoten ausgeführt wird.	Höchste Kosten pro Stunde.

Weitere Informationen finden Sie unter [ElastiCache Amazon-Preise](#).

Abrufen von Informationen über Angebote für reservierte Knoten

Vor dem Kauf von reservierten Knoten können Sie Informationen zu verfügbaren Angeboten für reservierte Knoten erhalten.

Die folgenden Beispiele zeigen, wie Sie mit der, und ElastiCache API Preise und Informationen zu verfügbaren Reserved Node-Angeboten abrufen können. AWS-Managementkonsole AWS CLI

Themen

- [Abrufen von Informationen über Angebote für reservierte Knoten \(Konsole\)](#)
- [Abrufen von Informationen über Angebote für reservierte Knoten \(AWS CLI\)](#)
- [Informationen über Reserved Node-Angebote \(ElastiCache API\) abrufen](#)

Abrufen von Informationen über Angebote für reservierte Knoten (Konsole)

Verwenden Sie das folgende Verfahren, um Preise und andere Informationen zu verfügbaren reservierten Cluster-Angeboten mit dem zu erhalten. AWS-Managementkonsole

So erhalten Sie Informationen zu verfügbaren Angeboten für reservierte Knoten

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Reserved Nodes (Reserved Nodes) aus.
3. Wählen Sie Purchase Reserved Node (Reservierten Knoten kaufen) aus.
4. Wählen Sie für Engine entweder Valkey, Memcached oder Redis OSS aus.
5. Treffen Sie zur Ermittlung der verfügbaren Angebote eine Auswahl für die folgenden Optionen:
 - Node Type
 - Laufzeit
 - Angebotstyp

Nachdem Sie Ihre Auswahl getroffen haben, werden die Kosten pro Knoten und die Gesamtknoten der ausgewählten Optionen unter Reservation details (Reservierungsdetails) angezeigt.

6. Wählen Sie Cancel, um den Kaufvorgang für diese Knoten abubrechen und Kosten zu vermeiden.

Abrufen von Informationen über Angebote für reservierte Knoten (AWS CLI)

Um Preise und andere Informationen zu verfügbaren Angeboten für reservierte Knoten für Valkey oder Redis OSS zu erhalten, geben Sie an der Befehlszeile den folgenden Befehl ein:

```
aws elasticache describe-reserved-cache-nodes-offerings
```

Dieser Vorgang erzeugt eine Ausgabe ähnlich der folgenden (JSON-Format):

```
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "All Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.X,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.xlarge",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "Partial Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.XXX,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 31536000,
```

```

    "FixedPrice": X.X,
    "UsagePrice": X.X,
    "ProductDescription": "redis",
    "OfferingType": "No Upfront",
    "RecurringCharges": [
      {
        "RecurringChargeAmount": X.XXX,
        "RecurringChargeFrequency": "Hourly"
      }
    ]
  }
}

```

Geben Sie an der Befehlszeile den folgenden Befehl ein, um Preise und andere Informationen zu verfügbaren Reserved Node-Angeboten für Memcached zu erhalten:

```

{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "memcached",
  "OfferingType": "All Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.X,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.xlarge",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "memcached",
  "OfferingType": "Partial Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.XXXX,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
}

```

```

    ],
    {
        "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
        "CacheNodeType": "cache.xx.12xlarge",
        "Duration": 31536000,
        "FixedPrice": X.X,
        "UsagePrice": X.X,
        "ProductDescription": "memcached",
        "OfferingType": "No Upfront",
        "RecurringCharges": [
            {
                "RecurringChargeAmount": X.XXXX,
                "RecurringChargeFrequency": "Hourly"
            }
        ]
    }
}

```

Weitere Informationen finden Sie in der Referenz unter [describe-reserved-cache-nodes-offers](#). AWS CLI

Informationen über Reserved Node-Angebote (ElastiCache API) abrufen

Um Preisinformationen und Informationen zu verfügbaren Angeboten für reservierte Knoten zu erhalten, rufen Sie die DescribeReservedCacheNodesOfferings-Aktion auf.

Example

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReservedCacheNodesOfferings
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>

```

Weitere Informationen finden Sie [DescribeReservedCacheNodesOfferings](#) in der ElastiCache API-Referenz.

Kauf eines reservierten Knotens

Die folgenden Beispiele zeigen, wie Sie mithilfe der API AWS-Managementkonsole, der und der ElastiCache API ein Angebot für reservierte Knoten erwerben können. AWS CLI

Important

Wenn Sie den Beispielen in diesem Abschnitt folgen, fallen Gebühren auf Ihrem AWS Konto an, die Sie nicht rückgängig machen können.

Themen

- [Kauf eines reservierten Knotens \(Konsole\)](#)
- [Kauf eines reservierten Knotens \(AWS CLI\)](#)
- [Kauf eines reservierten Knotens \(ElastiCache API\)](#)

Kauf eines reservierten Knotens (Konsole)

Das folgende Beispiel zeigt den Kauf eines bestimmten reservierten Knotens, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, mit der reservierten Knoten-ID myreservationID.

Das folgende Verfahren verwendet die AWS-Managementkonsole, um das Angebot für reservierte Knoten zu erwerben, indem die ID angeboten wird.

So kaufen Sie reservierte Knoten

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie in der Navigationsliste den Link Reserved Nodes (Reservierte Knoten) aus.
3. Wählen Sie die Schaltfläche Purchase Reserved Nodes (Reservierte Knoten kaufen) aus.
4. Wählen Sie für Engine Valkey, Memcached oder Redis OSS aus.
5. Treffen Sie zur Ermittlung der verfügbaren Angebote eine Auswahl für die folgenden Optionen:
 - Node Type
 - Laufzeit
 - Angebotstyp
 - Eine optionale Reserved-Node-ID

Nachdem Sie Ihre Auswahl getroffen haben, werden die Kosten pro Knoten und die Gesamtknoten der ausgewählten Optionen unter Reservation details (Reservierungsdetails) angezeigt.

6. Klicken Sie auf Purchase (Kaufen).

Kauf eines reservierten Knotens (AWS CLI)

Das folgende Beispiel zeigt den Erwerb des bestimmten reservierten Clusters, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, mit der Knoten-ID myreservationID.

Geben Sie den folgenden Befehl an einer Eingabeaufforderung ein:

Für Linux, macOS oder Unix:

```
aws elasticache purchase-reserved-cache-nodes-offering \
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \
  --reserved-cache-node-id myreservationID
```

Für Windows:

```
aws elasticache purchase-reserved-cache-nodes-offering ^
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^
  --reserved-cache-node-id myreservationID
```

Daraufhin erhalten Sie ein Ergebnis, das dem hier dargestellten entspricht:

RESERVATION	ReservationId	Class	Start Time	Duration
Fixed Price	Usage Price	Count	State	Description
Offering Type				
RESERVATION	myreservationid	cache.xx.small	2013-12-19T00:30:23.247Z	1y
XXX.XX USD	X.XXX USD	1	payment-pending	memcached
				Medium Utilization

Weitere Informationen finden Sie unter [purchase-reserved-cache-nodes-offering](#) in der AWS CLI Referenz.

Kauf eines reservierten Knotens (ElastiCache API)

Das folgende Beispiel zeigt den Kauf des bestimmten reservierten Knotens, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, mit der reservierten Cluster-ID myreservationID.

Rufen Sie die `PurchaseReservedCacheNodesOffering`-Operation mit folgenden Parametern auf:

- `ReservedCacheNodesOfferingId` = 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f
- `ReservedCacheNodeID` = myreservationID
- `CacheNodeCount` = 1

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=PurchaseReservedCacheNodesOffering  
  &ReservedCacheNodesOfferingId=649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  
  &ReservedCacheNodeID=myreservationID  
  &CacheNodeCount=1  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie [PurchaseReservedCacheNodesOffering](#) in der ElastiCache API-Referenz.

Abrufen von Informationen über Ihre reservierten Knoten

Informationen zu den reservierten Knoten, die Sie gekauft haben AWS-Managementkonsole, erhalten Sie mit der AWS CLI, und der ElastiCache API.

Themen

- [Abrufen von Informationen über Ihre reservierten Knoten \(Konsole\)](#)
- [Abrufen von Informationen über Ihre reservierten Knoten \(AWS CLI\)](#)
- [Informationen über Ihre reservierten Knoten \(ElastiCache API\) abrufen](#)

Abrufen von Informationen über Ihre reservierten Knoten (Konsole)

Das folgende Verfahren beschreibt, wie Sie mit dem Informationen AWS-Managementkonsole zu den von Ihnen gekauften reservierten Knoten abrufen können.

So rufen Sie Informationen über Ihre gekauften reservierten Knoten ab

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie in der Navigationsliste den Link Reserved Nodes (Reservierte Knoten) aus.

Die reservierten Knoten für Ihr Konto werden in der Liste "Reserved Nodes" (Reservierte Knoten) aufgeführt. Sie können einen beliebigen reservierten Knoten in der Liste auswählen, um sich detaillierte Informationen zu diesem Knoten im Detailbereich am unteren Rand der Konsole anzeigen zu lassen.

Abrufen von Informationen über Ihre reservierten Knoten (AWS CLI)

Um Informationen über reservierte Knoten für Ihr AWS Konto zu erhalten, geben Sie an der Befehlszeile den folgenden Befehl ein:

```
aws elasticache describe-reserved-cache-nodes
```

Dieser Vorgang erzeugt eine Ausgabe ähnlich der folgenden (JSON-Format):

```
{
  "ReservedCacheNodeId": "myreservationid",
  "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
```

```
"CacheNodeType": "cache.xx.small",  
"DataTiering": "disabled",  
"Duration": "31536000",  
"ProductDescription": "memcached",  
"OfferingType": "Medium Utilization",  
"MaxRecords": 0  
}
```

Weitere Informationen finden Sie [describe-reserved-cache-nodes](#) in der AWS CLI Referenz.

Informationen über Ihre reservierten Knoten (ElastiCache API) abrufen

Rufen Sie den DescribeReservedCacheNodes Vorgang auf, um Informationen über reservierte Knoten für Ihr AWS Konto zu erhalten.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReservedCacheNodes  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie [DescribeReservedCacheNodes](#) in der ElastiCache API-Referenz.

Migration von Knoten der vorherigen Generation

Knoten der vorherigen Generation sind Knotentypen, die nicht mehr verwendet werden. Wenn Sie keine vorhandenen Cluster haben, die einen Knotentyp der vorherigen Generation verwenden, ElastiCache unterstützt die Erstellung neuer Cluster mit diesem Knotentyp nicht.

Aufgrund der begrenzten Anzahl von Node-Typen der vorherigen Generation können wir keinen erfolgreichen Ersatz garantieren, wenn ein Node in Ihrem(n) Cluster(n) fehlerhaft wird. In einem solchen Szenario kann die Verfügbarkeit Ihres Clusters negativ beeinflusst werden.

Wir empfehlen, dass Sie Ihre(n) Cluster auf einen neuen Knotentyp migrieren, um die Verfügbarkeit und Leistung zu verbessern. Einen empfohlenen Knotentyp für die Migration finden Sie unter [Upgrade-Pfade](#). Eine vollständige Liste der unterstützten Knotentypen und Knotentypen der vorherigen Generation finden Sie unter [Unterstützte Knotentypen](#). ElastiCache

Migrieren von Knoten auf einem Valkey- oder Redis OSS-Cluster

Das folgende Verfahren beschreibt, wie Sie Ihren Valkey- oder Redis OSS-Cluster-Knotentyp mithilfe der Konsole migrieren. ElastiCache Während dieses Vorgangs bearbeitet Ihr Valkey- oder Redis OSS-Cluster weiterhin Anfragen mit minimaler Ausfallzeit. Abhängig von Ihrer Clusterkonfiguration können folgende Ausfallzeiten auftreten. Die folgenden Angaben sind Schätzungen und können je nach Ihrer spezifischen Konfiguration abweichen:

- Bei deaktiviertem Cluster-Modus (Einzelknoten) kann es zu einer Verzögerung von ca. 60 Sekunden kommen, die in erster Linie auf die DNS-Verbreitung zurückzuführen ist.
- Bei deaktiviertem Clustermodus (mit Replikatknoten) kann es bei Clustern, auf denen Valkey 7.2 und höher oder Redis OSS 5.0.6 und höher ausgeführt wird, etwa 1 Sekunde dauern. Bei allen älteren Versionen kann es zu einer Dauer von etwa 10 Sekunden kommen.
- Bei aktiviertem Cluster-Modus kann etwa 1 Sekunde angezeigt werden.

So ändern Sie einen Valkey- oder Redis OSS-Cluster-Knotentyp mithilfe der Konsole:

1. Melden Sie sich bei der Konsole an und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.
3. Wählen Sie aus der Cluster-Liste den gewünschten Cluster für die Migration aus.
4. Wählen Sie Aktionen und dann Ändern.
5. Wählen Sie den neuen Knotentyp aus der Knotentypliste aus.
6. Wenn Sie den Migrationsprozess sofort durchführen möchten, wählen Sie Sofort anwenden. Wenn die Option Apply immediately (Sofort anwenden) nicht gewählt wird, wird der Migrationsprozess während des nächsten Wartungsfensters des Clusters durchgeführt.
7. Wählen Sie Ändern aus. Wenn Sie im vorherigen Schritt Apply immediately aktiviert haben, ändert sich der Cluster-Status zu modifying. Wenn der Status zu available wechselt, ist die Änderung abgeschlossen und Sie können den neuen Cluster verwenden.

Um einen Valkey- oder Redis OSS-Cluster-Knotentyp zu ändern, verwenden Sie: AWS CLI

Verwenden Sie die [modify-replication-group](#)API wie folgt:

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group /  
--replication-group-id my-replication-group /  
--cache-node-type new-node-type /  
--apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id my-replication-group ^  
--cache-node-type new-node-type ^  
--apply-immediately
```

In diesem Szenario entspricht der Wert von *new-node-type* dem Knotentyp, zu dem Sie migrieren. Durch die Übergabe des `--apply-immediately`-Parameters wird die Aktualisierung sofort angewendet, wenn die Replikationsgruppe vom Status geändert in den Status verfügbar wechselt. Wenn die Option Apply immediately (Sofort anwenden) nicht gewählt wird, wird der Migrationsprozess während des nächsten Wartungsfensters des Clusters durchgeführt.

 Note

Wenn Sie den Cluster nicht ändern können, weil ein `InvalidCacheClusterState`-Fehler aufgetreten ist, müssen Sie zuerst einen fehlerhaften Wiederherstellungsknoten entfernen.

Reparieren oder Entfernen von `restore-failed-node` (en)

Das folgende Verfahren beschreibt, wie Sie Knoten, bei denen die Wiederherstellung fehlgeschlagen ist, aus Ihrem Valkey- oder Redis OSS-Cluster reparieren oder entfernen. Weitere Informationen darüber, wie ElastiCache Knoten in den Status „Wiederherstellung fehlgeschlagen“ übergehen, finden Sie unter [ElastiCache Knotenstatus anzeigen](#). Wir empfehlen, zuerst alle Knoten zu entfernen, die sich in einem Zustand befinden, in dem eine Wiederherstellung fehlgeschlagen ist, dann die verbleibenden Knoten der vorherigen Generation im ElastiCache Cluster auf einen Knotentyp der neueren Generation zu migrieren und schließlich die erforderliche Anzahl von Knoten wieder hinzuzufügen.

So entfernen Sie den Wiederherstellungsfehler-Knoten (Konsole):

1. Melden Sie sich bei der Konsole an und öffnen Sie die Konsole unter ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.
3. Wählen Sie in der Cluster-Liste den Cluster aus, aus dem Sie einen Knoten entfernen möchten.
4. Wählen Sie aus der Shard-Liste den Shard aus, aus dem Sie einen Knoten entfernen möchten. Überspringen Sie diesen Schritt, wenn der Cluster-Modus für den Cluster deaktiviert ist.
5. Wählen Sie aus der Knotenliste den Knoten mit einem Status `restore-failed` aus.
6. Wählen Sie Aktionen und wählen Sie dann Knoten löschen.

Nachdem Sie die Knoten, bei denen die Wiederherstellung fehlgeschlagen ist, aus Ihrem ElastiCache Cluster entfernt haben, können Sie jetzt zu einem neueren Generationstyp migrieren. Weitere Informationen finden Sie weiter oben unter [Migrieren von Knoten auf einem Valkey- oder Redis OSS-Cluster](#).

Informationen zum Hinzufügen von Knoten zu Ihrem ElastiCache Cluster finden Sie unter [Hinzufügen von Knoten zu einem ElastiCache Cluster](#)

Migrieren von Knoten auf einen Memcached-Cluster

Um ElastiCache für Memcached auf einen anderen Knotentyp zu migrieren, müssen Sie einen neuen Cluster erstellen, der immer leer ist und von Ihrer Anwendung aufgefüllt werden kann.

So migrieren Sie Ihren Cluster-Knotentyp ElastiCache for Memcached mithilfe der Konsole:
ElastiCache

- Erstellen Sie einen neuen Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).
- Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Die Endpunkte eines Clusters finden \(Konsole\) \(Memcached\)](#).
- Löschen Sie den alten Cluster. Weitere Informationen finden Sie unter [Löschen eines Clusters in ElastiCache](#).

Verwaltung von Clustern in ElastiCache

Ein Cluster ist eine Sammlung von einem oder mehreren Cache-Knoten, auf denen alle eine Instanz der Valkey-, Memcached- und Redis OSS-Engine-Software ausführen. Wenn Sie einen Cluster erstellen, geben Sie die Engine und die Version an, die von allen Knoten verwendet werden soll.

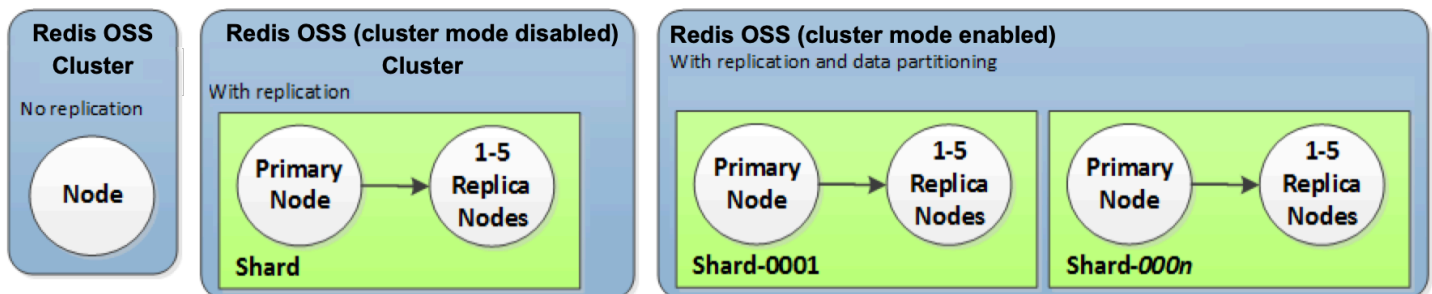
Valkey- und Redis OSS-Cluster

Das folgende Diagramm zeigt einen typischen Valkey- oder Redis OSS-Cluster. Diese Cluster können einen einzelnen Knoten oder bis zu sechs Knoten innerhalb eines Shard (API/CLI: Node Group) enthalten. Ein Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) mit einem Knoten hat keinen Shard und ein Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) mit mehreren Knoten hat einen einzigen Shard. Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) können bis zu 500 Shards haben, wobei Ihre Daten auf die Shards verteilt sind. Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Engine-Version Valkey 7.2 und höher oder Redis OSS 5.0.6 und höher ist. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#). Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

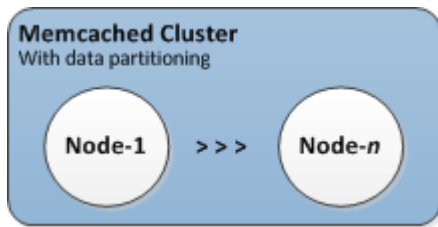
Wenn Sie mehrere Knoten in einem Valkey- oder Redis OSS-Shard haben, ist einer der Knoten ein primärer Knoten. read/write Alle anderen Knoten im Shard sind schreibgeschützte Replikate.

Typische Valkey- oder Redis OSS-Cluster sehen wie folgt aus.



Memcached-Cluster

Typische Memcached-Cluster sehen wie folgt aus. Memcached-Cluster enthalten 1 bis 60 Knoten, über die Sie Ihre Daten horizontal partitionieren.



Elasticache-Operationen für Valkey, Memcached und Redis OSS

Die meisten ElastiCache Operationen werden auf Clusterebene ausgeführt. Sie können einen Cluster mit einer bestimmten Anzahl von Knoten und einer Parametergruppe einrichten, die die Eigenschaften für jeden Knoten steuert. Alle Knoten innerhalb eines Clusters gehören demselben Knotentyp an und verfügen über die gleichen Einstellungen für Parameter und Sicherheitsgruppen.

Jeder Cluster muss über eine Cluster-Kennung verfügen. Die Cluster-Kennung ist ein vom Kunden angegebener Name für den Cluster. Diese Kennung gibt bei der Interaktion mit der ElastiCache API und den AWS CLI Befehlen einen bestimmten Cluster an. Die Cluster-ID muss für diesen Kunden in einer AWS Region eindeutig sein.

ElastiCache unterstützt mehrere Engine-Versionen. Sofern es keine bestimmten Gründe dagegen gibt, raten wir dazu, die neueste Version zu verwenden.

ElastiCache Cluster sind für den Zugriff über eine EC2 Amazon-Instance konzipiert. Wenn Sie Ihren Cluster in einer Virtual Private Cloud (VPC) starten, die auf dem Amazon VPC-Service basiert, können Sie von außerhalb AWS auf ihn zugreifen. Weitere Informationen finden Sie unter [Zugriff auf ElastiCache Ressourcen von außen AWS](#).

Eine Liste der unterstützten Versionen finden Sie unter [Unterstützte Engines und Versionen](#), [Unterstützte Redis OSS-Engine-Versionen](#), und [Wird ElastiCache für Memcached-Versionen unterstützt](#).

Auswahl eines Netzwerktyps in ElastiCache

ElastiCache unterstützt die Internetprotokoll-Versionen 4 und 6 (IPv4 und IPv6), sodass Sie Ihren Cluster so konfigurieren können, dass er Folgendes akzeptiert:

- nur IPv4 Verbindungen,
- nur IPv6 Verbindungen,
- IPv4 sowohl als auch IPv6 Verbindungen (Dual-Stack)

IPv6 [wird für Workloads unterstützt, die Valkey 7.2 und höher oder Redis OSS 6.2 und höher verwenden, auf allen Instanzen, die auf dem Nitro-System basieren](#). Für den Zugriff über fallen keine zusätzlichen Gebühren an. ElastiCache IPv6

Note

Die Migration von Clustern, die vor der Verfügbarkeit von IPV6 /Dual-Stack erstellt wurden, wird nicht unterstützt. Das Umschalten zwischen Netzwerktypen in neu erstellten Clustern wird ebenfalls nicht unterstützt.

IPv6 [wird für Workloads unterstützt, die Memcached 1.6.6 oder höher auf allen auf dem Nitro-System basierenden Instanzen verwenden](#). Für den Zugriff über fallen keine zusätzlichen Gebühren an. ElastiCache IPv6

Konfigurieren von Subnetzen für den Netzwerktyp

Wenn Sie einen Cluster in einer Amazon VPC erstellen, müssen Sie eine Subnetzgruppe angeben. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten zugeordnet werden sollen. ElastiCache Cluster benötigen ein Dual-Stack-Subnetz mit beiden IPv4 und ihnen zugewiesenen IPv6 Adressen, um im Dual-Stack-Modus betrieben werden zu können, und ein Subnetz nur für den Betrieb als IPv6 -only. IPv6

Verwenden von Dual-Stack

ElastiCache Bei Verwendung von Redis OSS im aktivierten Clustermodus unterscheidet sich die Verbindung mit allen Clusterknoten über den Konfigurationsendpunkt aus Sicht einer Anwendung nicht von der direkten Verbindung zu einem einzelnen Cache-Knoten. Um dies zu

erreichen, muss ein clusterfähiger Client einen Cluster-Erkennungsprozess durchführen und die Konfigurationsinformationen für alle Knoten anfordern. Das Erkennungsprotokoll von Redis unterstützt nur eine IP pro Knoten.

Wenn Sie einen Cluster mit ElastiCache for Memcached erstellen und Dual-Stack als Netzwerktyp wählen, müssen Sie anschließend einen IP-Erkennungstyp festlegen — entweder IPv4 oder IPv6. ElastiCache verwendet standardmäßig den Netzwerktyp und die IP-Erkennung IPv6, aber das kann geändert werden. Wenn Sie Auto Discovery verwenden, werden nur die IP-Adressen des von Ihnen gewählten IP-Typs an den Memcached-Client zurückgegeben. Weitere Informationen finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#).

Um die Abwärtskompatibilität mit allen vorhandenen Clients aufrechtzuerhalten, wurde die IP-Erkennung eingeführt, sodass Sie den IP-Typ (d. h. IPv4 oder IPv6) auswählen können, der im Discovery-Protokoll angekündigt werden soll. Dadurch wird die auto Erkennung zwar auf nur einen IP-Typ beschränkt, aber Dual-Stack ist für Workloads mit aktiviertem Clustermodus dennoch von Vorteil, da es Migrationen (oder Rollbacks) von einem IPv4 zu einem IPv6 Discovery-IP-Typ ohne Ausfallzeiten ermöglicht.

TLS-fähige Dual-Stack-Cluster ElastiCache

Wenn TLS für ElastiCache Cluster aktiviert ist, geben Cluster-Erkennungsfunktionen wie `cluster slotscluster shards`, und `cluster nodes` mit Valkey oder Redis OSS und `config get cluster` mit Memcached Hostnamen statt. IPs. Die Hostnamen werden dann verwendet, anstatt eine Verbindung IPs zum ElastiCache Cluster herzustellen und einen TLS-Handshake durchzuführen. Das bedeutet, dass Clients nicht vom IP-Discovery-Parameter betroffen sind. Bei TLS-fähigen Clustern hat der IP Discovery-Parameter keine Auswirkung auf das bevorzugte IP-Protokoll. Stattdessen wird das verwendete IP-Protokoll dadurch bestimmt, welches IP-Protokoll der Client bei der Auflösung von DNS-Hostnamen bevorzugt.

Beispiele zur Konfiguration einer IP-Protokollpräferenz bei der Auflösung von DNS-Hostnamen finden Sie unter [TLS-fähige ElastiCache Dual-Stack-Cluster](#).

Verwenden von AWS-Managementkonsole (Valkey und Redis OSS)

Wenn Sie einen Cluster mit der AWS-Managementkonsole erstellen, wählen Sie unter Konnektivität entweder IPv4, einen Netzwerktyp IPv6 oder Dual-Stack. Wenn Sie einen Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) erstellen und Dual-Stack wählen, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

Für weitere Informationen siehe [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#) oder [Erstellen eines Valkey- oder Redis-OSS \(Clustermodus deaktiviert\) \(Konsole\)](#).

Wählen Sie beim Erstellen einer Replikationsgruppe mit dem AWS-Managementkonsole einen Netzwerktyp IPv4, entweder IPv6 oder Dual-Stack. Wenn Sie sich für Dual-Stack entscheiden, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

Für weitere Informationen siehe [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Cluster Mode Disabled\) von Grund auf neu erstellen](#) oder [Eine Replikationsgruppe in Valkey oder Redis OSS \(Cluster Mode Enabled\) von Grund auf neu erstellen](#).

Verwenden von AWS-Managementkonsole(Memcached)

Wenn Sie einen Cluster mit dem erstellen AWS-Managementkonsole, wählen Sie unter Konnektivität entweder IPv4 einen Netzwerktyp IPv6 oder Dual-Stack. Wenn Sie sich für Dual-Stack entscheiden, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

Verwenden der CLI mit Valkey, Memcached oder Redis OSS.

Redis OSS

Wenn Sie einen Cluster mit Valkey oder Redis OSS mithilfe der CLI erstellen, verwenden Sie den [create-cache-cluster](#) Befehl und geben die NetworkType Parameter und an: IPDiscovery

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \
  --cache-cluster-id "cluster-test" \
  --engine redis \
  --cache-node-type cache.m5.large \
  --num-cache-nodes 1 \
  --network-type dual_stack \
  --ip-discovery ipv4
```

Für Windows:

```
aws elasticache create-cache-cluster ^
  --cache-cluster-id "cluster-test" ^
```

```
--engine redis ^  
--cache-node-type cache.m5.large ^  
--num-cache-nodes 1 ^  
--network-type dual_stack ^  
--ip-discovery ipv4
```

Wenn Sie mithilfe der CLI eine Replikationsgruppe mit deaktiviertem Clustermodus erstellen, verwenden Sie den [create-replication-group](#) Befehl und geben die IPDiscovery Parameter NetworkType und an:

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --primary-cluster-id redis01 \  
  --network-type dual_stack \  
  --ip-discovery ipv4
```

Für Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "demo cluster with replicas" ^  
  --num-cache-clusters 3 ^  
  --primary-cluster-id redis01 ^  
  --network-type dual_stack ^  
  --ip-discovery ipv4
```

Wenn Sie eine Replikationsgruppe mit aktiviertem Clustermodus erstellen und IPv4 für die IP-Erkennung mithilfe der CLI verwenden, verwenden Sie den [create-replication-group](#) Befehl und geben die IPDiscovery Parameter NetworkType und an:

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id demo-cluster \  
  --replication-group-description "demo cluster" \  
  --engine redis ^  
  --cache-node-type cache.m5.large ^  
  --num-cache-nodes 1 ^  
  --network-type dual_stack ^  
  --ip-discovery ipv4
```

```
--cache-node-type cache.m5.large \  
--num-node-groups 2 \  
--engine redis \  
--cache-subnet-group-name xyz \  
--network-type dual_stack \  
--ip-discovery ipv4 \  
--region us-east-1
```

Für Windows:

```
aws elasticache create-replication-group ^  
--replication-group-id demo-cluster ^  
--replication-group-description "demo cluster" ^  
--cache-node-type cache.m5.large ^  
--num-node-groups 2 ^  
--engine redis ^  
--cache-subnet-group-name xyz ^  
--network-type dual_stack ^  
--ip-discovery ipv4 ^  
--region us-east-1
```

Wenn Sie eine Replikationsgruppe mit aktiviertem Clustermodus erstellen und IPv6 für die IP-Erkennung mithilfe der CLI verwenden, verwenden Sie den [create-replication-group](#) Befehl und geben die IPDiscovery Parameter NetworkType und an:

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
--replication-group-id demo-cluster \  
--replication-group-description "demo cluster" \  
--cache-node-type cache.m5.large \  
--num-node-groups 2 \  
--engine redis \  
--cache-subnet-group-name xyz \  
--network-type dual_stack \  
--ip-discovery ipv6 \  
--region us-east-1
```

Für Windows:

```
aws elasticache create-replication-group ^  
--replication-group-id demo-cluster ^
```

```
--replication-group-description "demo cluster" ^
--cache-node-type cache.m5.large ^
--num-node-groups 2 ^
--engine redis ^
--cache-subnet-group-name xyz ^
--network-type dual_stack ^
--ip-discovery ipv6 ^
--region us-east-1
```

Memcached

Wenn Sie mithilfe der CLI einen Cluster mit Memcached erstellen, verwenden Sie den [create-cache-cluster](#) Befehl und geben die NetworkType Parameter und an: IPDiscovery

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \
  --cache-cluster-id "cluster-test" \
  --engine memcached \
  --cache-node-type cache.m5.large \
  --num-cache-nodes 1 \
  --network-type dual_stack \
  --ip-discovery ipv4
```

Für Windows:

```
aws elasticache create-cache-cluster ^
  --cache-cluster-id "cluster-test" ^
  --engine memcached ^
  --cache-node-type cache.m5.large ^
  --num-cache-nodes 1 ^
  --network-type dual_stack ^
  --ip-discovery ipv4
```

Identifizieren Sie automatisch Knoten in Ihrem Cluster (Memcached)

ElastiCache unterstützt bei Clustern, auf denen die Memcached-Engine ausgeführt wird, Auto Discovery — die Fähigkeit von Client-Programmen, automatisch alle Knoten in einem Cluster zu identifizieren und Verbindungen zu all diesen Knoten zu initiieren und aufrechtzuerhalten.

 Note

Auto Discovery wurde für Cluster hinzugefügt, die auf Amazon ElastiCache Memcached ausgeführt werden. Auto Discovery ist für Valkey- oder Redis OSS-Engines nicht verfügbar.

Mit Auto Discovery muss Ihre Anwendung keine manuelle Verbindung mit einzelnen Cache-Knoten herstellen. Stattdessen stellt sie eine Verbindung mit einem Memcached-Knoten her und ruft die Liste der Knoten ab. Anhand dieser Liste erhält Ihre Anwendung Informationen über die restlichen Knoten im Cluster und kann eine Verbindung mit einem beliebigen Knoten herstellen. Die einzelnen Endpunkte der Cache-Knoten in Ihrer Anwendung müssen nicht als vordefinierter Code aufgenommen werden.

Wenn Sie in Ihrem Cluster einen Dual-Stack-Netzwerktyp verwenden, gibt Auto Discovery nur IPv4 oder IPv6 Adressen zurück, je nachdem, welche Sie auswählen. Weitere Informationen finden Sie unter [Auswahl eines Netzwerktyps in ElastiCache](#).

Alle Cache-Knoten im Cluster führen eine Liste mit Metadaten über alle anderen Knoten. Diese Metadaten werden aktualisiert, sobald Knoten hinzugefügt oder aus dem Cluster entfernt werden.

Themen

- [Vorteile von Auto Discovery mit Memcached](#)
- [Funktionsweise von Auto Discovery](#)
- [Verwenden von Auto Discovery](#)
- [Manuelles Herstellen einer Verbindung zu Memcache-Cache-Knoten](#)
- [Hinzufügen von Auto Discovery zu Ihrer Memcached-Clientbibliothek](#)
- [ElastiCache Kunden mit auto Erkennung](#)

Vorteile von Auto Discovery mit Memcached

Bei der Verwendung von Memcached bietet Auto Discovery die folgenden Vorteile:

- Wenn Sie die Anzahl der Knoten in einem Cluster erhöhen, registrieren sich die neuen Knoten selbst beim Konfigurationsendpunkt und bei allen anderen Knoten. Wenn Sie Knoten aus dem Cache-Cluster entfernen, heben die entsprechenden Cluster ihre Registrierung selbst auf. In beiden Fällen werden alle anderen Knoten im Cluster mit den neuen Cache-Knotenmetadaten aktualisiert.
- Ausfälle von Cache-Knoten werden automatisch erkannt und ausgefallene Knoten automatisch ersetzt.

Note

Der Knoten fällt solange aus, bis er ersetzt wird.

- Ein Client-Programm muss nur eine Verbindung mit dem Konfigurationsendpunkt herstellen. Anschließend stellt die Auto Discovery-Bibliothek eine Verbindung mit allen anderen Knoten im Cluster her.
- Client-Programme fragen den Cluster einmal pro Minute ab (dieses Intervall kann bei Bedarf geändert werden). Wenn Änderungen an der Cluster-Konfiguration vorgenommen wurden, z. B. neue oder gelöschte Knoten, erhält der Client eine aktualisierte Liste mit Metadaten. Anschließend stellt der Client je nach Bedarf eine Verbindung mit diesen Knoten her bzw. trennt die Verbindung.

Auto Discovery ist auf allen ElastiCache Memcached-Clustern aktiviert. Sie müssen Ihre Cache-Knoten nicht neu starten, um diese Funktion zu verwenden.

Funktionsweise von Auto Discovery

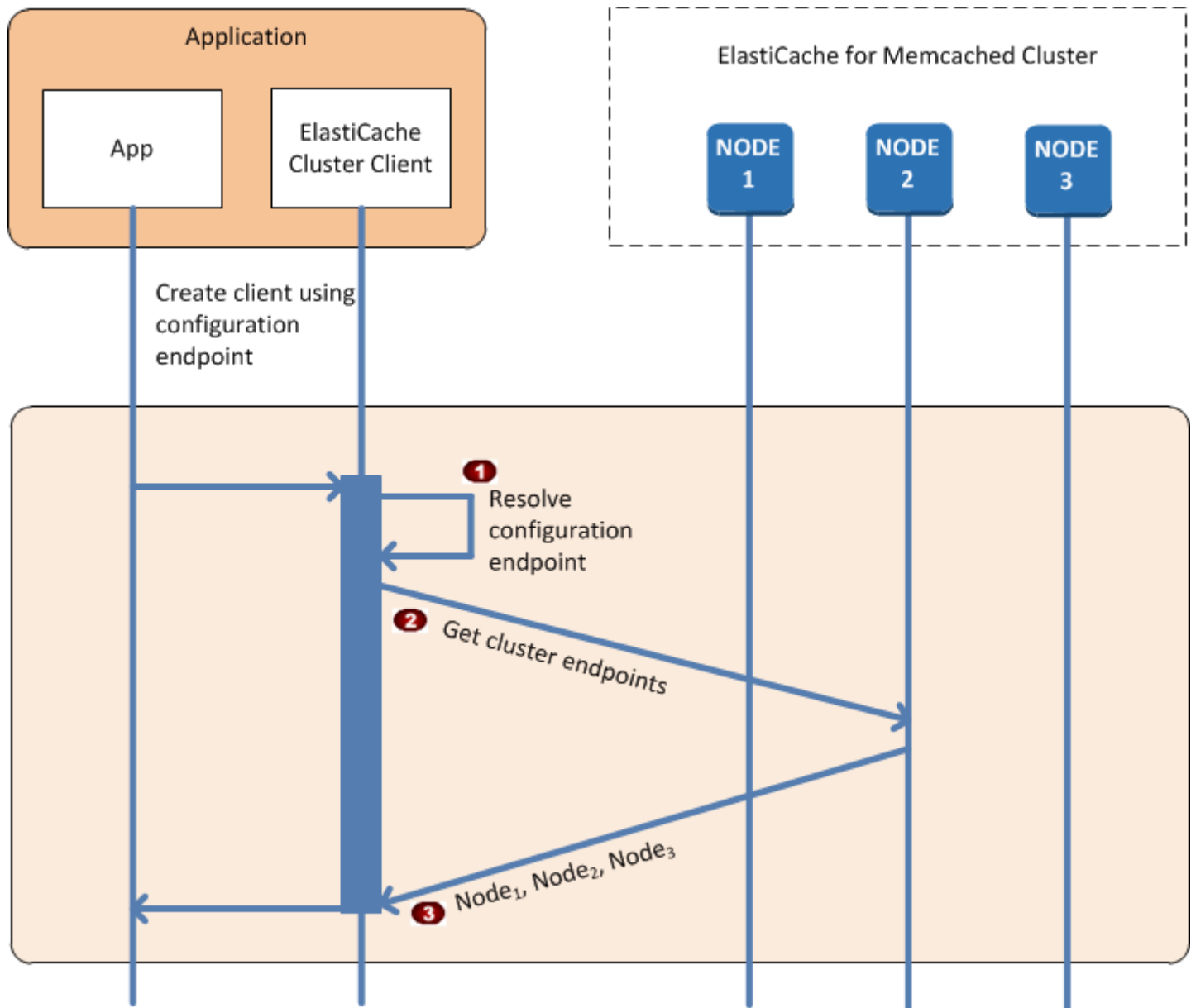
Themen

- [Herstellen von Verbindungen mit Cache-Knoten](#)
- [Normale Cluster-Operationen](#)
- [Weitere Operationen in](#)

In diesem Abschnitt wird beschrieben, wie Clientanwendungen den ElastiCache Clusterclient verwenden, um Cache-Knotenverbindungen zu verwalten und mit Datenelementen im Cache zu interagieren.

Herstellen von Verbindungen mit Cache-Knoten

Von der Anwendungsseite aus betrachtet, unterscheidet sich das Herstellen einer Verbindung mit dem Cluster-Konfigurationsendpunkt nicht vom Herstellen einer direkten Verbindung mit einem einzelnen Cache-Knoten. Das folgende Ablaufdiagramm zeigt den Prozess zum Herstellen der Verbindung mit Cache-Knoten.



Prozess zum Herstellen von Verbindungen mit Cache-Knoten

- Die Anwendung löst den DNS-Namen des Konfigurationsendpunkts auf. Da der Konfigurationsendpunkt CNAME-Einträge für alle Cache-Knoten verwaltet, wird der DNS-Name in einen der Knoten aufgelöst. Der Client kann dann eine Verbindung mit diesem Knoten herstellen.
- Der Client fordert die Konfigurationsinformationen für alle anderen Knoten an. Da jeder Knoten Konfigurationsinformationen für alle Knoten im Cluster verwaltet, kann jeder Knoten auf Anfrage Konfigurationsinformationen an den Client weitergeben.

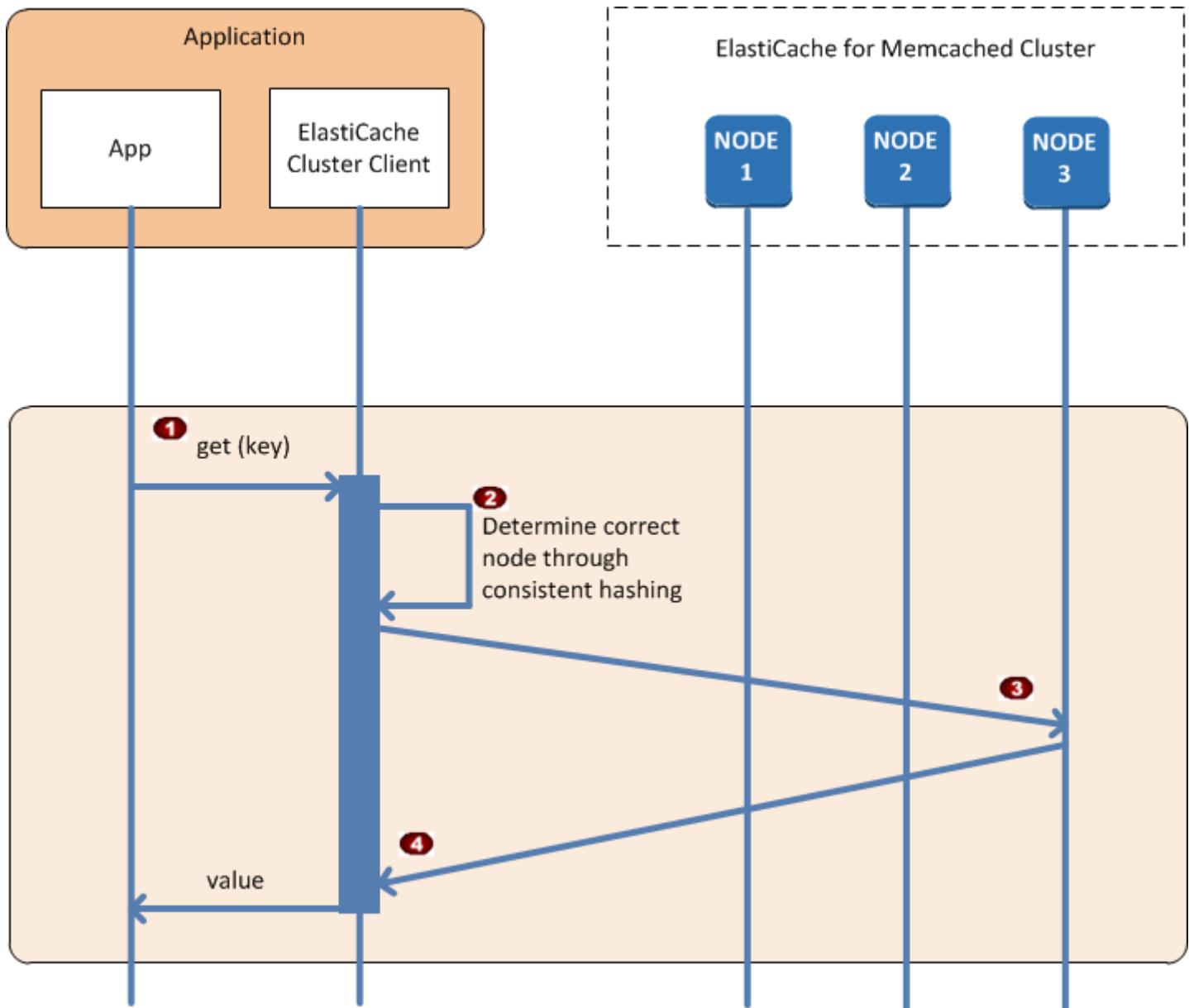
- Der Client empfängt die aktuelle Liste mit den Hostnamen und IP-Adressen der Cache-Knoten. Anschließend kann er eine Verbindung mit allen anderen Knoten im Cluster herstellen.

Note

Das Client-Programm aktualisiert seine Liste mit den Hostnamen und IP-Adressen der Cache-Knoten einmal pro Minute. Dieses Abfrageintervall kann bei Bedarf angepasst werden.

Normale Cluster-Operationen

Wenn die Anwendung eine Verbindung zu allen Cache-Knoten hergestellt hat, bestimmt der ElastiCache Cluster-Client, auf welchen Knoten einzelne Datenelemente gespeichert werden sollen und welche Knoten später nach diesen Datenelementen abgefragt werden sollen. Das folgende Ablaufdiagramm zeigt den Prozess für normale Cluster-Operationen.



Prozess für normale Cluster-Operationen

- Die Anwendung erstellt eine get-Anforderung für ein bestimmtes Datenelement, das durch seinen Schlüssel identifiziert wird.
- Der Client verwendet einen Hashing-Algorithmus für den Schlüssel, um den Cache-Knoten zu ermitteln, der das Datenelement enthält.
- Das Datenelement wird vom entsprechenden Knoten angefordert.
- Das Datenelement wird an die Anwendung zurückgegeben.

Weitere Operationen in

In manchen Situationen kann es vorkommen, dass Sie eine Änderung an den Knoten eines Clusters vornehmen. So können Sie beispielsweise einen zusätzlichen Knoten hinzufügen, um zusätzlichen Bedarf zu decken, oder einen Knoten löschen, um in Zeiten geringerer Nachfrage Geld zu sparen. Oder Sie ersetzen einen Knoten, weil er auf die eine oder andere Weise ausgefallen ist.

Bei einer Änderung im Cluster, die eine Aktualisierung der Metadaten auf die Endpunkte des Clusters erfordert, wird diese Änderung auf allen Knoten gleichzeitig vorgenommen. So sind die Metadaten in einem bestimmten Knoten konsistent mit den Metadaten in allen anderen Knoten im Cluster.

In jedem Fall sind die Metadaten für alle Knoten jederzeit konsistent, da sie für alle Knoten im Cluster gleichzeitig aktualisiert werden. Verwenden Sie immer den Konfigurationsendpunkt, um die Endpunkte der verschiedenen Knoten im Cluster zu erhalten. Durch Verwenden des Konfigurationsendpunkts stellen Sie sicher, dass Sie keine Endpunktdaten von einem Knoten erhalten, der "verschwindet".

Hinzufügen eines Knotens

Während der Zeit, in der der Knoten eingerichtet wird, ist sein Endpunkt nicht in den Metadaten enthalten. Sobald der Knoten verfügbar ist, wird er den jeweiligen Metadaten der Cluster-Knoten hinzugefügt. In diesem Szenario sind die Metadaten für alle Knoten konsistent und Sie können mit dem neuen Knoten erst interagieren, sobald er verfügbar ist. Vorher liegen Ihnen keine Informationen darüber vor und Sie interagieren mit den Knoten in Ihrem Cluster so, als ob der neue Knoten nicht vorhanden wäre.

Löschen eines Knotens

Wenn ein Knoten entfernt wird, wird sein Endpunkt erst in den Metadaten gelöscht und anschließend wird der Knoten aus dem Cluster entfernt. In diesem Szenario sind die Metadaten in allen Knoten konsistent und zu keiner Zeit enthalten sie den Endpunkt für den Knoten, der entfernt werden soll, während der Knoten nicht verfügbar ist. Während des Zeitraums, in dem der Knoten entfernt wird, erfolgt keine Meldung in den Metadaten. Demzufolge interagiert Ihre Anwendung nur mit den verbleibenden $n-1$ Knoten, so als ob der Knoten nicht vorhanden wäre.

Ersetzen eines Knotens

Wenn ein Knoten ausfällt, wird ElastiCache dieser Knoten heruntergefahren und ein Ersatzknoten installiert. Das Ersetzen dauert einige Minuten. Während dieser Zeit zeigen die Metadaten in allen

Knoten den Endpunkt für den ausgefallenen Knoten noch an, doch jeder Versuch, mit dem Knoten zu interagieren, schlägt fehl. Daher sollte Ihre Logik immer Logik für Wiederholungsversuche umfassen.

Verwenden von Auto Discovery

Gehen Sie wie folgt vor, um Auto Discovery ElastiCache für Memcached zu verwenden:

- [Rufen Sie den Konfigurationsendpunkt ab](#)
- [Laden Sie den ElastiCache Cluster-Client herunter](#)
- [Ändern Sie Ihr Anwendungsprogramm](#)

Rufen Sie den Konfigurationsendpunkt ab

Zum Herstellen der Verbindung mit einem Cluster müssen die Client-Programme den Cluster-Konfigurationsendpunkt kennen. Weitere Informationen finden Sie im Thema [Die Endpunkte eines Clusters finden \(Konsole\) \(Memcached\)](#).

Sie können auch den Befehl `aws elasticache describe-cache-clusters` mit dem Parameter `--show-cache-node-info` verwenden:

Unabhängig von der Methode, mit der Sie die Endpunkte des Clusters suchen, enthält der Konfigurationsendpunkt in seiner Adresse immer `.cfg`.

Example Finden von Endpunkten mit dem for AWS CLI ElastiCache

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id mycluster \  
  --show-cache-node-info
```

Für Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

Dieser Vorgang erzeugt eine Ausgabe ähnlich der folgenden (JSON-Format):

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  

```

```

    {
      "CacheNodeId": "0001",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.fnjyzo.cfg.0001.use1.cache.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
      "CustomerAvailabilityZone": "us-east-1e"
    },
    {
      "CacheNodeId": "0002",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.fnjyzo.cfg.0002.use1.cache.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
      "CustomerAvailabilityZone": "us-east-1a"
    }
  ],
  "CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "CacheParameterGroupName": "default.memcached1.4",
    "ParameterApplyStatus": "in-sync"
  },
  "CacheClusterId": "mycluster",
  "PreferredAvailabilityZone": "Multiple",
  "ConfigurationEndpoint": {
    "Port": 11211,
    "Address": "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
  },
  "CacheSecurityGroups": [],
  "CacheClusterCreateTime": "2016-10-12T21:39:28.001Z",
  "AutoMinorVersionUpgrade": true,
  "CacheClusterStatus": "available",
  "NumCacheNodes": 2,
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
  "CacheSubnetGroupName": "default",
  "EngineVersion": "1.4.24",
  "PendingModifiedValues": {},

```



```
        "PreferredMaintenanceWindow": "sat:06:00-sat:07:00",  
        "CacheNodeType": "cache.r3.large"  
    }  
]  
}
```

Laden Sie den ElastiCache Cluster-Client herunter

Um Auto Discovery nutzen zu können, müssen Client-Programme den ElastiCache-Cluster-Client verwenden. Der ElastiCache Cluster-Client ist für Java, PHP und .NET verfügbar und enthält die gesamte erforderliche Logik, um all Ihre Cache-Knoten zu erkennen und eine Verbindung zu ihnen herzustellen.

Um den ElastiCache Cluster-Client herunterzuladen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie in der ElastiCache Konsole ElastiCache Cluster Client und dann Herunterladen aus.

Der Quellcode für den ElastiCache Cluster-Client für Java ist unter <https://github.com/amazonwebservices/aws-elasticache-cluster-client> verfügbarmemcached-for-java. Diese Bibliothek basiert auf dem häufig verwendeten Spymemcached-Client. Der ElastiCache Cluster-Client ist unter der Amazon-Softwarelizenz <https://aws.amazon.com/asl> veröffentlicht. Es steht Ihnen frei, den Quellcode nach eigenen Wünschen zu ändern. Sie können den Code sogar in andere Memcached-Open-Source-Bibliotheken oder in Ihren eigenen Client-Code integrieren.

Note

Um den ElastiCache Cluster Client für PHP verwenden zu können, müssen Sie ihn zunächst auf Ihrer EC2 Amazon-Instance installieren. Weitere Informationen finden Sie unter [Installation des ElastiCache Cluster-Clients für PHP](#).

Für einen von TLS unterstützten Client laden Sie die Binärdatei mit PHP-Version 7.4 oder höher herunter.

Um den ElastiCache Cluster Client für .NET verwenden zu können, müssen Sie ihn zunächst auf Ihrer EC2 Amazon-Instance installieren. Weitere Informationen finden Sie unter [Installation des ElastiCache Clusterclients für .NET](#).

Ändern Sie Ihr Anwendungsprogramm

Ändern Sie Ihr Anwendungsprogramm, sodass es Auto Discovery verwendet. In den folgenden Abschnitten wird gezeigt, wie Sie den ElastiCache Cluster-Client für Java, PHP und .NET verwenden.

Important

Stellen Sie beim Angeben des Cluster-Konfigurationsendpunkts sicher, dass der Endpunkt in seiner Adresse ".cfg" enthält, wie hier dargestellt. Verwenden Sie keinen CNAME oder einen Endpunkt ohne ".cfg" in der Adresse.

```
"mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
```

Wenn Sie den Cluster-Konfigurationsendpunkt nicht explizit angeben, erfolgt die Konfiguration für einen spezifischen Knoten.

Den ElastiCache Cluster-Client für Java verwenden

Das folgende Programm zeigt, wie Sie den ElastiCache Cluster-Client verwenden, um eine Verbindung zu einem Cluster-Konfigurationsendpunkt herzustellen und dem Cache ein Datenelement hinzuzufügen. Mithilfe von Auto Discovery stellt das Programm eine Verbindung mit allen Knoten im Cluster her, ohne dass ein Eingreifen erforderlich ist.

```
package com.amazon.elasticache;

import java.io.IOException;
import java.net.InetSocketAddress;

// Import the &AWS;-provided library with Auto Discovery support
import net.spy.memcached.MemcachedClient;

public class AutoDiscoveryDemo {

    public static void main(String[] args) throws IOException {

        String configEndpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
        Integer clusterPort = 11211;

        MemcachedClient client = new MemcachedClient(
            new InetSocketAddress(configEndpoint,
```

```

        clusterPort));

    // The client will connect to the other cache nodes automatically.

    // Store a data item for an hour.
    // The client will decide which cache host will store this item.
    client.set("theKey", 3600, "This is the data value");
}
}

```

Den ElastiCache Cluster-Client für PHP verwenden

Das folgende Programm zeigt, wie Sie den ElastiCache Cluster-Client verwenden, um eine Verbindung zu einem Cluster-Konfigurationsendpunkt herzustellen und dem Cache ein Datenelement hinzuzufügen. Mithilfe von Auto Discovery stellt das Programm eine Verbindung mit allen Knoten im Cluster her, ohne dass ein Eingreifen erforderlich ist.

Um den ElastiCache Cluster Client für PHP verwenden zu können, müssen Sie ihn zunächst auf Ihrer EC2 Amazon-Instance installieren. Weitere Informationen finden Sie unter [Installation des ElastiCache Cluster-Clients für PHP](#).

```

<?php

/**
 * Sample PHP code to show how to integrate with the Amazon ElastiCache
 * Auto Discovery feature.
 */

/* Configuration endpoint to use to initialize memcached client.
 * This is only an example. */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the ElastiCache cluster.
 * This is only an example */
$server_port = 11211;

/**
 * The following will initialize a Memcached client to utilize the Auto Discovery
 * feature.
 *
 * By configuring the client with the Dynamic client mode with single endpoint, the
 * client will periodically use the configuration endpoint to retrieve the current
 * cache

```

```
* cluster configuration. This allows scaling the cluster up or down in number of
nodes
* without requiring any changes to the PHP application.
*
* By default the Memcached instances are destroyed at the end of the request.
* To create an instance that persists between requests,
*   use persistent_id to specify a unique ID for the instance.
* All instances created with the same persistent_id will share the same connection.
* See http://php.net/manual/en/memcached.construct.php for more information.
*/
$dynamic_client = new Memcached('persistent-id');
$dynamic_client->setOption(Memcached::OPT_CLIENT_MODE,
Memcached::DYNAMIC_CLIENT_MODE);
$dynamic_client->addServer($server_endpoint, $server_port);

/**
 * Store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
$dynamic_client->set('key', 'value', 60);

/**
 * Configuring the client with Static client mode disables the usage of Auto Discovery
 * and the client operates as it did before the introduction of Auto Discovery.
 * The user can then add a list of server endpoints.
 */
$static_client = new Memcached('persistent-id');
$static_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::STATIC_CLIENT_MODE);
$static_client->addServer($server_endpoint, $server_port);

/**
 * Store the data without expiration.
 * The client will decide which cache host will store this item.
 */
$static_client->set('key', 'value');
?>
```

Ein Beispiel zur Verwendung des ElastiCache Cluster-Clients mit aktiviertem TLS finden Sie unter [Verwenden von In-Transit-Verschlüsselung mit PHP und Memcached](#).

Den ElastiCache Cluster-Client für .NET verwenden

Note

ElastiCache Der.NET-Clusterclient ist seit Mai 2022 veraltet.

.NET-Client für ElastiCache ist Open Source unter. <https://github.com/aws-labs/elasticache-cluster-config-net>

Die .NET-Anwendungen erhalten ihre Konfigurationen in der Regel von ihrer Config-Datei. Nachfolgend finden Sie ein Beispiel einer Anwendungs-Config-Datei.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section
      name="clusterclient"
      type="Amazon.ElastiCacheCluster.ClusterConfigSettings,
Amazon.ElastiCacheCluster" />
  </configSections>

  <clusterclient>
    <!-- the hostname and port values are from step 1 above -->
    <endpoint hostname="mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
port="11211" />
  </clusterclient>
</configuration>
```

Das unten stehende C#-Programm zeigt, wie Sie den ElastiCache Cluster-Client verwenden, um eine Verbindung zu einem Clusterkonfigurationsendpunkt herzustellen und dem Cache ein Datenelement hinzuzufügen. Mithilfe von Auto Discovery stellt das Programm eine Verbindung mit allen Knoten im Cluster her, ohne dass ein Eingreifen erforderlich ist.

```
// *****
// Sample C# code to show how to integrate with the Amazon ElastiCache Auto Discovery
// feature.

using System;

using Amazon.ElastiCacheCluster;
```

```
using Enyim.Caching;
using Enyim.Caching.Memcached;

public class DotNetAutoDiscoveryDemo {

    public static void Main(String[] args) {

        // instantiate a new client.
        ElastiCacheClusterConfig config = new ElastiCacheClusterConfig();
        MemcachedClient memClient = new MemcachedClient(config);

        // Store the data for 3600 seconds (1hour) in the cluster.
        // The client will decide which cache host will store this item.
        memClient.Store(StoreMode.Set, 3600, "This is the data value.");

    } // end Main

} // end class DotNetAutoDiscoverDemo
```

Manuelles Herstellen einer Verbindung zu Memcache-Cache-Knoten

Wenn Ihr Client-Programm Auto Discovery nicht verwendet, kann es manuell eine Verbindung zu den einzelnen Memcache-Cache-Knoten herstellen. Dies ist das Standardverhalten für Memcached-Clients.

Eine Liste der Hostnamen und Portnummern der Cache-Knoten erhalten Sie über die [AWS Managementkonsole](#). Sie können den AWS CLIaws `elasticache describe-cache-clusters` Befehl auch mit dem `--show-cache-node-info` Parameter verwenden.

Example

Der folgende Java-Codeausschnitt zeigt, wie eine Verbindung zu allen Knoten in einem Cluster mit vier Knoten hergestellt wird:

```
...

ArrayList<String> cacheNodes = new ArrayList<String>(
    Arrays.asList(
        "mycachecluster.fnjyzo.0001.use1.cache.amazonaws.com:11211",
        "mycachecluster.fnjyzo.0002.use1.cache.amazonaws.com:11211",
        "mycachecluster.fnjyzo.0003.use1.cache.amazonaws.com:11211",
        "mycachecluster.fnjyzo.0004.use1.cache.amazonaws.com:11211"));

MemcachedClient cache = new MemcachedClient(AddrUtil.getAddresses(cacheNodes));

...
```

Important

Wenn Sie Ihren Cluster durch Hinzufügen oder Entfernen von Knoten nach oben oder unten skalieren, müssen Sie die Liste der Knoten im Client-Code aktualisieren.

Hinzufügen von Auto Discovery zu Ihrer Memcached-Clientbibliothek

Die Konfigurationsinformationen für Auto Discovery werden redundant in jedem Memcached-Clusterknoten gespeichert. Client-Anwendungen können jeden Cache-Knoten abfragen und die Konfigurationsinformationen für alle Knoten im Cluster abrufen.

Die jeweilige Methode, die eine Anwendung dazu nutzt, hängt von der Cache-Engine-Version ab:

- Wenn die Cache-Engine-Version 1.4.14 oder höher ist, verwenden Sie den Befehl `config`.
- Wenn die Cache-Engine-Version älter als 1.4.14 ist, verwenden Sie den Befehl `get AmazonElastiCache:cluster`.

Die Ausgaben dieser beiden Befehle sind identisch und werden im Abschnitt [Ausgabeformat](#) beschrieben.

Cache-Engine-Version 1.4.14 oder höher

Für die Cache-Engine-Version 1.4.14 oder höher verwenden Sie den Befehl `config`. Dieser Befehl wurde den Memcached ASCII- und Binärprotokollen von hinzugefügt und ist im Cluster ElastiCache Client implementiert. ElastiCache Wenn Sie Auto Discovery mit einer anderen Client-Bibliothek verwenden möchten, muss diese Bibliothek um den Support für den Befehl `config` erweitert werden.

Note

Die folgende Dokumentation bezieht sich zwar auf das ASCII-Protokoll, der Befehl `config` unterstützt jedoch sowohl das ASCII- als auch das Binärprotokoll. Wenn Sie Auto Discovery-Unterstützung mithilfe des Binärprotokolls hinzufügen möchten, lesen Sie im [Quellcode für den Cluster-Client](#) nach. ElastiCache

Syntax

```
config [sub-command] [key]
```

Optionen

Name	Description	Erforderlich
sub-command		Ja

Name	Description	Erforderlich
	Der Unterbefehl, der zum Interagieren mit einem Cache-Knoten verwendet wird. Für Auto Discovery lautet dieser Unterbefehl <code>get</code> .	
key	Der Schlüssel, mit dem die Cluster-Konfiguration gespeichert ist. Für Auto Discovery lautet dieser Schlüssel <code>cluster</code> .	Ja

Verwenden Sie zum Abrufen der Cluster-Konfigurationsinformationen den folgenden Befehl:

```
config get cluster
```

Cache-Engine-Version 1.4.14 oder niedriger

Verwenden Sie zum Abrufen der Cluster-Konfigurationsinformationen den folgenden Befehl:

```
get AmazonElastiCache:cluster
```

Note

Verändern Sie nicht den Schlüssel „:clusterAmazonElastiCache“, da sich dort die Cluster-Konfigurationsinformationen befinden. Wenn Sie diesen Schlüssel überschreiben, ist der Client möglicherweise für einen kurzen Zeitraum (nicht länger als 15 Sekunden) falsch konfiguriert, bevor die Konfigurationsinformationen ElastiCache automatisch und korrekt aktualisiert werden.

Ausgabeformat

Wenn Sie `config get cluster` oder `get AmazonElastiCache:cluster` verwenden, umfasst die Antwort zwei Zeilen:

- Die Versionsnummer der Konfigurationsinformationen. Jedes Mal, wenn ein Knoten dem Cluster hinzugefügt oder daraus entfernt wird, erhöht sich die Versionsnummer um eins.
- Eine Liste der Cache-Knoten. Jeder Knoten in der Liste wird durch eine `hostname|ip-address|port-`Gruppe repräsentiert und durch ein Leerzeichen getrennt.

Am Ende jeder Zeile erscheint ein Wagenrücklauf- und Zeilenvorschubzeichen (CR + LF). Die Datenzeile enthält am Ende ein Zeilenvorschubzeichen (LF), dem die Zeichen CR + LF hinzugefügt werden. Die Config-Versionszeile wird durch LF ohne CR beendet.

Ein Cluster mit drei Knoten würde wie folgt dargestellt:

```
configversion\n
hostname|ip-address|port hostname|ip-address|port hostname|ip-address|port\n\r\n
```

Jeder Knoten wird sowohl mit dem CNAME als auch der privaten IP-Adresse angezeigt. Der CNAME ist immer vorhanden. Wenn die private IP-Adresse nicht verfügbar ist, wird sie nicht angezeigt. Die Pipe-Zeichen "|" werden dennoch gedruckt.

Example

Das nachfolgende Beispiel zeigt die Nutzlast, die zurückgegeben wird, wenn Sie die Konfigurationsinformationen abfragen:

```
CONFIG cluster 0 136\r\n
12\n
myCluster.pc4ldq.0001.use1.cache.amazonaws.com|10.82.235.120|11211
  myCluster.pc4ldq.0002.use1.cache.amazonaws.com|10.80.249.27|11211\n\r\n
END\r\n
```

Note

- Die zweite Zeile gibt an, dass die Konfigurationsinformationen bisher zwölfmal geändert wurden.
- In der dritten Zeile wird die Liste der Knoten in alphabetischer Reihenfolge nach Hostname angegeben. Diese Reihenfolge kann sich von der Sequenz unterscheiden, die Sie derzeit in Ihrer Client-Anwendung verwenden.

ElastiCache Kunden mit auto Erkennung

Cluster-Client-Programme können automatisch alle Clusterknoten identifizieren und eine Verbindung zu ihnen herstellen, auf denen die Memcached-Engine ausgeführt wird.

In diesem Abschnitt wird die Installation und Konfiguration der ElastiCache PHP- und .NET-Clients für die Verwendung mit Auto Discovery beschrieben.

Themen

- [Installieren und Kompilieren von Cluster-Clients](#)
- [Konfiguration von ElastiCache Clients](#)

Installieren und Kompilieren von Cluster-Clients

Dieser Abschnitt behandelt die Installation, Konfiguration und Kompilierung der PHP- und .NET Amazon ElastiCache Auto Discovery-Cluster-Clients.

Themen

- [Installation des ElastiCache Clusterclients für.NET](#)
- [Installation des ElastiCache Cluster-Clients für PHP](#)
- [Den Quellcode für den ElastiCache Cluster-Client für PHP kompilieren](#)

Installation des ElastiCache Clusterclients für.NET

Den Code für den ElastiCache .NET Cluster Client als Open Source finden Sie unter <https://github.com/awslabs/elasticache-cluster-config-net>.

In diesem Abschnitt wird beschrieben, wie Sie die .NET-Komponenten für den ElastiCache Cluster Client auf EC2 Amazon-Instances installieren, aktualisieren und entfernen. Weitere Informationen zu Auto Discovery finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#). Beispiele für .NET-Code zur Verwendung des Clients finden Sie unter [Den ElastiCache Cluster-Client für.NET verwenden](#).

Themen

- [Installieren von .NET](#)
- [Laden Sie den ElastiCache .NET-Clusterclient herunter für ElastiCache](#)
- [Installieren Sie Assemblys AWS mit NuGet](#)

Installieren von .NET

Sie müssen .NET 3.5 oder höher installiert haben, um AWS das .NET-SDK für verwenden zu können ElastiCache. Wenn .NET 3.5 oder höher nicht installiert ist, können Sie die aktuelle Version unter <http://www.microsoft.com/net> herunterladen, um sie zu installieren.

Laden Sie den ElastiCache .NET-Clusterclient herunter für ElastiCache

Um ElastiCache den .NET-Clusterclient herunterzuladen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie im Navigationsbereich auf ElastiCache Cluster Client.
3. Wählen Sie in der Liste ElastiCache Memcached Cluster Client herunterladen die Option .NET aus, und klicken Sie dann auf Herunterladen.

Installieren Sie Assemblys AWS mit NuGet

NuGet ist ein Paketverwaltungssystem für die .NET-Plattform. NuGet kennt die Abhängigkeiten von Assemblys und installiert alle erforderlichen Dateien automatisch. NuGet installierte Assemblys werden zusammen mit Ihrer Lösung gespeichert und nicht etwa an einem zentralen Ort `Program Files`, sodass Sie anwendungsspezifische Versionen installieren können, ohne dass Kompatibilitätsprobleme auftreten.

Wird installiert NuGet

NuGet kann über die Installationsgalerie auf MSDN installiert werden; siehe <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c>. Wenn Sie Visual Studio 2010 NuGet oder höher verwenden, wird es automatisch installiert.

Sie können entweder NuGet den Solution Explorer oder die Package Manager Console verwenden.

NuGet Vom Solution Explorer aus verwenden

Zur Verwendung NuGet aus dem Solution Explorer in Visual Studio 2010

1. Wählen Sie im Menü Extras den Bibliotheks-Paket-Manager aus.
2. Klicken Sie auf Paket-Manager-Konsole.

Zur Verwendung NuGet aus dem Solution Explorer in Visual Studio 2012 oder Visual Studio 2013

1. Wählen Sie im Menü Tools die Option NuGet Package Manager aus.
2. Klicken Sie auf Paket-Manager-Konsole.

An der Befehlszeile können Sie die Komponenten mit Install-Package wie im Folgenden dargestellt installieren.

```
Install-Package Amazon.ElastiCacheCluster
```

Eine Seite für jedes Paket, das verfügbar ist NuGet, z. B. für die AWS SDK- und AWS.Extensions-Assemblies, finden Sie auf der NuGet Website unter <http://www.nuget.org>. Die Seite für jedes Paket enthält eine Beispielbefehlszeile für die Installation des Pakets mithilfe der Konsole und eine Liste der früheren Versionen des Pakets, die über NuGet verfügbar sind.

Weitere Informationen zu den Befehlen der Paket-Manager-Konsole finden Sie unter <http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29>.

Installation des ElastiCache Cluster-Clients für PHP

In diesem Abschnitt wird beschrieben, wie Sie die PHP-Komponenten für den ElastiCache Cluster-Client auf EC2 Amazon-Instances installieren, aktualisieren und entfernen. Weitere Informationen zu Auto Discovery finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#). Beispiele für PHP-Code zur Verwendung des Clients finden Sie unter [Den ElastiCache Cluster-Client für PHP verwenden](#).

Themen

- [Herunterladen des Installationspakets](#)
- [Für Benutzer, die bereits die Erweiterung php-memcached installiert haben](#)
- [Installationsschritte für neue Benutzer](#)
- [Entfernen des PHP-Cluster-Clients](#)

Herunterladen des Installationspakets

Um sicherzustellen, dass Sie die richtige Version des ElastiCache Cluster Client für PHP verwenden, müssen Sie wissen, welche Version von PHP auf Ihrer EC2 Amazon-Instance installiert ist. Sie müssen auch wissen, ob auf Ihrer EC2 Amazon-Instance eine 64-Bit- oder 32-Bit-Version von Linux ausgeführt wird.

So ermitteln Sie die auf Ihrer EC2 Amazon-Instance installierte PHP-Version

- Führen Sie in der Befehlszeile den folgenden Befehl aus:

```
php -v
```

Die PHP-Version wird wie in diesem Beispiel ausgegeben:

```
PHP 5.4.10 (cli) (built: Jan 11 2013 14:48:57)
Copyright (c) 1997-2012 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies
```

Note

Wenn Ihre PHP- und Memcached-Versionen inkompatibel sind, wird in etwa folgende Fehlermeldung ausgegeben:

```
PHP Warning: PHP Startup: memcached: Unable to initialize module
Module compiled with module API=20100525
PHP compiled with module API=20131226
These options need to match
in Unknown on line 0
```

In diesem Fall müssen Sie das Modul aus dem Quellcode kompilieren. Weitere Informationen finden Sie unter [Den Quellcode für den ElastiCache Cluster-Client für PHP kompilieren](#).

So ermitteln Sie Ihre Amazon EC2 AMI-Architektur (64-Bit oder 32-Bit)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie in der Instance-Liste auf Ihre EC2 Amazon-Instance.
3. Suchen Sie auf der Registerkarte Description nach dem Feld AMI:. Die Beschreibung einer 64-Bit-Instance enthält x86_64, bei einer 32-Bit-Instance enthält dieses Feld i386 oder i686.

Sie sind jetzt bereit, den ElastiCache Cluster Client herunterzuladen.

Um den ElastiCache Cluster-Client für PHP herunterzuladen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie in der ElastiCache Konsole ElastiCache Cluster Client aus.
3. Wählen Sie aus der Liste ElastiCache Memcached Cluster Client herunterladen den ElastiCache Cluster-Client aus, der Ihrer PHP-Version und AMI-Architektur entspricht, und klicken Sie dann auf die Schaltfläche Herunterladen.

Für Benutzer, die bereits die Erweiterung php-memcached installiert haben

So aktualisieren Sie die **php-memcached**-Installation

1. Entfernen Sie die bisherige Installation der Memcached-Erweiterung für PHP, wie im Thema [Entfernen des PHP-Cluster-Clients](#) beschrieben.

2. Installieren Sie die neue ElastiCache php-memcached Erweiterung wie zuvor unter beschrieben [Installationsschritte für neue Benutzer](#).

Installationsschritte für neue Benutzer

Themen

- [Installieren von PHP 7.x für neue Benutzer](#)
- [Installieren von PHP 5.x für neue Benutzer](#)

Installieren von PHP 7.x für neue Benutzer

Themen

- [Um PHP 7 auf einem Ubuntu-Server 14.04 LTS AMI \(64-Bit und 32-Bit\) zu installieren](#)
- [So installieren Sie PHP 7 auf einem Amazon Linux 201609 AMI](#)
- [So installieren Sie PHP 7 auf einem SUSE Linux-AMI](#)

Um PHP 7 auf einem Ubuntu-Server 14.04 LTS AMI (64-Bit und 32-Bit) zu installieren

1. Starten Sie eine neue Instance aus dem AMI.
2. Führen Sie die folgenden Befehle aus:

```
sudo apt-get update
sudo apt-get install gcc g++
```

3. Installieren Sie PHP 7.

```
sudo yum install php70
```

4. Laden Sie den Amazon ElastiCache Cluster Client herunter.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. Extrahieren Sie latest-64bit.

```
tar -zxvf latest-64bit
```


6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei `amazon-elasticache-cluster-client.so` nach `/usr/lib/php/20151012`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib/php/20151012
```

7. Fügen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` in die Datei `/etc/php/7.0/cli/php.ini` ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php/7.0/cli/php.ini
```

8. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

So installieren Sie PHP 7 auf einem Amazon Linux 201609 AMI

1. Starten Sie eine neue Instance aus dem AMI.
2. Führen Sie den folgenden Befehl aus:

```
sudo yum install gcc-c++
```

3. Installieren Sie PHP 7.

```
sudo yum install php70
```

4. Laden Sie den Amazon ElastiCache Cluster Client herunter.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. Extrahieren Sie `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei `amazon-elasticache-cluster-client.so` nach `/usr/lib64/php/7.0/modules/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.0/modules/
```

7. Erstellen Sie die Datei `50-memcached.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php-7.0.d/50-memcached.ini
```

8. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

So installieren Sie PHP 7 auf einem SUSE Linux-AMI

1. Starten Sie eine neue Instance aus dem AMI.
2. Führen Sie den folgenden Befehl aus:

```
sudo zypper install gcc
```

3. Installieren Sie PHP 7.

```
sudo yum install php70
```

4. Laden Sie den Amazon ElastiCache Cluster Client herunter.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. Extrahieren Sie `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei `amazon-elasticache-cluster-client.so` nach `/usr/lib64/php7/extensions/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php7/extensions/
```

7. Fügen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` in die Datei `/etc/php7/cli/php.ini` ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php7/cli/php.ini
```

8. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Installieren von PHP 5.x für neue Benutzer

Themen

- [So installieren Sie PHP 5 auf einem Amazon Linux 2014.03-AMI \(64-Bit und 32-Bit\)](#)
- [So installieren Sie PHP 5 auf einem Red Hat Enterprise Linux 7.0-AMI \(64-Bit und 32-Bit\)](#)
- [So installieren Sie PHP 5 auf einem Ubuntu-Server-14.04-LTS-AMI \(64-Bit und 32-Bit\)](#)
- [So installieren Sie PHP 5 für SUSE-Linux-Enterprise-Server-11-AMI \(64-Bit oder 32-Bit\)](#)
- [Andere Linux-Distributionen](#)

So installieren Sie PHP 5 auf einem Amazon Linux 2014.03-AMI (64-Bit und 32-Bit)

1. Starten Sie eine Amazon Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. PHP-Abhängigkeiten installieren:

```
sudo yum install gcc-c++ php php-pear
```

3. Laden Sie das richtige php-memcached Paket für Ihre EC2 Amazon-Instance und PHP-Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren php-memcached. Die URI ist der Download-Pfad für das Installationspaket:

```
sudo pecl install <package download path>
```

Hier ist ein Beispielininstallationsbefehl für PHP 5.4, 64-Bit-Linux. Ersetzen Sie es in diesem Beispiel **X.Y.Z** durch die tatsächliche Versionsnummer:

```
sudo pecl install /home/AmazonElastiCacheClusterClient-X.Y.Z-PHP54-64bit.tgz
```

 Note

Verwenden Sie immer die aktuelle Version des Installationsartefakts.

5. Fügen Sie mit entsprechender root/sudo Genehmigung eine neue Datei mit dem Namen `memcached.ini` im `/etc/php.d` Verzeichnis hinzu und fügen Sie „extension= amazon-elasticache-cluster-client .so“ in die Datei ein:

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php.d/memcached.ini
```

6. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

So installieren Sie PHP 5 auf einem Red Hat Enterprise Linux 7.0-AMI (64-Bit und 32-Bit)

1. Starten Sie eine Red Hat Enterprise Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. PHP-Abhängigkeiten installieren:

```
sudo yum install gcc-c++ php php-pear
```

3. Laden Sie das richtige php-memcached Paket für Ihre EC2 Amazon-Instance und PHP-Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren php-memcached. Die URI ist der Download-Pfad für das Installationspaket:

```
sudo pecl install <package download path>
```

5. Fügen Sie mit entsprechender root/sudo Genehmigung eine neue Datei mit dem Namen `memcached.ini` im `/etc/php.d` Verzeichnis hinzu und fügen Sie sie `extension=amazon-elasticache-cluster-client.so` in die Datei ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/  
php.d/memcached.ini
```

6. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

So installieren Sie PHP 5 auf einem Ubuntu-Server-14.04-LTS-AMI (64-Bit und 32-Bit)

1. Starten Sie eine Ubuntu Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. PHP-Abhängigkeiten installieren:

```
sudo apt-get update  
sudo apt-get install gcc g++ php5 php-pear
```

3. Laden Sie das richtige php-memcached Paket für Ihre EC2 Amazon-Instance und PHP-Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren php-memcached. Die URI ist der Download-Pfad für das Installationspaket.

```
sudo pecl install <package download path>
```

Note

In diesem Installationsschritt wird das Build-Artefakt `amazon-elasticache-cluster-client.so` im Verzeichnis `/usr/lib/php5/20121212*` installiert. Überprüfen Sie den absoluten Pfad des Build-Artefakts, da er im nächsten Schritt benötigt wird.

Wenn der vorherige Befehl fehlschlägt, müssen Sie das PHP-Client-Artefakt `amazon-elasticache-cluster-client.so` manuell aus der heruntergeladenen Datei `*.tgz` extrahieren und in das Verzeichnis `/usr/lib/php5/20121212*` kopieren.

```
tar -xvf <package download path>  
cp amazon-elasticache-cluster-client.so /usr/lib/php5/20121212/
```

5. Fügen Sie mit entsprechender root/sudo Genehmigung eine neue Datei mit dem Namen `memcached.ini` im `/etc/php5/cli/conf.d` Verzeichnis hinzu und fügen Sie „`extension=<absolute path to amazon-elasticache-cluster-client.so>`“ in die Datei ein.

```
echo "extension=<absolute path to amazon-elasticache-cluster-client.so>" | sudo tee --append /etc/php5/cli/conf.d/memcached.ini
```

6. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

So installieren Sie PHP 5 für SUSE-Linux-Enterprise-Server-11-AMI (64-Bit oder 32-Bit)

1. Starten Sie eine SUSE-Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. PHP-Abhängigkeiten installieren:

```
sudo zypper install gcc php53-devel
```

3. Laden Sie das richtige php-memcached Paket für Ihre EC2 Amazon-Instance und PHP-Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren php-memcached. Die URI ist der Download-Pfad für das Installationspaket.

```
sudo pecl install <package download path>
```

5. Fügen Sie mit entsprechender root/sudo Genehmigung eine neue Datei mit dem Namen `memcached.ini` im `/etc/php5/conf.d` Verzeichnis hinzu und fügen Sie sie **`extension=amazon-elasticache-cluster-client.so`** in die Datei ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php5/conf.d/memcached.ini
```

6. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

 Note

Sollte Schritt 5 für die vorherigen Plattformen nicht funktionieren, überprüfen Sie den Installationspfad für `amazon-elasticache-cluster-client.so`. Geben Sie auch den vollständigen Pfad der Binary in der Erweiterung an. Bei der verwendeten PHP-Version muss es sich um eine unterstützte Version handeln. Die Versionen 5.3 bis 5.5 werden unterstützt.

Andere Linux-Distributionen

Auf einigen Systemen, insbesondere CentOS 7 und Red Hat Enterprise Linux (RHEL) 7.1, wurde `libsasl2.so.3` durch `libsasl2.so.2` ersetzt. Wenn Sie auf diesen Systemen den ElastiCache Cluster-Client laden, versucht er, ihn zu finden und zu laden `libsasl2.so.2`, aber er schlägt fehl. Um dieses Problem zu beheben, erstellen Sie einen symbolischen Link auf `libsasl2.so.3`, damit der Client beim Laden von `libsasl2.so.2` auf `libsasl2.so.3` weitergeleitet wird. Mit dem folgenden Code wird dieser symbolische Link erstellt.

```
cd /usr/lib64
sudo ln libsasl2.so.3 libsasl2.so.2
```

Entfernen des PHP-Cluster-Clients

Themen

- [Entfernen einer früheren Version von PHP 7](#)
- [Entfernen einer früheren Version von PHP 5](#)

Entfernen einer früheren Version von PHP 7

So entfernen Sie eine frühere Version von PHP 7

1. Entfernen Sie die Datei `amazon-elasticache-cluster-client.so` aus dem entsprechenden PHP-Bibliotheksverzeichnis, wie zuvor in den Installationsanleitungen erläutert. Weitere Informationen finden Sie in Installationsabschnitt unter [Für Benutzer, die bereits die Erweiterung php-memcached installiert haben](#).
2. Entfernen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` aus der Datei `php.ini`.

3. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Entfernen einer früheren Version von PHP 5

So entfernen Sie eine frühere Version von PHP 5

1. Entfernen Sie die Erweiterung php-memcached:

```
sudo pecl uninstall __uri/AmazonElastiCacheClusterClient
```

2. Entfernen Sie die Datei `memcached.ini` aus dem entsprechenden Verzeichnis, wie zuvor in den Installationsanleitungen erläutert.

Den Quellcode für den ElastiCache Cluster-Client für PHP kompilieren

In diesem Abschnitt wird beschrieben, wie Sie den Quellcode für den ElastiCache Cluster-Client für PHP abrufen und kompilieren.

[Es gibt zwei Pakete, aus denen Sie Daten abrufen GitHub und kompilieren müssen: aws-elasticache-cluster-client-libmemcached und -. aws-elasticache-cluster-client memcached-for-php](#)

Themen

- [Kompilieren der libmemcached-Bibliothek](#)
- [Den ElastiCache Memcached Auto Discovery Client für PHP kompilieren](#)

Kompilieren der libmemcached-Bibliothek

Um die Bibliothek -libmemcached zu kompilieren aws-elasticache-cluster-client

1. Starten Sie eine EC2 Amazon-Instance.
2. Installieren Sie die Abhängigkeiten der Bibliothek.

- Auf dem Amazon Linux 201509 AMI

```
sudo yum install gcc gcc-c++ autoconf libevent-devel
```

- Auf dem Ubuntu 14.04 AMI

```
sudo apt-get update
sudo apt-get install libevent-dev gcc g++ make autoconf libsasl2-dev
```

3. Führen Sie einen Pull des Repositorys aus und kompilieren Sie den Code.

Download and install <https://github.com/aws-labs/aws-elasticache-cluster-client-libmemcached/archive/v1.0.18.tar.gz>

Den ElastiCache Memcached Auto Discovery Client für PHP kompilieren

In den folgenden Abschnitten wird beschrieben, wie der ElastiCache Memcached Auto Discovery Client kompiliert wird

Themen

- [Kompilieren des ElastiCache Memcached-Clients für PHP 7](#)
- [Kompilieren des ElastiCache Memcached-Clients für PHP 5](#)

Kompilieren des ElastiCache Memcached-Clients für PHP 7

Führen Sie die folgenden Befehle im Code-Verzeichnis aus.

```
git clone https://github.com/awslabs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
git checkout php7
sudo yum install php70-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory> --disable-memcached-sasl
make
make install
```

Note

Sie können die libmemcached-Bibliothek mit der PHP-Binärdatei statisch verknüpfen, sodass sie über verschiedene Plattformen portiert werden kann. Führen Sie dazu den folgenden Befehl vor make aus:

```
sed -i "s#-lmemcached#<libmemcached-install-directory>/lib/libmemcached.a -lcrypt -lpthread -lm -lstdc++ -lsasl2#" Makefile
```

Kompilieren des ElastiCache Memcached-Clients für PHP 5

Kompilieren Sie den aws-elasticache-cluster-client-memcached-for-php, indem Sie die folgenden Befehle im Ordner aws-elasticache-cluster-client-memcached-for-php/ ausführen.

```
git clone https://github.com/awslabs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
sudo yum install zlib-devel
phpize
```

```
./configure --with-libmemcached-dir=<libmemcached-install-directory>  
make  
make install
```

Konfiguration von ElastiCache Clients

Ein ElastiCache Cluster ist protokollkonform mit Valkey, Memcached und Redis OSS. Der Code, die Anwendungen und die beliebtesten Tools, die Sie heute in Ihrer bestehenden Umgebung verwenden, funktionieren problemlos mit dem Service.

Dieser Abschnitt befasst sich mit den spezifischen Überlegungen für das Verbinden mit Cache-Knoten in ElastiCache.

Themen

- [Eingeschränkte Befehle](#)
- [Finden von Knotenendpunkten und Portnummern](#)
- [Verbinden für die Verwendung der automatischen Erkennung](#)
- [Verbindung zu Knoten in einem Valkey- oder Redis OSS-Cluster herstellen](#)
- [DNS-Namen und zugrunde liegende IP](#)

Eingeschränkte Befehle

ElastiCache schränkt den Zugriff auf bestimmte Cache-Engine-spezifische Befehle ein, die erweiterte Rechte erfordern, um ein Managed Service-Erlebnis zu bieten. Für Cluster, auf denen Redis OSS ausgeführt wird, sind die folgenden Befehle nicht verfügbar:

- `bgrewriteaof`
- `bgsave`
- `config`
- `debug`
- `migrate`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

Finden von Knotenendpunkten und Portnummern

Um eine Verbindung zu einem Cache-Knoten herzustellen, muss Ihre Anwendung den Endpunkt und die Portnummer für diesen Knoten kennen.

Finden von Knotenendpunkten und Portnummern (Konsole)

So bestimmen Sie Knotenendpunkte und Portnummern

1. Melden Sie sich bei der [Amazon ElastiCache Management Console](#) an und wählen Sie die Engine aus, die auf Ihrem Cluster läuft.

Es wird eine Liste aller Cluster der ausgewählten Engine angezeigt.

2. Fahren Sie mit der Anleitung für die ausgewählte Engine und Konfiguration fort.
3. Wählen Sie den Namen des gewünschten Clusters aus.
4. Entnehmen Sie die benötigten Informationen für den gewünschten Knoten aus den Spalten Port und Endpoint.

Finden von Cache-Knotenendpunkten und Portnummern (AWS CLI)

Verwenden Sie den Befehl `describe-cache-clusters` mit dem Parameter `--show-cache-node-info`, um die Endpunkte und Portnummern von Cache-Knoten zu bestimmen.

```
aws elasticache describe-cache-clusters --show-cache-node-info
```

Sie finden die vollständig qualifizierten DNS-Namen und Portnummern im Abschnitt Endpunkt in der Ausgabe.

Suche nach Endpunkten und Portnummern von Cache-Knoten (ElastiCache API)

Verwenden Sie die Aktion `DescribeCacheClusters` mit dem Parameter `ShowCacheNodeInfo=true`, um die Endpunkte und Portnummern von Cache-Knoten zu bestimmen.

Example

```
https://elasticache.us-west-2.amazonaws.com /
```

```
?Action=DescribeCacheClusters
&ShowCacheNodeInfo=true
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20140421T220302Z
&Version=2014-09-30
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20140421T220302Z
&X-Amz-Expires=20140421T220302Z
&X-Amz-Signature=<signature>
&X-Amz-SignedHeaders=Host
```

Verbinden für die Verwendung der automatischen Erkennung

Wenn Sie in Ihren Anwendungen Auto Discovery verwenden, müssen Sie anstelle der einzelnen Endpunkte für jeden Cache-Knoten den Konfigurationsendpunkt für das Cluster kennen. Weitere Informationen finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#).

Note

Derzeit ist Auto Discovery nur für Cluster verfügbar, auf denen Memcached ausgeführt wird.

Verbindung zu Knoten in einem Valkey- oder Redis OSS-Cluster herstellen

Note

Derzeit werden Cluster (API/CLI: Replikationsgruppen), die Replikation und Read Replicas unterstützen, nur für Cluster unterstützt, auf denen Valkey oder Redis OSS ausgeführt wird.

ElastiCache Stellt für Cluster Konsolen-, CLI- und API-Schnittstellen bereit, um Verbindungsinformationen für einzelne Knoten abzurufen.

Für reine Lesezugriffe können Anwendungen eine Verbindung zu beliebigen Knoten im Cluster herstellen. Für Schreibaktivitäten empfehlen wir jedoch, dass Ihre Anwendungen eine Verbindung zum primären Endpunkt (Valkey oder Redis OSS (Clustermodus deaktiviert)) oder Konfigurationsendpunkt (Valkey oder Redis OSS (Clustermodus aktiviert)) für den Cluster herstellen, anstatt eine direkte Verbindung zu einem Knoten herzustellen. Dadurch wird sichergestellt, dass Ihre

Anwendungen immer den korrekten Knoten finden, auch wenn Sie das Cluster neu konfigurieren und einem Lesereplikat die primäre Rolle zuweisen.

Verbinden mit Clustern in einer Replikationsgruppe (Konsole)

So bestimmen Sie Endpunkte und Portnummern

- Weitere Informationen finden Sie auch im Thema [Finden Sie die Endpunkte eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

Verbinden mit Clustern in einer Replikationsgruppe (AWS CLI)

So bestimmen Sie die Cache-Knoten-Endpunkte und Portnummern

Verwenden Sie den Befehl `describe-replication-groups` mit dem Namen Ihrer Replikationsgruppe:

```
aws elasticache describe-replication-groups redis2x2
```

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
      "Description": "2 shards, 2 nodes (1 + 1 replica)",
      "NodeGroups": [
        {
          "Status": "available",
          "Slots": "0-8191",
          "NodeGroupId": "0001",
          "NodeGroupMembers": [
            {
              "PreferredAvailabilityZone": "us-west-2c",
              "CacheNodeId": "0001",
              "CacheClusterId": "redis2x2-0001-001"
            },
            {
              "PreferredAvailabilityZone": "us-west-2a",
              "CacheNodeId": "0001",
              "CacheClusterId": "redis2x2-0001-002"
            }
          ]
        }
      ]
    }
  ]
}
```

```

        ]
      },
      {
        "Status": "available",
        "Slots": "8192-16383",
        "NodeGroupId": "0002",
        "NodeGroupMembers": [
          {
            "PreferredAvailabilityZone": "us-west-2b",
            "CacheNodeId": "0001",
            "CacheClusterId": "redis2x2-0002-001"
          },
          {
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheNodeId": "0001",
            "CacheClusterId": "redis2x2-0002-002"
          }
        ]
      }
    ],
    "ConfigurationEndpoint": {
      "Port": 6379,
      "Address": "redis2x2.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "redis2x2",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "13:00-14:00",
    "MemberClusters": [
      "redis2x2-0001-001",
      "redis2x2-0001-002",
      "redis2x2-0002-001",
      "redis2x2-0002-002"
    ],
    "CacheNodeType": "cache.m3.medium",
    "PendingModifiedValues": {}
  }
]
}

```

Verbindung zu Clustern in einer Replikationsgruppe (API) herstellen ElastiCache

So bestimmen Sie die Cache-Knoten-Endpunkte und Portnummern

Rufen Sie `DescribeReplicationGroups` mit folgendem Parameter auf:

`ReplicationGroupId` = der Name Ihrer Replikationsgruppe

Example

```
https://elasticache.us-west-2.amazonaws.com /  
?Action=DescribeCacheClusters  
&ReplicationGroupId=repgroup01  
&Version=2014-09-30  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20140421T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20140421T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20140421T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

DNS-Namen und zugrunde liegende IP

Clients verfügen über eine Serverliste mit den Adressen und Ports der Server, auf denen Cache-Daten abgelegt werden. Bei der Verwendung ElastiCache gibt die `DescribeCacheClusters` API (oder das `describe-cache-clusters` Befehlszeilenprogramm) einen vollqualifizierten DNS-Eintrag und eine Portnummer zurück, die für die Serverliste verwendet werden können.

Important

Es ist wichtig, dass Client-Anwendungen so konfiguriert werden, dass DNS-Namen von Cache-Knoten regelmäßig aufgelöst werden, wenn sie versuchen, eine Verbindung zu einem Cache-Knotenendpunkt herzustellen.

ElastiCache stellt sicher, dass der DNS-Name der Cache-Knoten derselbe bleibt, wenn Cache-Knoten im Falle eines Fehlers wiederhergestellt werden.

Persistente Cache-Knotenverbindungen werden von den meisten Client-Bibliotheken standardmäßig unterstützt. Wir empfehlen, beim Verwenden von ElastiCache persistente Cache-Knotenverbindungen zu nutzen. Clientseitige Zwischenspeicherung von DNS-Informationen kann

an mehreren Orten erfolgen, darunter Client-Bibliotheken, der Sprache der Laufzeitumgebung oder dem Betriebssystem des Clients. Prüfen Sie die Konfiguration Ihrer Anwendung auf allen Ebenen, um sicherzustellen, dass IP-Adressen für Ihre Cache-Knoten regelmäßig aufgelöst werden.

Datenzuweisung ElastiCache

ElastiCache Bei Valkey- oder Redis-OSS-Clustern, die eine Replikationsgruppe bilden und einen Knotentyp aus der R6GD-Familie verwenden, werden die Daten zwischen Arbeitsspeicher und lokalem SSD-Speicher (Solid State Drives) aufgeteilt. Data Tiering bietet eine neue Preis-Leistungs-Option für Valkey- oder Redis-OSS-Workloads, da neben der Speicherung von Daten im Arbeitsspeicher auch kostengünstigere Solid-State-Laufwerke (SSDs) in jedem Clusterknoten verwendet werden. Diese Option ist ideal für Workloads, die regelmäßig auf bis zu 20 Prozent ihres gesamten Datensatzes zugreifen, und für Anwendungen, die beim Zugriff auf Daten auf SSD zusätzliche Latenz tolerieren können.

ElastiCache Überwacht auf ElastiCache Clustern mit Daten-Tiering die letzte Zugriffszeit jedes gespeicherten Elements. Wenn der verfügbare Speicher (DRAM) vollständig verbraucht ist, ElastiCache verwendet es einen LRU-Algorithmus (Least-Recently Used), um Objekte, auf die selten zugegriffen wird, automatisch vom Arbeitsspeicher auf die SSD zu verschieben. Wenn anschließend auf Daten auf der SSD zugegriffen wird, werden sie ElastiCache automatisch und asynchron zurück in den Arbeitsspeicher verschoben, bevor die Anforderung verarbeitet wird. Wenn Sie eine Workload haben, die regelmäßig nur auf eine Teilmenge ihrer Daten zugreift, ist Daten-Tiering eine optimale Möglichkeit, Ihre Kapazität kostengünstig zu skalieren.

Beachten Sie, dass bei der Verwendung von Daten-Tiering die Schlüssel selbst immer im Speicher verbleiben, während die LRU die Platzierung von Werten in dem Speicher im Vergleich zur Festplatte regelt. Im Allgemeinen empfehlen wir, dass Ihre Schlüsselgrößen kleiner als Ihre Wertgrößen sind, wenn Sie Daten-Tiering verwenden.

Das Daten-Tiering ist so konzipiert, dass es minimale Auswirkungen auf die Leistung von Anwendungs-Workloads hat. Bei Annahme von 500-Byte-String-Werten können Sie beispielsweise durchschnittlich zusätzliche 300 Mikrosekunden Latenzzeit für Anforderungen an auf SSD gespeicherte Daten im Vergleich zu Anforderungen an Daten im Speicher erwarten.

Mit der größten Daten-Tiering-Knotengröße (cache.r6gd.16xlarge) können Sie bis zu 1 Petabyte Daten in einem einzigen Cluster mit 500 Knoten speichern (500 TB bei Verwendung von 1 Lesereplikat). Data Tiering ist mit allen Valkey- oder Redis-OSS-Befehlen und Datenstrukturen kompatibel, die in unterstützt werden. ElastiCache Um diese Funktion nutzen zu können, sind keine clientseitigen Änderungen erforderlich.

Themen

- [Bewährte Methoden](#)
- [Einschränkungen](#)
- [Preisgestaltung](#)
- [Überwachen](#)
- [Verwenden von Daten-Tiering](#)
- [Wiederherstellen von Daten aus einem Backup in Cluster mit aktiviertem Daten-Tiering](#)

Bewährte Methoden

Wir empfehlen Ihnen, die folgenden bewährten Methoden:

- Daten-Tiering ist ideal für Workloads, die regelmäßig auf bis zu 20 Prozent ihres gesamten Datensatzes zugreifen, und für Anwendungen, die beim Zugriff auf Daten auf SSD zusätzliche Latenz tolerieren können.
- Bei Verwendung von SSD-Kapazität, die auf Daten-Tiering-Knoten verfügbar ist, empfehlen wir, dass die Wertgröße größer als die Schlüsselgröße ist. Wenn Elemente zwischen DRAM und SSD verschoben werden, bleiben die Schlüssel immer im Speicher und nur die Werte werden in die SSD-Ebene verschoben.

Einschränkungen

Für Daten-Tiering gelten die folgenden Beschränkungen:

- Sie können Daten-Tiering nur für Cluster verwenden, die Teil einer Replikationsgruppe sind.
- Der verwendete Knotentyp muss aus der r6gd-Familie stammen, die in den folgenden Regionen verfügbar ist: us-east-2, us-east-1, us-west-2, us-west-1, eu-west-1, eu-central-1, eu-north-1, eu-west-3, ap-northeast-1, ap-southeast-1, ap-southeast-2, ap-south-1, ca-central-1 und sa-east-1.
- Sie müssen eine Engine verwenden, die Valkey 7.2 oder höher ist, oder eine Redis OSS 6.2 oder höher.
- Sie können ein Backup eines R6gd-Clusters nur dann in einem anderen Cluster wiederherstellen, wenn dieses ebenfalls R6gd verwendet.
- Backups für Cluster mit Daten-Tiering können nicht nach Amazon S3 exportiert werden.
- Die Online-Migration wird für Cluster mit R6gd-Knotentyp nicht unterstützt.

- Die Skalierung von einem Cluster mit Daten-Tiering (z. B. ein Cluster, der einen R6gd-Knotentyp verwendet) zu einem Cluster ohne Daten-Tiering (z. B. ein Cluster, der einen R6g-Knotentyp verwendet) wird nicht unterstützt. Weitere Informationen finden Sie unter [Skalierung ElastiCache](#).
- Auto Scaling wird auf Clustern unterstützt, die Data Tiering für Valkey Version 7.2 und höher sowie Redis OSS Version 7.0.7 und höher verwenden. Weitere Informationen finden Sie unter [Auto Scaling Valkey- und Redis OSS-Cluster](#).
- Daten-Tiering unterstützt nur die maxmemory-Richtlinien `volatile-lru`, `allkeys-lru`, `volatile-lfu`, `allkeys-lfu` und `noeviction`.
- Forkless Save wird für Valkey Version 7.2 und höher sowie Redis OSS Version 7.0.7 und höher unterstützt. Weitere Informationen finden Sie unter [So werden Synchronisation und Backup implementiert](#).
- Elemente, die größer als 128 MiB sind, werden nicht auf SSD verschoben.
- Ab Valley 8.1 und höher wird ein Element, dessen Schlüssel+Wertgröße weniger als 40 Byte beträgt, nicht auf die SSD verschoben.

Preisgestaltung

R6gd-Knoten haben eine 4,8-mal höhere Gesamtkapazität (Speicher + SSD). Bei maximaler Auslastung können Sie durch ihren Einsatz Einsparungen von über 60 Prozent im Vergleich zu R6g-Knoten (nur Speicher) erzielen. Weitere Informationen finden Sie unter [ElastiCache Preise](#).

Überwachen

ElastiCache bietet Metriken, die speziell für die Überwachung von Leistungsclustern entwickelt wurden, die Datenklassifizierung verwenden. Um das Verhältnis der Elemente in DRAM im Vergleich zu SSD zu überwachen, können Sie die `CurrlItems` Metrik unter [Metrics for Valkey und Redis OSS](#) verwenden. Sie können den Prozentsatz wie folgt berechnen: $(\text{CurrlItems mit Dimension: Tier} = \text{Speicher} * 100) / (\text{ohne CurrlItems Dimensionsfilter})$.

Wenn die konfigurierte Löschrichtlinie dies zulässt, ElastiCache wird mit dem Löschen von Elementen begonnen, wenn der Prozentsatz der Elemente im Speicher unter 5 Prozent sinkt. Auf Knoten, für die die Noeviction-Richtlinie konfiguriert wurde, wird bei Schreibvorgängen der Fehler „Nicht genügend Arbeitsspeicher“ angezeigt.

Es wird dennoch empfohlen, die Skalierung für Cluster mit aktiviertem Clustermodus oder die Hochskalierung für Cluster mit deaktiviertem Clustermodus in Betracht zu ziehen, wenn der Prozentsatz der Elemente im Arbeitsspeicher unter 5 Prozent sinkt. Weitere Informationen zur

Skalierung finden Sie unter [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#). Weitere Informationen zu Metriken für Valkey- oder Redis OSS-Cluster, die Data Tiering verwenden, finden Sie unter. [Metriken für Valkey und Redis OSS](#)

Verwenden von Daten-Tiering

Verwenden von Data Tiering mit dem AWS-Managementkonsole

Wenn Sie einen Cluster als Teil einer Replikationsgruppe erstellen, verwenden Sie Daten-Tiering, indem Sie einen Knotentyp aus der R6gd-Familie auswählen, z. B. cache.r6gd.xlarge. Bei Auswahl dieses Knotentyps wird das Daten-Tiering automatisch aktiviert.

Weitere Informationen zum Erstellen von Clustern finden Sie unter [Einen Cluster für Valkey oder Redis OSS erstellen](#).

Aktivieren von Data Tiering mit dem AWS CLI

Beim Erstellen einer Replikationsgruppe mithilfe von verwenden Sie Data Tiering AWS CLI, indem Sie einen Knotentyp aus der r6gd-Familie auswählen, z. B. cache.r6gd.xlarge, und den Parameter festlegen. `--data-tiering-enabled`

Sie können sich das Daten-Tiering nicht abwählen, wenn Sie einen Knotentyp aus der R6gd-Familie auswählen. Wenn Sie den Parameter `--no-data-tiering-enabled` festlegen, schlägt die Operation fehl.

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id redis-dt-cluster \  
  --replication-group-description "Redis OSS cluster with data tiering" \  
  --num-node-groups 1 \  
  --replicas-per-node-group 1 \  
  --cache-node-type cache.r6gd.xlarge \  
  --engine redis \  
  --cache-subnet-group-name default \  
  --automatic-failover-enabled \  
  --data-tiering-enabled
```

Für Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id redis-dt-cluster ^
```

```
--replication-group-description "Redis OSS cluster with data tiering" ^
--num-node-groups 1 ^
--replicas-per-node-group 1 ^
--cache-node-type cache.r6gd.xlarge ^
--engine redis ^
--cache-subnet-group-name default ^
--automatic-failover-enabled ^
--data-tiering-enabled
```

Nach dem Ausführen dieses Vorgangs sehen Sie eine Antwort ähnlich dem folgenden:

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "redis-dt-cluster",
    "Description": "Redis OSS cluster with data tiering",
    "Status": "creating",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "redis-dt-cluster"
    ],
    "AutomaticFailover": "enabled",
    "DataTiering": "enabled",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:00-07:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r6gd.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

Wiederherstellen von Daten aus einem Backup in Cluster mit aktiviertem Daten-Tiering

Sie können ein Backup auf einem neuen Cluster mit aktiviertem Data Tiering mithilfe der (Konsole), () oder (API) wiederherstellen. AWS CLI ElastiCache Wenn Sie einen Cluster mit Knotentypen in der R6gd-Familie erstellen, ist Daten-Tiering aktiviert.

Wiederherstellen von Daten aus einem Backup in Cluster mit aktiviertem Daten-Tiering (Konsole)

So stellen Sie ein Backup in einem neuen Cluster mit aktiviertem Daten-Tiering wieder her (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.

2. Wählen Sie im Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Sicherungen das Kontrollkästchen links neben dem Namen der Sicherung, aus der Sie wiederherstellen möchten.
4. Wählen Sie Restore (Wiederherstellen) aus.
5. Füllen Sie das Dialogfeld Restore Cluster aus. Füllen Sie unbedingt alle Pflichtfelder aus sowie alle anderen, deren Standardeinstellungen Sie ändern möchten.
 1. Cluster-ID – Erforderlich. Der Name des neuen Clusters.
 2. Cluster-Modus aktiviert (Scale Out) — Wählen Sie diese Option für einen Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert).
 3. Node Type (Knotentyp) – Geben Sie cache.r6gd.xlarge oder einen anderen Knotentyp aus der R6gd-Familie an.
 4. Anzahl der Shards – Wählen Sie die Anzahl der Shards aus, die sich im neuen Cluster (API/ CLI: Knotengruppen) befinden sollen.
 5. Replicas pro Shard – Wählen Sie die Anzahl der Lesereplikat-Knoten aus, die sich in jedem Shard befinden sollen.
 6. Slots und Keyspaces – Wählen Sie aus, wie die Schlüssel auf die Shards verteilt werden sollen. Wenn Sie die Schlüsselverteilungen angeben möchten, füllen Sie die Tabelle mit den Schlüsselbereichen für jeden Shard aus.
 7. Availability zone(s) – Geben Sie an, wie die Availability Zones des Clusters ausgewählt werden sollen.
 8. Port – Ändern Sie diese Einstellung nur, wenn der neue Cluster einen anderen Port verwenden soll.
 9. VPC auswählen – Wählen Sie die VPC aus, in der dieser Cluster erstellt werden soll.
 - 10 Parametergruppe — Wählen Sie eine Parametergruppe, die ausreichend Speicher für den Valkey- oder Redis-OSS-Overhead für den ausgewählten Knotentyp reserviert.
6. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Erstellen.

Weitere Informationen zum Erstellen von Clustern finden Sie unter [Einen Cluster für Valkey oder Redis OSS erstellen](#).

Wiederherstellen von Daten aus einem Backup in Cluster mit aktiviertem Daten-Tiering (AWS CLI)

Beim Erstellen einer Replikationsgruppe mithilfe von wird Data Tiering standardmäßig verwendet AWS CLI, indem ein Knotentyp aus der r6gd-Familie ausgewählt wird, z. B. `cache.r6gd.xlarge`, und der Parameter festgelegt wird. `--data-tiering-enabled`

Sie können sich das Daten-Tiering nicht abwählen, wenn Sie einen Knotentyp aus der R6gd-Familie auswählen. Wenn Sie den Parameter `--no-data-tiering-enabled` festlegen, schlägt die Operation fehl.

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id redis-dt-cluster \  
  --replication-group-description "Redis OSS cluster with data tiering" \  
  --num-node-groups 1 \  
  --replicas-per-node-group 1 \  
  --cache-node-type cache.r6gd.xlarge \  
  --engine redis \  
  --cache-subnet-group-name default \  
  --automatic-failover-enabled \  
  --data-tiering-enabled \  
  --snapshot-name my-snapshot
```

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group ^  
  --replication-group-id redis-dt-cluster ^  
  --replication-group-description "Redis OSS cluster with data tiering" ^  
  --num-node-groups 1 ^  
  --replicas-per-node-group 1 ^  
  --cache-node-type cache.r6gd.xlarge ^  
  --engine redis ^  
  --cache-subnet-group-name default ^  
  --automatic-failover-enabled ^  
  --data-tiering-enabled ^  
  --snapshot-name my-snapshot
```

Nach dem Ausführen dieses Vorgangs sehen Sie eine Antwort ähnlich dem folgenden:

```
{  
  "ReplicationGroup": {
```



```
{
  "ReplicationGroupId": "redis-dt-cluster",
  "Description": "Redis OSS cluster with data tiering",
  "Status": "creating",
  "PendingModifiedValues": {},
  "MemberClusters": [
    "redis-dt-cluster"
  ],
  "AutomaticFailover": "enabled",
  "DataTiering": "enabled",
  "SnapshotRetentionLimit": 0,
  "SnapshotWindow": "06:00-07:00",
  "ClusterEnabled": false,
  "CacheNodeType": "cache.r6gd.xlarge",
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false
}
```

Einen Cluster vorbereiten in ElastiCache

Im Folgenden finden Sie Anweisungen zum Erstellen eines Clusters mithilfe der ElastiCache Konsole, AWS CLI, oder der ElastiCache API.

Sie können einen ElastiCache Cluster auch mithilfe von erstellen [AWS CloudFormation](#). Weitere Informationen finden Sie unter [AWS:ElastiCache:: CacheCluster](#) im AWS Cloud Formation-Benutzerhandbuch, das Anleitungen zur Implementierung dieses Ansatzes enthält.

Wenn Sie einen Cluster oder eine Replikationsgruppe erstellen, ist es ratsam, einige Vorbereitungen zu treffen, damit Sie nicht sofort ein Upgrade durchführen oder Änderungen vornehmen müssen.

Themen

- [Ermitteln Ihrer ElastiCache Cluster-Anforderungen](#)
- [Auswahl der Knotengröße](#)

Ermitteln Ihrer ElastiCache Cluster-Anforderungen

Vorbereitung

Wenn Sie die Antworten auf die folgenden Fragen kennen, kann die ElastiCache Clustererstellung reibungsloser vonstatten gehen:

- Welchen Knoten-Instance-Typ benötigen Sie?

Eine Anleitung bei der Auswahl eines Knoten-Instance-Typs finden Sie unter [Auswahl der Knotengröße](#).

- Werden Sie Ihren Cluster in einer Virtual Private Cloud (VPC) auf der Basis von Amazon VPC starten?

 Important

Wenn Sie den Cluster in einer VPC starten, müssen Sie in derselben VPC eine Subnetzgruppe erstellen, bevor mit dem Erstellen eines Clusters beginnen. Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).

ElastiCache ist so konzipiert, dass der Zugriff AWS über Amazon von innen möglich ist EC2. Wenn Sie jedoch in einer VPC starten, die auf Amazon VPC basiert, und Ihr Cluster sich in einer VPC befindet, können Sie den Zugriff von außerhalb AWS ermöglichen. Weitere Informationen finden Sie unter [Zugriff auf ElastiCache Ressourcen von außen AWS](#).

- Müssen Sie irgendwelche Parameterwerte anpassen?

Erstellen Sie in diesem Fall eine benutzerdefinierte Parametergruppe. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

Wenn Sie Valkey oder Redis OSS verwenden, sollten Sie die Einstellung reserved-memory von oder in Betracht ziehen. reserved-memory-percent Weitere Informationen finden Sie unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

- Müssen Sie eine eigene VPC-Sicherheitsgruppe erstellen?

Weitere Informationen finden Sie unter [Sicherheit in Ihrer VPC](#).

- Haben Sie vor, Fehlertoleranz zu implementieren?

Weitere Informationen finden Sie unter [Minimieren von Ausfällen](#).

Themen

- [ElastiCache Speicher- und Prozessoranforderungen](#)
- [Konfiguration von Memcached-Cluster](#)
- [Valkey- und Redis OSS-Clusterkonfiguration](#)

- [ElastiCache Skalierungsanforderungen](#)
- [ElastiCache Zugriffsanforderungen](#)
- [Anforderungen an Region, Availability Zone und Lokale Zone für ElastiCache](#)

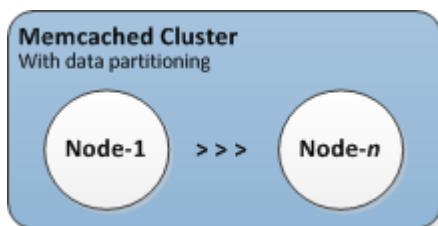
ElastiCache Speicher- und Prozessoranforderungen

Der Grundbaustein von Amazon ElastiCache ist der Node. Knoten werden zur Bildung von Clustern einzeln oder in Gruppierungen konfiguriert. Berücksichtigen Sie bei der Bestimmung des für den Cluster zu verwendenden Knotentyps die Knotenkonfiguration des Clusters und die zu speichernde Datenmenge.

Die Memcached-Engine arbeitet mit mehreren Threads. Daher wirkt sich die Anzahl der Kerne eines Knotens auf die für den Cluster verfügbare Rechenleistung aus.

Konfiguration von Memcached-Cluster

ElastiCache für Memcached bestehen Cluster aus 1 bis 60 Knoten. Die Daten in einem Memcached-Cluster werden über Knoten im Cluster hinweg partitioniert. Ihre Anwendung stellt über eine als Endpunkt bezeichnete Netzwerkadresse eine Verbindung mit einem Memcached-Cluster her. Jeder Knoten in einem Memcached-Cluster verfügt über seinen eigenen Endpunkt. Dieser wird von der Anwendung zum Lesen aus oder Schreiben in einem bestimmten Knoten genutzt. Zusätzlich zu den Knotenendpunkten hat der Memcached-Cluster selbst einen Endpunkt, der als Konfigurationsendpunkt bezeichnet wird. Ihre Anwendung kann diesen Endpunkt verwenden, um aus dem Cluster zu lesen oder in den Cluster zu schreiben, wobei die Bestimmung des Knotens, aus dem gelesen oder in den geschrieben werden soll, Auto Discovery überlassen bleibt.

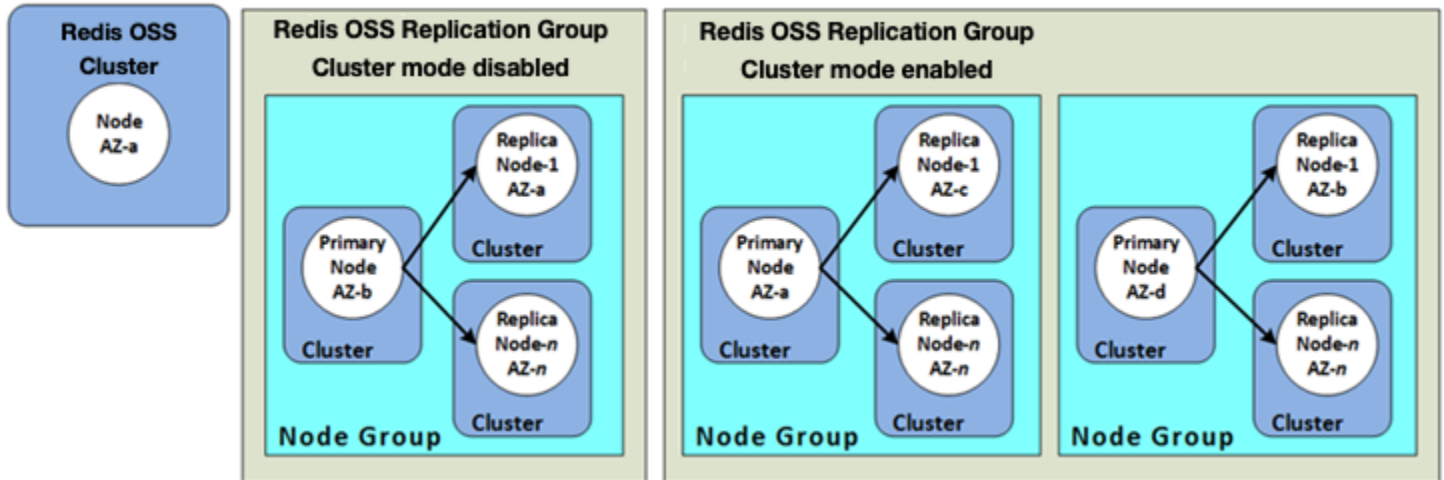


Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Valkey- und Redis OSS-Clusterkonfiguration

ElastiCache für Valkey und Redis bestehen OSS-Cluster aus 0 bis 500 Shards (auch Knotengruppen genannt). Die Daten in einem Valkey- oder Redis OSS-Cluster sind auf die Shards im Cluster verteilt. Ihre Anwendung stellt über eine Netzwerkadresse, die als Endpunkt bezeichnet wird,

eine Verbindung zu einem Valkey- oder Redis-OSS-Cluster her. Die Knoten in einem Valkey- oder Redis OSS-Shard erfüllen eine von zwei Rollen: eine read/write primäre und alle anderen Knoten sind schreibgeschützte Sekundärknoten (auch Read Replicas genannt). Zusätzlich zu den Knotenendpunkten verfügt der Valkey- oder Redis-OSS-Cluster selbst über einen Endpunkt, den sogenannten Konfigurationsendpunkt. Ihre Anwendung kann diesen Endpunkt verwenden, um aus dem Cluster zu lesen oder in ihn zu schreiben. Die Entscheidung, von welchem Knoten gelesen oder auf welchen geschrieben werden soll, bleibt Redis OSS ElastiCache überlassen.



Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

ElastiCache Skalierungsanforderungen

Alle Cluster können hochskaliert werden, indem ein neuer Cluster mit einem neuen, größeren Knotentyp erstellt wird. Wenn Sie einen Memcached-Cluster hochskalieren, ist der neue Cluster zunächst leer. Wenn Sie einen Valkey- oder Redis OSS-Cluster hochskalieren, können Sie ihn von einem Backup aus starten und so verhindern, dass der neue Cluster leer anfängt.

Amazon ElastiCache for Memcached-Cluster können horizontal oder horizontal skaliert werden. Ein Memcached-Cluster wird einfach durch Hinzufügen oder Entfernen von Knoten aus dem Cluster nach oben bzw. nach unten skaliert. Wenn Automatic Discovery aktiviert ist und die Anwendung mit dem Konfigurationsendpunkt des Clusters verbunden ist, sind zum Hinzufügen oder Entfernen von Knoten keine Änderungen an Ihrer Anwendung erforderlich.

Weitere Informationen finden Sie unter [Skalierung ElastiCache](#) in diesem Handbuch.

ElastiCache Zugriffsanforderungen

Standardmäßig wird auf ElastiCache Amazon-Cluster von EC2 Amazon-Instances aus zugegriffen. Der Netzwerkzugriff auf einen ElastiCache Cluster ist auf das Konto beschränkt, mit dem der Cluster

erstellt wurde. Bevor Sie von einer EC2 Amazon-Instance aus auf einen Cluster zugreifen können, müssen Sie daher die EC2 Amazon-Instance autorisieren, auf den Cluster zuzugreifen. Die Schritte dazu variieren, je nachdem, ob Sie mit EC2 VPC oder EC2 -Classic gestartet sind.

Wenn Sie Ihren Cluster in EC2 VPC gestartet haben, müssen Sie dem Cluster Netzwerkzugang gewähren. Wenn Sie Ihren Cluster in EC2 -Classic gestartet haben, müssen Sie der Amazon Elastic Compute Cloud-Sicherheitsgruppe, die mit der Instance verknüpft ist, Zugriff auf Ihre ElastiCache Sicherheitsgruppe gewähren. Detaillierte Anweisungen finden Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) in diesem Handbuch.

Anforderungen an Region, Availability Zone und Lokale Zone für ElastiCache

Amazon ElastiCache unterstützt alle AWS Regionen. Indem Sie Ihre ElastiCache Cluster in einer AWS Region in der Nähe Ihrer Anwendung platzieren, können Sie die Latenz reduzieren. Wenn Ihr Cluster aus mehreren Knoten besteht, können Sie die Auswirkungen von Ausfällen auf Ihren Cluster verringern, indem Sie Ihre Knoten in verschiedenen Availability Zones oder in Local Zones unterbringen.

Weitere Informationen finden Sie hier:

- [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#)
- [Verwenden von lokalen Zonen mit ElastiCache](#)
- [Minimieren von Ausfällen](#)

Auswahl der Knotengröße

Die Knotengröße, die Sie für Ihren ElastiCache Cluster auswählen, wirkt sich auf Kosten, Leistung und Fehlertoleranz aus.

Knotengröße (Valkey und Redis OSS)

Informationen zu den Vorteilen von Graviton-Prozessoren finden Sie unter [AWS -Graviton-Prozessor](#).

Die Beantwortung der folgenden Fragen kann Ihnen helfen, den minimalen Knotentyp zu ermitteln, den Sie für Ihre Valkey- oder Redis OSS-Implementierung benötigen:

- Erwarten Sie durchsatzgebundene Workloads mit mehreren Client-Verbindungen?

Wenn dies der Fall ist und Sie Redis OSS Version 5.0.6 oder höher verwenden, können Sie mit unserer erweiterten I/O Funktion, die, sofern verfügbar CPUs , für die Entlastung der Client-

Verbindungen im Namen der Redis OSS-Engine verwendet wird, einen besseren Durchsatz und eine bessere Latenz erzielen. Wenn Sie Redis OSS Version 7.0.4 oder höher ausführen, lesen Threads jetzt zusätzlich zu den erweiterten I/O, you will get additional acceleration with enhanced I/O multiplexing, where each dedicated network IO thread pipelines commands from multiple clients into the Redis OSS engine, taking advantage of Redis OSS' ability to efficiently process commands in batches. In ElastiCache for Redis OSS v7.1 and above, we extended the enhanced I/O threads functionality to also handle the presentation layer logic. By presentation layer, what we mean is that Enhanced I/O Threads nicht nur die Client-Eingaben, sondern analysieren die Eingabe auch in das Redis OSS-Binärbefehlsformat, das dann zur Ausführung an den Haupt-Thread weitergeleitet wird, was zu einer Leistungssteigerung führt. Weitere Informationen finden Sie im [Blogbeitrag](#) und auf der Seite mit den [unterstützten Versionen](#).

- Haben Sie Workloads, die regelmäßig auf einen kleinen Prozentsatz ihrer Daten zugreifen?

Wenn dies der Fall ist und Sie die Redis OSS-Engine Version 6.2 oder höher verwenden, können Sie das Data-Tiering nutzen, indem Sie den Knotentyp r6gd wählen. Bei Verwendung von Daten-Tiering werden zuletzt verwendete Daten auf dem SSD gespeichert. Beim Abruf entsteht eine geringe Latenzzeit, die jedoch durch Kosteneinsparungen ausgeglichen wird. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Wie viel Gesamtspeicher benötigen Sie für Ihre Daten?

Um eine allgemeine Schätzung zu erhalten, nehmen Sie die Größe der Objekte, die Sie zwischenspeichern möchten. Multiplizieren Sie diese Größe mit der Anzahl der Objekte, die Sie gleichzeitig im Cache halten wollen. Um eine vernünftige Schätzung der Elementgröße zu erhalten, serialisieren Sie zunächst Ihre Cache-Elemente und zählen dann die Zeichen. Teilen Sie diese Zahl dann durch die Anzahl der Shards in Ihrem Cluster.

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Welche Version von Redis OSS verwenden Sie?

Bei Redis OSS-Versionen vor 2.8.22 müssen Sie mehr Speicher für Failover, Snapshots, Synchronisation und das Heraufstufen eines Replikats auf den primären Betrieb reservieren. Diese Anforderung besteht, weil ausreichend Speicher für alle Schreibvorgänge während des Prozesses zur Verfügung stehen muss.

Redis OSS Version 2.8.22 und höher verwenden einen forkless-Speicherprozess, der weniger verfügbaren Speicher benötigt als der vorherige Prozess.

Weitere Informationen finden Sie hier:

- [So werden Synchronisation und Backup implementiert](#)
- [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#)
- Wie hoch ist der Anteil der Schreibvorgänge Ihrer Anwendung?

Anwendungen mit vielen Schreibvorgängen können deutlich mehr verfügbaren Speicher, d. h. Speicher, der nicht von Daten belegt wird, beim Erstellen von Snapshots oder beim Failover erfordern. Wenn der BGSAVE-Prozess ausgeführt wird, müssen Sie über genügend Speicher verfügen, der nicht durch Daten belegt ist, um alle Schreibvorgänge, die während des BGSAVE-Prozesses stattfinden, unterzubringen. Beispiele sind die Erstellung eines Snapshots, die Synchronisierung eines primären Clusters mit einem Replikat in einem Cluster und die Aktivierung der Append-Only-Datei-Funktion (AOF). Ein weiteres ist das Heraufstufen eines Replikats zum primären Knoten (wenn Sie Multi-AZ aktiviert haben). Im schlimmsten Fall werden alle Ihre Daten während des Prozesses neu geschrieben. In diesem Fall benötigen Sie eine Knoten-Instance-Größe mit doppelt so viel Speicher wie für die Daten allein benötigt wird.

Detailliertere Informationen erhalten Sie unter [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#).

- Handelt es sich bei Ihrer Implementierung um einen eigenständigen Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) oder um einen Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) mit mehreren Shards?

Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert)

Wenn Sie einen Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) implementieren, muss Ihr Knotentyp in der Lage sein, all Ihre Daten zuzüglich des erforderlichen Overheads aufzunehmen, wie im vorherigen Punkt bullet.

Nehmen Sie zum Beispiel an, dass die Gesamtgröße aller Ihrer Objekte 12 GB beträgt. In diesem Fall können Sie einen `cache.m3.xlarge`-Knoten mit 13,3 GB Speicher oder einen `cache.r3.large`-Knoten mit 13,5 GB Speicher verwenden. Sie benötigen jedoch möglicherweise mehr Speicher für BGSAVE-Operationen. Wenn Ihre Anwendung sehr schreibintensiv ist, sollten Sie den Speicherbedarf auf mindestens 24 GB verdoppeln. Verwenden Sie also entweder eine `cache.m3.2xlarge` mit 27,9 GB Speicher oder eine `cache.r3.xlarge` mit 30,5 GB Speicher.

Valkey oder Redis OSS (Clustermodus aktiviert) mit mehreren Shards

Wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mit mehreren Shards implementieren, muss der Knotentyp Datenbytes aufnehmen können. $\text{bytes-for-data-and-overhead} / \text{number-of-shards}$

Angenommen, Sie schätzen die Gesamtgröße aller Ihrer Objekte auf 12 GB und Sie haben zwei Shards. In diesem Fall können Sie einen `cache.m3.large`-Knoten mit 6,05 GB Speicher (12 GB / 2) verwenden. Sie benötigen jedoch möglicherweise mehr Speicher für BGSAVE-Operationen. Wenn Ihre Anwendung schreibintensiv ist, verdoppeln Sie die Speicheranforderungen auf mindestens 12 GB pro Shard. Verwenden Sie also entweder eine `cache.m3.xlarge` mit 13,3 GB Speicher oder eine `cache.r3.large` mit 13,5 GB Speicher.

- Verwenden Sie Local Zones?

[Local Zones](#) ermöglichen es Ihnen, Ressourcen wie einen ElastiCache Cluster an mehreren Standorten in der Nähe Ihrer Benutzer zu platzieren. Bei der Auswahl der Knotengröße sollten Sie jedoch beachten, dass die verfügbaren Knotengrößen unabhängig von den Kapazitätsanforderungen derzeit auf die folgenden beschränkt sind:

- Aktuelle Generation:

M5-Knotentypen: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`, `cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

R5-Knotentypen: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

T3-Knotentypen: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Während Ihr Cluster läuft, können Sie die Messwerte für Speicherverbrauch, Prozessorauslastung, Cache-Treffer und Cache-Fehlschläge überwachen, die veröffentlicht werden CloudWatch. Sie werden vielleicht feststellen, dass Ihr Cluster nicht die gewünschte Trefferquote hat oder dass die Schlüssel zu oft entfernt werden. Sie können in diesen Fällen eine andere Knotengröße mit größeren CPU- und Speicherspezifikationen wählen.

Denken Sie bei der Überwachung der CPU-Auslastung daran, dass Valkey und Redis OSS Single-Threading verwenden. Multiplizieren Sie also die gemeldete CPU-Nutzung mit der Anzahl der CPU-Kerne, um die tatsächliche Nutzung zu erhalten. Beispielsweise ist eine Vierkern-CPU, die eine Nutzungsrate von 20 Prozent meldet, tatsächlich die Einkern-CPU von Redis, die mit einer Auslastung von 80 Prozent läuft.

Knotengröße (Memcached)

Memcached-Cluster enthalten gegebenenfalls mehrere Knoten mit über die Knoten partitionierten Cluster-Daten. Aus diesem Grund sind die Speicheranforderungen des Clusters und die eines Knotens zwar ähnlich, jedoch nicht identisch. Sie können Ihre erforderliche Cluster-Speicherkapazität mit wenigen großen Knoten oder mit vielen kleineren Knoten abdecken. Mit wechselnden Anforderungen können Sie Knoten zum Cluster hinzufügen oder aus diesem entfernen und zahlen so nur für die Knoten, die Sie tatsächlich brauchen.

Die Gesamtspeicherkapazität des Clusters wird durch Multiplizieren der Anzahl von Knoten im Cluster mit der RAM-Kapazität der einzelnen Knoten (nach Abzug der für die Systemverwaltung erforderlichen Kapazität) berechnet. Die Kapazität jedes Knotens basiert auf dem Knotentyp.

```
cluster_capacity = number_of_nodes * (node_capacity - system_overhead)
```

Die Anzahl von Knoten im Cluster ist ein Schlüsselfaktor für die Verfügbarkeit Ihres Clusters, der Memcached ausführt. Der Ausfall eines einzelnen Knotens kann Auswirkungen auf die Verfügbarkeit Ihrer Anwendung und die Belastung Ihrer Backend-Datenbank haben. Stellt in einem solchen Fall einen Ersatz ElastiCache für einen ausgefallenen Knoten bereit und dieser wird erneut aufgefüllt. Um diese Auswirkungen auf die Verfügbarkeit zu verringern, sollten Sie die Speicher- und Rechenkapazität auf mehrere Knoten mit geringerer Kapazität verteilen, anstatt weniger Knoten mit hoher Kapazität zu verwenden.

In einem Szenario, in dem Sie 35 GB Cache-Speicher haben möchten, können Sie eine der folgenden Konfigurationen einrichten:

- 11 `cache.t2.medium`-Knoten mit 3,22 GB Speicher und jeweils 2 Threads = 35,42 GB und 22 Threads.
- 6 `cache.m4.large`-Knoten mit 6,42 GB Speicher und jeweils 2 Threads = 38,52 GB und 12 Threads.
- 3 `cache.r4.large`-Knoten mit 12,3 GB Speicher und jeweils 2 Threads = 36,90 GB und 6 Threads.
- 3 `cache.m4.xlarge`-Knoten mit 14,28 GB Speicher und jeweils 4 Threads = 42,84 GB und 12 Threads.

Vergleichen der Knotenoptionen

Knotentyp	Speicher (in GiB)	Kerne	Kosten pro Stunde*	Erforderl iche Knoten	Gesamtspe icher (in GiB)	Kerne gesamt	Monatlich er Preis
cache.t2. medium	3,22	2	0,068 USD	11	35,42	22	538,56 USD
cache.m4. large	6,42	2	0,156 USD	6	38,52	12	673,92 USD
cache.m4. xlarge	14,28	4	0,311 USD	3	42,84	12	671,76 USD
cache.m5. xlarge	12,93	4	0,311 USD	3	38,81	12	671,76 USD
cache.m6g .large	6,85	2	\$0,147	6	41,1	12	\$635
cache.r4. large	12,3	2	0,228 USD	3	36,9	6	492,48 USD
cache.r5. large	13,07	2	0,216 USD	3	39,22	6	466,56 USD
cache.r6g .large	13,07	2	\$0,205	3	42,12	6	\$442

* Preis pro Stunde und Knoten ab 08. Oktober 2020.

Kosten pro Monat bei 100 %-iger Auslastung für 30 Tage (720 Stunden).

Diese Optionen bieten jeweils ähnliche Speicherkapazität bei unterschiedlicher Rechenkapazität und Kosten. Um die Kosten Ihrer spezifischen Optionen zu vergleichen, sehen Sie sich die [ElastiCache Amazon-Preise](#) an.

Für Cluster, die Memcached ausführen, wird ein Teil des verfügbaren Speichers auf jedem Knoten für den Verbindungs-Overhead verwendet. Weitere Informationen finden Sie unter [Overhead von Memcached-Verbindungen](#).

Bei der Verwendung mehrerer Knotenpunkte müssen die Schlüssel auf diese verteilt werden. Jeder Knoten besitzt einen eigenen Endpunkt. Für eine einfache Endpunktverwaltung können Sie ElastiCache die Auto Discovery-Funktion verwenden, mit der Client-Programme automatisch alle Knoten in einem Cluster identifizieren können. Weitere Informationen finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#).

In manchen Fällen sind Sie vielleicht unsicher, wie viel Kapazität Sie benötigen. Wenn das so ist, empfehlen wir, zu Testzwecken mit einem `cache.m5.large`-Knoten zu beginnen. Überwachen Sie dann die Speicherauslastung, die CPU-Auslastung und die Cache-Trefferquote anhand der auf Amazon veröffentlichten ElastiCache Metriken CloudWatch. Weitere Informationen zu CloudWatch Metriken für ElastiCache finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#). Für die Produktion und größere Workloads bieten R5-Knoten die beste Leistung und einen optimalen RAM-Einstandswert.

Wenn Ihr Cluster nicht die gewünschte Trefferquote aufweist, können Sie einfach weitere Knoten hinzufügen, um den gesamten verfügbaren Speicher in Ihrem Cluster zu erhöhen.

Wenn Ihr Cluster durch die CPU begrenzt ist, aber eine ausreichende Trefferquote aufweist, richten Sie einen neuen Cluster mit einem Knotentyp ein, der mehr Rechenleistung bietet.

Einen Cluster für Valkey oder Redis OSS erstellen

Die folgenden Beispiele zeigen, wie ein Valkey- oder Redis-OSS-Cluster mithilfe der AWS-Managementkonsole API, und erstellt wird. AWS CLI ElastiCache

Erstellen eines Valkey- oder Redis-OSS (Clustermodus deaktiviert) (Konsole)

ElastiCache unterstützt die Replikation, wenn Sie die Valkey- oder Redis OSS-Engine verwenden. Um die Latenz zwischen dem Schreiben von Daten in einen read/write primären Valkey- oder Redis OSS-Cluster und der Weitergabe an einen schreibgeschützten sekundären Cluster zu überwachen, ElastiCache fügt dem Cluster einen speziellen Schlüssel hinzu, `ElastiCacheMasterReplicationTimestamp`. Dieser Schlüssel ist die aktuelle Weltzeit (UTC). Da ein Valkey- oder Redis OSS-Cluster möglicherweise zu einem späteren Zeitpunkt zu einer Replikationsgruppe hinzugefügt wird, ist dieser Schlüssel in allen Valkey- oder Redis OSS-Clustern enthalten, auch wenn sie zunächst nicht Mitglieder einer Replikationsgruppe sind. Weitere Informationen zu Replikationsgruppen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

Gehen Sie wie unter beschrieben vor, um einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) zu erstellen. [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)

Sobald der Status Ihres Clusters verfügbar ist, können Sie Amazon EC2 Zugriff darauf gewähren, eine Verbindung herstellen und ihn verwenden. Weitere Informationen erhalten Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Valkey- oder Redis OSS-Clusters (Cluster-Modus aktiviert) (Konsole)

Wenn Sie Redis OSS 3.2.4 oder höher ausführen, können Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) erstellen. Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) unterstützen die Partitionierung Ihrer Daten auf 1 bis 500 Shards (API/CLI: Knotengruppen), jedoch mit einigen Einschränkungen. Einen Vergleich von Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert) finden Sie unter [Unterstützte Engines und Versionen](#)

So erstellen Sie mit der Konsole einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert)
ElastiCache

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der Sie diesen Cluster starten möchten.
3. Wählen Sie im Navigationsbereich Get started (Erste Schritte) aus.
4. Wählen Sie Create VPC (VPC erstellen) aus und befolgen Sie die Schritte unter [Erstellen einer Virtual Private Cloud \(VPC\)](#).
5. Wählen Sie auf der ElastiCache Dashboard-Seite Create Cluster und dann Create Valkey cluster oder Create Redis OSS cluster aus.
6. Führen Sie unter Cluster settings (Cluster-Einstellungen) die folgenden Schritte aus:
 - a. Wählen Sie Configure and create a new cluster (Neuen Cluster konfigurieren und erstellen) aus.
 - b. Wählen Sie für Cluster mode (Cluster-Modus) Enabled (Aktiviert) aus.
 - c. Geben Sie für Cluster info (Cluster-Info) einen Wert für Name (Name) ein.
 - d. (Optional) Geben Sie einen Wert für Description (Beschreibung) ein.
7. Unter Location (Speicherort):

AWS Cloud

1. Wir empfehlen für AWS Cloud die Voreinstellungen für Multi-AZ und Auto-failover (Automatisches Failover) zu akzeptieren. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache für Redis](#) OSS mit Multi-AZ.
2. Unter Cluster settings (Cluster-Einstellungen):
 - a. Wählen Sie unter Engine version (Engine-Version) eine verfügbare Version aus.
 - b. Verwenden Sie für Port den Standardport 6379. Wenn es einen Grund gibt, einen anderen Port zu verwenden, geben Sie die betreffende Portnummer ein.
 - c. Wählen Sie für die Parameter group (Parametergruppe) eine Parametergruppe aus oder erstellen Sie eine neue Parametergruppe. Parametergruppen steuern die Laufzeitparameter Ihres Clusters. Weitere Informationen zu Parametergruppen finden Sie unter [Valkey- und Redis OSS-Parameter](#) und [Eine ElastiCache Parametergruppe erstellen](#).

 Note

Wenn Sie eine Parametergruppe zum Festlegen der Engine-Konfigurationswerte auswählen, wird diese Parametergruppe auf alle Cluster im globalen Datenspeicher angewendet. Auf der Seite Parameter Groups (Parametergruppen) gibt das Ja/nein-Attribut Global an, ob eine Parteigruppe Teil eines globalen Datenspeichers ist.

- d. Klicken Sie für Node type (Knotentyp) auf den Abwärtspfeil (▼).

Wählen Sie im Dialogfeld Change node type (Knotentyp ändern) einen Wert für Instance family (Instance-Familie) für den gewünschten Knotentyp aus. Wählen Sie dann den Knotentyp aus, den Sie für diesen Cluster verwenden möchten, und wählen Sie dann Save (Speichern).

Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Wenn Sie einen r6gd-Knotentyp wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

- e. Wählen Sie unter Anzahl der Shards die Anzahl der Shards (Partitionen/Knotengruppen) aus, die Sie für diesen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) benötigen.

Bei einigen Versionen von Valkey oder Redis OSS (Clustermodus aktiviert) können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern:

- Redis OSS 3.2.10 und höher — Wenn auf Ihrem Cluster Redis OSS 3.2.10 oder spätere Versionen ausgeführt werden, können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern. Weitere Informationen finden Sie unter [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#).
- Andere Redis OSS-Versionen — Wenn auf Ihrem Cluster eine Version von Redis OSS vor Version 3.2.10 ausgeführt wird, gibt es einen anderen Ansatz. Um die Anzahl der Shards in Ihrem Cluster in diesem Fall zu ändern, erstellen Sie einen neuen Cluster mit der neuen Anzahl von Shards. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).

- f. Wählen Sie für Replicas per shard (Replikate pro Shard) die Anzahl der Read Replica-Knoten aus, die sich in jedem Shard befinden sollen.

Die folgenden Einschränkungen gelten für Valkey oder Redis OSS (Clustermodus aktiviert).

- Wenn Sie Multi-AZ aktiviert haben, stellen Sie sicher, dass mindestens ein Replikat pro Shard vorhanden ist.
- Die Anzahl der Replikate ist für jeden Shard gleich, wenn der Cluster mithilfe der Konsole erstellt wird.
- Die Anzahl der Lesereplikate pro Shard ist ein fester Wert, der nicht geändert werden kann. Wenn Sie feststellen, dass Sie mehr oder weniger Replikate pro Shard (API/CLI: Knotengruppe) benötigen, müssen Sie einen neuen Cluster mit der neuen Anzahl von Replikaten erstellen. Weitere Informationen finden Sie unter [Tutorial: Seeding eines neuen knotenbasierten Clusters mit einem extern erstellten Backup](#).

3. Unter Connectivity (Konnektivität)

- a. Wählen Sie als Network type (Netzwerktyp) die IP-Version(en) aus, die dieser Cluster unterstützen soll.
- b. Wählen Sie für Subnetzgruppen das Subnetz aus, das Sie auf diesen Cluster anwenden möchten. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten zugeordnet werden sollen. ElastiCache Cluster benötigen ein Dual-Stack-Subnetz mit beiden IPv4 und ihnen zugewiesenen IPv6 Adressen, um im Dual-Stack-Modus betrieben werden zu können, und ein Subnetz nur für den Betrieb als IPv6 -only. IPv6

Geben Sie beim Erstellen einer neuen Subnetzgruppe die VPC ID (VPC-ID) ein, zu der sie gehört.

Wählen Sie einen Discovery IP type (Erkennungs-IP-Typ) aus. Es werden nur die IP-Adressen des von Ihnen ausgewählten Protokolls zurückgegeben.

Weitere Informationen finden Sie unter:

- [Auswahl eines Netzwerktyps in ElastiCache](#).
- [Erstellen eines Subnetzes in der VPC](#)

Wenn Sie [Verwenden von lokalen Zonen mit ElastiCache](#) sind, müssen Sie ein Subnetz erstellen oder auswählen, das in der lokalen Zone liegt.

Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).

4. Für Availability zone placements (Availability-Zone-Platzierungen) haben Sie zwei Optionen:

- Keine Präferenz — wählt die Availability Zone. ElastiCache
- Availability Zones angeben – Sie geben die Availability Zone für jeden Cluster an.

Wenn Sie die Availability Zones angeben, wählen Sie für jeden Cluster in jedem Shard die Availability Zone aus der Liste aus.

Weitere Informationen finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

5. Wählen Sie Next (Weiter).
6. Unter Erweiterte Valkey-Einstellungen oder Erweiterte Redis OSS-Einstellungen oder

- Für Security (Sicherheit):
 - i. Zur Verschlüsselung Ihrer Daten haben Sie die folgenden Optionen:
 - Verschlüsselung im Ruhezustand – Ermöglicht die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

 Note

Sie haben die Möglichkeit, einen anderen Verschlüsselungsschlüssel anzugeben, indem Sie Customer Managed AWS KMS Key und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwendung von kundenverwalteten Schlüsseln aus AWS-KMS](#).

- Verschlüsselung während der Übertragung – Ermöglicht die Verschlüsselung von Daten während der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey 7.2

und höher oder Redis OSS 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:

- Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
- Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern aus, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
- AUTH Default User — Ein Authentifizierungsmechanismus für einen Valkey- oder Redis OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).
- AUTH — Ein Authentifizierungsmechanismus für den Valkey- oder Redis-OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).

 Note

Für Redis OSS-Versionen zwischen 3.2.6 und höher, mit Ausnahme von Version 3.2.10, ist AUTH die einzige Option.

- ii. Wählen Sie für Security groups (Sicherheitsgruppen) die gewünschten Sicherheitsgruppen für diesen Cluster aus. Eine security group (Sicherheitsgruppe) fungiert als Firewall, um den Netzwerkzugriff auf Ihren Cluster zu steuern. Sie können die Standardsicherheitsgruppe für Ihre VPC verwenden oder eine neue erstellen.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Sicherheitsgruppen für Ihre VPC](#) im Benutzerhandbuch zu Amazon VPC.
7. Wenn Sie regelmäßig geplante automatische Sicherungen möchten, aktivieren Sie Enable automatic backups und geben Sie ein, wie viele Tage lang jede automatische Sicherung beibehalten werden soll, bevor sie automatisch gelöscht wird. Wenn Sie keine regelmäßig geplanten automatischen Sicherungen möchten, deaktivieren Sie das Kontrollkästchen Enable automatic backups. In beiden Fällen haben Sie jederzeit die Option, manuelle Sicherungen zu erstellen.

Weitere Informationen zu Sicherung und Wiederherstellung finden Sie unter [Snapshot und Wiederherstellung](#).

8. (Optional) Geben Sie ein Wartungsfenster an. Das Maintenance window (Wartungsfenster) ist der (in der Regel eine Stunde lange) allwöchentliche Zeitraum, für den ElastiCache die Systemwartung Ihres Clusters plant. Sie können es ElastiCache überlassen, einen Tag und eine Uhrzeit für das Wartungsfenster auszuwählen (No preference (Keine Präferenz)) oder Sie können den Tag, die Uhrzeit und die Dauer selbst wählen (Specify maintenance window (Wartungsfenster angeben)). Treffen Sie bei Wahl von Specify maintenance window eine Auswahl in den Listen Start day, Start time und Duration (in Stunden) für Ihr Wartungsfenster. Alle Zeiten sind UCT-Zeiten.

Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

9. (Optional) Für Protokolle:
 - Unter Protokollformat wählen Sie entweder Text oder JSON aus.
 - Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen aus und geben Sie entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und wählen Sie dann entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Streamnamen aus.
10. Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen unter Tags zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).
11. Wählen Sie Weiter aus.
12. Überprüfen Sie alle Ihre Einträge und ausgewählten Optionen und machen Sie dann evtl. erforderliche Korrekturen. Sobald Sie bereit sind, klicken Sie auf Create (Erstellen).

On premises

1. Für On premises (On-Premises) empfehlen wir Ihnen, Auto-failover (Automatisches Failover) aktiviert zu lassen. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache für Redis OSS](#) mit Multi-AZ
2. Befolgen Sie die Schritte unter [Verwenden von Outposts](#).

Um das Äquivalent mithilfe der ElastiCache API oder AWS CLI anstelle der ElastiCache Konsole zu erstellen, gehen Sie wie folgt vor:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Sobald der Status Ihres Clusters verfügbar ist, können Sie EC2 Zugriff darauf gewähren, eine Verbindung herstellen und ihn verwenden. Weitere Informationen erhalten Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

 Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Clusters (AWS CLI)

Verwenden Sie den `create-cache-cluster` Befehl AWS CLI, um einen Cluster mit dem zu erstellen.

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) (CLI)

Example— Ein Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) ohne Read Replicas

Der folgende CLI-Code erstellt einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) ohne Replikate.

Note

Wenn Sie einen Cluster mit einem Knotentyp aus der R6gd-Familie erstellen, müssen Sie den Parameter `data-tiering-enabled` übergeben.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine redis \  
--num-cache-nodes 1 \  
--cache-parameter-group default.redis6.x \  
--snapshot-arns arn:aws:s3:::amzn-s3-demo-bucket/snapshot.rdb
```

Für Windows:

```
aws elasticache create-cache-cluster ^
```

```
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine redis ^  
--num-cache-nodes 1 ^  
--cache-parameter-group default.redis6.x ^  
--snapshot-arns arn:aws:s3:::amzn-s3-demo-bucket/snapshot.rdb
```

Erstellen eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) (AWS CLI)

Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) (API/CLI: Replikationsgruppen) können mit diesem Vorgang nicht erstellt werden. `create-cache-cluster` Informationen zum Erstellen eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) finden Sie unter: [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Cluster Mode Enabled\) von Grund auf neu erstellen \(AWS CLI\)](#)

Weitere Informationen finden Sie im Referenzthema AWS CLI ElastiCache [create-replication-group](#)

Einen Cluster für Valkey oder Redis OSS (ElastiCache API) erstellen

Verwenden Sie die Aktion, um einen Cluster mithilfe der ElastiCache API zu erstellen.

`CreateCacheCluster`

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Themen

- [Erstellen eines Valkey- oder Redis OSS-Clusters \(Clustermodus deaktiviert\) \(ElastiCache API\)](#)
- [Erstellen eines Clusters in Valkey oder Redis OSS \(Clustermodus aktiviert\) \(API\) ElastiCache](#)

Erstellen eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) (ElastiCache API)

Der folgende Code erstellt einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) (API).
ElastiCache

Die Zeilenumbrüche dienen der besseren Lesbarkeit.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=CreateCacheCluster  
  &CacheClusterId=my-cluster  
  &CacheNodeType=cache.r4.large  
  &CacheParameterGroup=default.redis3.2  
  &Engine=redis  
  &EngineVersion=3.2.4  
  &NumCacheNodes=1  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &SnapshotArns.member.1=arn%3Aaws%3As3%3A%3A%3AmyS3Bucket%2Fdump.rdb  
  &Timestamp=20150508T220302Z  
  &Version=2015-02-02  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Credential=<credential>  
  &X-Amz-Date=20150508T220302Z  
  &X-Amz-Expires=20150508T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Signature=<signature>
```

Erstellen eines Clusters in Valkey oder Redis OSS (Clustermodus aktiviert) (API) ElastiCache

Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) (API/CLI: Replikationsgruppen) können mit diesem Vorgang nicht erstellt werden. `CreateCacheCluster` Informationen zum Erstellen eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) finden Sie unter. [Eine Replikationsgruppe in Valkey oder Redis OSS \(Cluster Mode Enabled\) von Grund auf neu erstellen \(ElastiCache API\)](#)

Weitere Informationen finden Sie im API-Referenzthema. ElastiCache [CreateReplicationGroup](#)

Einen Cluster für Memcached erstellen

Die folgenden Beispiele zeigen, wie ein Cluster mithilfe der API AWS-Managementkonsole, AWS CLI und ElastiCache erstellt wird.

Erstellen eines Memcached-Clusters (Konsole)

Wenn Sie die Memcached-Engine verwenden, ElastiCache unterstützt Amazon die horizontale Partitionierung Ihrer Daten über mehrere Knoten. Da Memcached Auto Discovery ermöglicht, brauchen Sie die Endpunkte eines jeden Knotens nicht zu verfolgen. Memcached verfolgt

den Endpunkt jedes Knotens und aktualisiert die Endpunktliste beim Hinzufügen oder Entfernen von Knoten. Ihre Anwendung braucht nur mit dem Cluster zu kommunizieren, der als Konfigurationsendpunkt fungiert.

Um einen Memcache-Cluster über die Konsole zu erstellen, folgen Sie den Schritten unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#). Wenn Sie Schritt fünf erreicht haben, wählen Sie Create Memcached Cache aus.

Sobald der Status Ihres Clusters verfügbar ist, können Sie Amazon EC2 Zugriff darauf gewähren, eine Verbindung herstellen und ihn verwenden. Weitere Informationen finden Sie in den ähnlichen Schritten [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

 Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Clusters (AWS CLI)

Verwenden Sie den `create-cache-cluster` Befehl AWS CLI, um einen Cluster mit dem zu erstellen.

 Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Memcached-Cache-Clusters (AWS CLI)

Der folgende CLI-Code erstellt einen Memcached-Cluster mit 3 Knoten.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine memcached \  
--engine-version 1.4.24 \  
--cache-parameter-group default.memcached1.4 \  
--num-cache-nodes 3
```

Für Windows:

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine memcached ^  
--engine-version 1.4.24 ^  
--cache-parameter-group default.memcached1.4 ^  
--num-cache-nodes 3
```

Einen Cluster für Memcached (API) erstellen ElastiCache

Verwenden Sie die Aktion, um mithilfe der ElastiCache API einen Cluster zu erstellen.

CreateCacheCluster

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Themen

- [Einen Memcached-Cluster \(API\) ElastiCache erstellen](#)

Einen Memcached-Cluster (API) ElastiCache erstellen

Der folgende Code erstellt einen Memcache-Cluster mit 3 Knoten (API). ElastiCache

Die Zeilenumbrüche dienen der besseren Lesbarkeit.


```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=CreateCacheCluster  
  &CacheClusterId=my-cluster  
  &CacheNodeType=cache.r4.large  
  &Engine=memcached  
  &NumCacheNodes=3  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150508T220302Z  
  &Version=2015-02-02  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Credential=<credential>  
  &X-Amz-Date=20150508T220302Z  
  &X-Amz-Expires=20150508T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Signature=<signature>
```

Details eines ElastiCache Clusters anzeigen

Sie können detaillierte Informationen zu einem oder mehreren Clustern mithilfe der ElastiCache Konsole oder der ElastiCache API anzeigen. AWS CLI

Details eines Memcached-Clusters anzeigen (Konsole)

Sie können die Details eines Memcached-Clusters mithilfe der ElastiCache Konsole, des For-Clusters oder der AWS CLI API ElastiCache anzeigen. ElastiCache

Das folgende Verfahren beschreibt, wie Sie die Details eines Memcache-Clusters mithilfe der Konsole anzeigen. ElastiCache

So zeigen Sie die Details eines Memcached-Clusters an

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, für die Sie sich interessieren.
3. Wählen Sie im ElastiCache Engine-Dashboard Memcached aus. Dadurch wird eine Liste all Ihrer Cluster angezeigt, die auf der Memcached-Engine ausgeführt werden.
4. Um Details eines Clusters anzuzeigen, aktivieren Sie das Kontrollkästchen links neben den Namen des Clusters.
5. Wenn Sie Knoteninformationen anzeigen möchten, wählen Sie die Registerkarte Nodes (Knoten) aus, auf der Informationen zum Status und zum Endpunkt der Knoten angezeigt werden.
6. Wenn Sie Metriken anzeigen möchten, wählen Sie die Registerkarte Metrics (Metriken) aus, auf der die relevanten Metriken für alle Knoten im Cluster angezeigt werden. Weitere Informationen finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#).
7. Wählen Sie die Registerkarte Network and security (Netzwerk und Sicherheit) aus, um Details zur Konfiguration von Netzwerkkonnektivität und Subnetzgruppen sowie zur VPC-Sicherheitsgruppe des Clusters anzuzeigen. Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).
8. Wählen Sie die Registerkarte Maintenance (Wartung) aus, um Details zu den Wartungseinstellungen des Clusters anzuzeigen. Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

9. Wählen Sie die Registerkarte Tags aus, um Details zu allen Tags anzuzeigen, die auf Cluster-Ressourcen angewendet wurden. Weitere Informationen finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).

Details zu Valkey oder Redis OSS (Clustermodus deaktiviert) anzeigen (Konsole)

Sie können die Details eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) mithilfe der ElastiCache Konsole, des For-Clusters oder der AWS CLI API ElastiCache anzeigen. ElastiCache

Das folgende Verfahren beschreibt, wie Sie die Details eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) mithilfe der Konsole anzeigen. ElastiCache

So zeigen Sie die Details eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) an

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Engine-Dashboard entweder Valkey oder Redis OSS aus, um eine Liste all Ihrer Cluster anzuzeigen, die auf dieser Engine laufen.
3. Markieren Sie das Kontrollkästchen links neben dem Cluster-Namen, um die Details zum Cluster anzuzeigen. Stellen Sie sicher, dass Sie einen Cluster auswählen, auf dem die Valkey- oder Redis OSS-Engine ausgeführt wird, und nicht Clustered Valkey oder Clustered Redis OSS. Damit werden Details des Clusters angezeigt, einschließlich des primären Endpunkts des Clusters.
4. Anzeigen der Knoteninformationen:
 - a. Wählen Sie den Cluster-Namen aus.
 - b. Wählen Sie die Registerkarte Shards and Nodes (Shards und Knoten) aus. Es werden die Details zu jedem Knoten angezeigt, einschließlich des Knotenendpunkts, der zum Lesen vom Cluster benötigt wird.
5. Wenn Sie Metriken anzeigen möchten, wählen Sie die Registerkarte Metrics (Metriken) aus, auf der die relevanten Metriken für alle Knoten im Cluster angezeigt werden. Weitere Informationen finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#).
6. Wählen Sie zum Anzeigen von Protokollen die Registerkarte Logs (Protokolle) aus. Sie gibt an, ob der Cluster Slow-Protokolle oder Engine-Protokolle verwendet, und enthält relevante Details. Weitere Informationen finden Sie unter [Protokollzustellung](#).

7. Wählen Sie die Registerkarte Network and security (Netzwerk und Sicherheit) aus, um Details zur Konfiguration von Netzwerkkonnektivität des Clusters und zur Subnetzgruppe anzuzeigen. Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).
8. Wählen Sie die Registerkarte Maintenance (Wartung) aus, um Details zu den Wartungseinstellungen des Clusters anzuzeigen. Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).
9. Wählen Sie die Registerkarte Service updates (Service-Updates) aus, um Details zu allen verfügbaren Service-Updates sowie deren empfohlenes Mindesthaltbarkeitsdatum einzusehen. Weitere Informationen finden Sie unter [Service-Updates in ElastiCache](#).
10. Wählen Sie die Registerkarte Tags aus, um Details zu allen Tags anzuzeigen, die auf Cluster-Ressourcen angewendet wurden. Weitere Informationen finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).

Details für einen Valkey- oder Redis OSS-Cluster (Cluster Mode Enabled) anzeigen (Konsole)

Sie können die Details eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) mithilfe der ElastiCache Konsole, des For-Clusters oder der AWS CLI API ElastiCache anzeigen. ElastiCache

Das folgende Verfahren beschreibt, wie Sie die Details eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) mithilfe der Konsole anzeigen. ElastiCache

So zeigen Sie die Details eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) an

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, für die Sie sich interessieren.
3. Wählen Sie im ElastiCache Engine-Dashboard Valkey oder Redis OSS aus, um eine Liste all Ihrer Cluster anzuzeigen, die auf dieser Engine laufen.
4. Um Details zu einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) anzuzeigen, wählen Sie das Feld links neben dem Namen des Clusters aus. Stellen Sie sicher, dass Sie einen Cluster auswählen, auf dem die Valkey- oder Clustered Redis OSS-Engine ausgeführt wird.

Der Bildschirm wird unterhalb des Clusters erweitert und zeigt Details zum Cluster an, darunter auch den Konfigurationsendpunkt des Clusters.

5. Wenn Sie eine Auflistung der Shards des Clusters und der Anzahl der Knoten in jedem Shard anzeigen möchten, wählen Sie die Registerkarte Shards and nodes (Shards und Knoten) aus.
6. So zeigen Sie spezifische Informationen zu einem Knoten an:
 - Wählen Sie die ID des Shards aus.

Damit werden Informationen zu jedem Knoten angezeigt, einschließlich des Endpunkts des Knotens, den Sie zum Lesen von Daten aus dem Cluster benötigen.

7. Wenn Sie Metriken anzeigen möchten, wählen Sie die Registerkarte Metrics (Metriken) aus, auf der die relevanten Metriken für alle Knoten im Cluster angezeigt werden. Weitere Informationen finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#).
8. Wählen Sie zum Anzeigen von Protokollen die Registerkarte Logs (Protokolle) aus. Sie gibt an, ob der Cluster Slow-Protokolle oder Engine-Protokolle verwendet, und enthält relevante Details. Weitere Informationen finden Sie unter [Protokollzustellung](#).
9. Wählen Sie die Registerkarte Network and security (Netzwerk und Sicherheit) aus, um Details zur Konfiguration von Netzwerkkonnektivität und Subnetzgruppen des Clusters, zur VPC-Sicherheitsgruppe sowie welche Verschlüsselungsmethode für den Cluster aktiviert ist, sofern zutreffend. Weitere Informationen erhalten Sie unter [Subnetze und Subnetzgruppen](#) und [Datensicherheit bei Amazon ElastiCache](#).
10. Wählen Sie die Registerkarte Maintenance (Wartung) aus, um Details zu den Wartungseinstellungen des Clusters anzuzeigen. Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).
11. Wählen Sie die Registerkarte Service updates (Service-Updates) aus, um Details zu allen verfügbaren Service-Updates sowie deren empfohlenes Mindesthaltbarkeitsdatum einzusehen. Weitere Informationen finden Sie unter [Service-Updates in ElastiCache](#).
12. Wählen Sie die Registerkarte Tags aus, um Details zu allen Tags anzuzeigen, die auf Cluster-Ressourcen angewendet wurden. Weitere Informationen finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).

Details eines ElastiCache Clusters anzeigen (AWS CLI)

Der folgende Code listet die Details für auf *my-cluster*:

```
aws elasticache describe-cache-clusters --cache-cluster-id my-cluster
```

my-cluster Ersetzen Sie es durch den Namen Ihres Clusters, wenn der Cluster mit einem Cache-Knoten und 0 Shards mithilfe des `create-cache-cluster` Befehls erstellt wurde.

```
{
  "CacheClusters": [
    {
      "CacheClusterStatus": "available",
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "Engine": "redis",
      "PreferredMaintenanceWindow": "wed:12:00-wed:13:00",
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "08:30-09:30",
      "TransitEncryptionEnabled": false,
      "AtRestEncryptionEnabled": false,
      "CacheClusterId": "my-cluster1",
      "CacheClusterCreateTime": "2018-02-26T21:06:43.420Z",
      "PreferredAvailabilityZone": "us-west-2c",
      "AuthTokenEnabled": false,
      "PendingModifiedValues": {},
      "CacheNodeType": "cache.r4.large",
      "DataTiering": "disabled",
      "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis3.2"
      },
      "SnapshotRetentionLimit": 0,
      "AutoMinorVersionUpgrade": true,
      "EngineVersion": "3.2.10",
      "CacheSecurityGroups": [],
      "NumCacheNodes": 1
    }
  ]
}
```

```
{
  "CacheClusters": [
    {
```

```

    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2a",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-001",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": false,

```

```

    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",

```



```

    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis3.2"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.10",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
}

```

```

{
  "CacheClusters": [
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "AuthTokenEnabled": true,
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "12:30-13:30",
      "AutoMinorVersionUpgrade": true,
      "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
      "CacheClusterStatus": "available",
      "AtRestEncryptionEnabled": true,
      "PreferredAvailabilityZone": "us-west-2a",
      "TransitEncryptionEnabled": true,
      "ReplicationGroupId": "my-cluster3",
      "Engine": "redis",
      "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
      "CacheClusterId": "my-cluster3-0001-001",
      "PendingModifiedValues": {},
      "CacheNodeType": "cache.r4.large",

```

```

    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
},
{
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0001-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis3.2.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.6",
    "CacheSecurityGroups": [],

```

```

        "NumCacheNodes": 1
    },
    {
        "SecurityGroups": [
            {
                "Status": "active",
                "SecurityGroupId": "sg-dbe93fa2"
            }
        ],
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
        "AuthTokenEnabled": true,
        "CacheSubnetGroupName": "default",
        "SnapshotWindow": "12:30-13:30",
        "AutoMinorVersionUpgrade": true,
        "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
        "CacheClusterStatus": "available",
        "AtRestEncryptionEnabled": true,
        "PreferredAvailabilityZone": "us-west-2c",
        "TransitEncryptionEnabled": true,
        "ReplicationGroupId": "my-cluster3",
        "Engine": "redis",
        "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
        "CacheClusterId": "my-cluster3-0001-003",
        "PendingModifiedValues": {},
        "CacheNodeType": "cache.r4.large",
        "DataTiering": "disabled",
        "CacheParameterGroup": {
            "CacheNodeIdsToReboot": [],
            "ParameterApplyStatus": "in-sync",
            "CacheParameterGroupName": "default.redis6.x.cluster.on"
        },
        "SnapshotRetentionLimit": 0,
        "EngineVersion": "6.0",
        "CacheSecurityGroups": [],
        "NumCacheNodes": 1
    },
    {
        "SecurityGroups": [
            {
                "Status": "active",
                "SecurityGroupId": "sg-dbe93fa2"
            }
        ],
    },

```

```

    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-001",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
},
{
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,

```

```

    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis3.2.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.6",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
},
{
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2a",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",

```

```
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  }
]
```

Wenn der Cluster mit dem erstellt wird AWS-Managementkonsole (Clusterknoten aktiviert oder deaktiviert mit einem oder mehreren Shards), verwenden Sie den folgenden Befehl, um die Clusterdetails zu beschreiben (*my-cluster* ersetzen Sie ihn durch den Namen der Replikationsgruppe (Name Ihres Clusters)):

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

Weitere Informationen finden Sie im ElastiCache Thema AWS CLI [describe-cache-clusters](#) für.

Details eines ElastiCache Clusters anzeigen (ElastiCache API)

Sie können die Details für einen Cluster mithilfe der ElastiCache DescribeCacheClusters API-Aktion anzeigen. Wenn der Parameter CacheClusterId eingeschlossen wird, werden Details für den angegebenen Cluster zurückgegeben. Wenn der Parameter CacheClusterId weggelassen wird, werden Details für bis zu MaxRecords (Standard 100) Cluster zurückgegeben. Der Wert für MaxRecords darf nicht kleiner als 20 oder größer als 100 sein.

Der folgende Code listet die Details für my-cluster auf.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheClusters
&CacheClusterId=my-cluster
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Der folgende Code listet die Details für bis zu 25 Cluster auf.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&MaxRecords=25  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie im ElastiCache API-Referenzthema [DescribeCacheClusters](#).

Einen ElastiCache Cluster ändern

Neben dem Hinzufügen oder Entfernen von Knoten aus einem ElastiCache Cluster kann es vorkommen, dass Sie weitere Änderungen vornehmen müssen, z. B. das Hinzufügen einer Sicherheitsgruppe, das Ändern des Wartungsfensters oder einer Parametergruppe.

Es wird empfohlen, dass das Wartungsfenster in den Zeitraum mit der geringsten Nutzung fällt. Dies muss folglich zeitweise korrigiert werden.

Wenn Sie die Parameter eines Clusters ändern, wird die Änderung entweder sofort oder nach dem Neustart des Clusters angewendet. Dies gilt unabhängig davon, ob Sie die Parametergruppe des Clusters oder einen Parameterwert innerhalb der Parametergruppe des Clusters ändern. Informationen darüber, wann eine bestimmte Parameteränderung angewendet wird, finden Sie im Abschnitt „Änderungen werden wirksam“ in der Spalte „Details“ in den Tabellen für [Memcached-spezifische Parameter](#) und [Valkey- und Redis OSS-Parameter](#). Weitere Informationen zum Neustarten eines Cluster-Knotens finden Sie unter [Knoten neu starten](#).

Unter Verwendung der ElastiCache AWS-Managementkonsole

So modifizieren Sie einen Cluster:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der sich der Cluster befindet, den Sie ändern möchten.
3. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, den Sie modifizieren möchten.

Es wird eine Liste der ausgewählten Engine-Cluster angezeigt.

4. Wählen Sie in der Liste der Cluster den Namen des Clusters aus, den Sie modifizieren möchten..
5. Wählen Sie Aktionen und dann Ändern.

Das Fenster Modify Cluster wird angezeigt.

6. Nehmen Sie im Fenster Modify Cluster die gewünschten Änderungen vor. Zu den Optionen gehören:

- Beschreibung

- Clustermodus – Wenn Sie den Clustermodus von Deaktiviert auf Aktiviert ändern möchten, müssen Sie den Clustermodus zuerst auf Kompatibel festlegen.

Im kompatiblen Modus können Ihre Valkey- oder Redis OSS-Clients sowohl im Clustermodus als auch im deaktivierten Clustermodus eine Verbindung herstellen. Nachdem Sie alle Valkey- oder Redis OSS-Clients zur Verwendung des aktivierten Clustermodus migriert haben, können Sie die Konfiguration des Clustermodus abschließen und den Clustermodus auf Aktiviert setzen.

- Kompatibilität der Engine-Version

 Important

Sie können auf neuere Engine-Versionen aktualisieren. Falls Sie eine Hauptversion der Engine aktualisieren, z. B. von 5.0.6 auf 6.0, müssen Sie eine Parametergruppenfamilie auswählen, die mit der neuen Engine-Version kompatibel ist. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#). Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

- VPC Security group(s) (VPC-Sicherheitsgruppe(n))
- Parametergruppe
- Node Type

 Note

Wenn der Cluster einen Knotentyp aus der R6gd-Familie verwendet, können Sie nur eine andere Knotengröße aus dieser Familie auswählen. Wenn Sie einen Knotentyp aus der R6gd-Familie wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Daten-Tiering](#).

- Multi-AZ
- Automatisches Failover (nur im deaktivierten Cluster-Modus)
- Aktivieren automatischer Sicherungen
- ID des Backup-Knotens
- Aufbewahrungszeitraum für Backups
- Backup-Fenster

- Thema für SNS-Benachrichtigung
- Versionskompatibilität der Memcached Engine
- Netzwerktyp

 Note

Wenn Sie von IPv4 zu wechseln IPv6, müssen Sie Subnetzgruppen auswählen oder erstellen, die kompatibel sind mit IPv6. Weitere Informationen finden Sie unter [Auswahl eines Netzwerktyps in ElastiCache](#).

- VPC Security group(s) (VPC-Sicherheitsgruppe(n))
- Parametergruppe
- Wartungsfenster
- Thema für SNS-Benachrichtigung

Das Feld Sofort anwenden gilt nur für Änderungen der Engine-Version und des Knotentyps. Um die Änderungen sofort zu übernehmen, aktivieren Sie das Kontrollkästchen Apply Immediately (Sofort anwenden). Wenn dieses Kontrollkästchen nicht aktiviert ist, werden Änderungen an der Engine-Version während des nächsten Wartungsfensters übernommen. Andere Änderungen, wie z. B. Ändern des Wartungsfensters, werden umgehend übernommen.

Um die Lieferung für Redis zu enable/disable protokollieren

1. Suchen Sie in der Liste der Cluster den Cluster aus, den Sie ändern möchten. Wählen Sie den Clusternamen und nicht das Kontrollkästchen daneben.
2. Wählen Sie auf der Seite mit den Cluster-Details die Registerkarte Logs aus.
3. Um langsame Logs zu aktivieren oder zu deaktivieren, wählen Sie entweder Aktivieren oder Deaktivieren.

Wenn Sie „Aktivieren“ wählen:

- a. Wählen Sie unter Log format (Protokollformat) entweder JSON oder Text aus.
- b. Wählen Sie unter Protokollzieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.

- c. Unter Protokollziel können Sie Create new auswählen und entweder Ihren CloudWatchLogs Log-Gruppennamen oder Ihren Kinesis Data Firehose-Stream-Namen eingeben. Sie können auch Select existing auswählen und dann entweder Ihren CloudWatchLogs Loggruppennamen oder Ihren Kinesis Data Firehose Firehose-Streamnamen wählen.
- d. Wählen Sie Enable (Aktivieren) aus.

So ändern Sie Ihre Konfiguration für Redis:

1. Wählen Sie Ändern aus.
2. Wählen Sie unter Log format (Protokollformat) entweder JSON oder Text aus.
3. Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
4. Wählen Sie unter Protokollziel entweder Neu erstellen aus und geben Sie Ihren CloudWatchLogs Protokollgruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen ein. Oder wählen Sie Select existing und wählen Sie dann Ihren CloudWatchLogs Log-Gruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen.

Verwenden Sie das mit AWS CLI ElastiCache

Sie können einen vorhandenen Cluster mithilfe der AWS CLI `modify-cache-cluster` Operation ändern. Um den Konfigurationswert eines Clusters zu ändern, geben Sie die ID des Clusters, den zu ändernden Parameter und den neuen Wert des Parameters ein. Das folgende Beispiel ändert das Wartungsfenster für einen Cluster namens `my-cluster` und übernimmt die Änderung umgehend.

 Important

Sie können ein Upgrade auf neuere Versionen der Memcached Engine durchführen. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#). Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

 Important

Sie können auf neuere Valkey- oder Redis OSS-Engine-Versionen aktualisieren. Wenn Sie wichtige Engine-Versionen aktualisieren, z. B. von Redis OSS 5.0.6 auf Redis OSS 6.0, müssen Sie eine Parametergruppenfamilie auswählen, die mit der neuen Engine-

Version kompatibel ist. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#). Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

Der Parameter `--apply-immediately` gilt nur für Änderungen am Knotentyp, an der Engine-Version und der Anzahl von Knoten in einem Cluster. Wenn Sie solche Änderungen umgehend übernehmen möchten, verwenden Sie den Parameter `--apply-immediately`. Wenn Sie solche Änderungen lieber bis zum nächsten Wartungsfenster aufschieben möchten, verwenden Sie den Parameter `--no-apply-immediately`. Andere Änderungen, wie z. B. Ändern des Wartungsfensters, werden umgehend übernommen.

Weitere Informationen finden Sie im Thema für AWS CLI ElastiCache [modify-cache-cluster](#)

Verwenden der ElastiCache API

Sie können einen vorhandenen Cluster mithilfe der ElastiCache `ModifyCacheCluster` API-Operation ändern. Um den Konfigurationswert eines Clusters zu ändern, geben Sie die ID des Clusters, den zu ändernden Parameter und den neuen Wert des Parameters ein. Das folgende Beispiel ändert das Wartungsfenster für einen Cluster namens `my-cluster` und übernimmt die Änderung umgehend.

Important

Sie können ein Upgrade auf neuere Versionen der Memcached Engine durchführen. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#).

Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

 Important

Sie können auf neuere Valkey- oder Redis OSS-Engine-Versionen aktualisieren. Wenn Sie wichtige Engine-Versionen aktualisieren, z. B. von Redis OSS 5.0.6 auf Redis OSS 6.0, müssen Sie eine Parametergruppenfamilie auswählen, die mit der neuen Engine-Version kompatibel ist. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#). Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

Die Zeilenumbrüche dienen der besseren Lesbarkeit.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&CacheClusterId=my-cluster  
&PreferredMaintenanceWindow=sun:23:00-mon:02:00  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150901T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150901T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Der Parameter `ApplyImmediately` gilt nur für Änderungen am Knotentyp, an der Engine-Version und der Anzahl von Knoten in einem Cluster. Wenn Sie solche Änderungen umgehend übernehmen möchten, stellen Sie den Parameter `ApplyImmediately` auf `true` ein. Wenn Sie solche Änderungen lieber bis zum nächsten Wartungsfenster aufschieben möchten, stellen Sie den Parameter `ApplyImmediately` auf `false` ein. Andere Änderungen, wie z. B. Ändern des Wartungsfensters, werden umgehend übernommen.

Weitere Informationen finden Sie im ElastiCache API-Referenzthema. [ModifyCacheCluster](#)

Hinzufügen von Knoten zu einem ElastiCache Cluster

Das Hinzufügen von Knoten zu einem Memcached-Cluster erhöht die Anzahl Ihrer Cluster-Partitionen. Wenn Sie die Anzahl der Partitionen in einem Cluster ändern, müssen einige Ihrer Schlüsselbereiche neu zugeordnet werden, damit sie auf den richtigen Knoten abgebildet werden. Die Neuordnung von Schlüsselräumen erhöht vorübergehend die Anzahl der Cache-Fehler auf dem Cluster. Weitere Informationen finden Sie unter [Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich \(Memcached\)](#).

Informationen zur Neukonfiguration Ihres Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) finden Sie unter [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#)

Sie können die ElastiCache Management Console, die AWS CLI oder ElastiCache API verwenden, um Ihrem Cluster Knoten hinzuzufügen.

Mit dem ElastiCache AWS-Managementkonsole

Wenn Sie einen Knoten zu einem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) mit einem einzigen Knoten hinzufügen möchten (einem Cluster ohne aktivierte Replikation), müssen Sie zuerst die Replikation hinzufügen und dann einen Replikatknoten hinzufügen.

Topics

- [Um die Replikation zu einem Valkey- oder Redis OSS-Cluster ohne Shards hinzuzufügen](#)
- [Um Knoten zu einem ElastiCache Cluster \(Konsole\) hinzuzufügen](#)

Mit dem folgenden Verfahren wird die Replikation zu einem Valkey- oder Redis-OSS mit einem Knoten hinzugefügt, für den die Replikation nicht aktiviert ist. Wenn Sie Replikation hinzufügen, wird der vorhandene Knoten zum primären Knoten im replikationsfähigen Cluster. Nachdem die Replikation hinzugefügt wurde, können Sie dem Cluster bis zu 5 Replikationsknoten hinzufügen.

Um die Replikation zu einem Valkey- oder Redis OSS-Cluster ohne Shards hinzuzufügen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.

Eine Liste der Cluster, auf denen diese Engine ausgeführt wird, wird angezeigt.

3. Wählen Sie den Namen eines Clusters, nicht das Kästchen links vom Namen des Clusters, dem Sie Knoten hinzufügen möchten.

Folgendes gilt für einen Redis OSS-Cluster, für den die Replikation nicht aktiviert ist:

- Es läuft Redis OSS, nicht Clustered Redis OSS.
- Er besitzt keine Shards.

Wenn der Cluster über Shards verfügt, ist die Replikation für ihn bereits aktiviert und Sie können mit [Um Knoten zu einem ElastiCache Cluster \(Konsole\) hinzuzufügen](#) fortfahren.

4. Wählen Sie Add replication.
5. Geben Sie unter Add Replication eine Beschreibung für diesen replikationsfähigen Cluster ein.
6. Wählen Sie Hinzufügen aus.

Sobald der Status des Clusters wieder available lautet, können Sie mit dem nächsten Verfahren fortfahren und Replikate zum Cluster hinzufügen.

Um Knoten zu einem ElastiCache Cluster (Konsole) hinzuzufügen

Die folgende Vorgehensweise kann verwendet werden, um Knoten zu einem Cluster hinzuzufügen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, dem Sie Knoten hinzufügen möchten.

Es wird eine Liste der Cluster der ausgewählten Engine angezeigt.

3. Wählen Sie in der Liste der Cluster den Namen des Clusters aus, dem Sie einen Knoten hinzufügen möchten..

Wenn es sich bei Ihrem Cluster um einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) handelt, finden Sie weitere Informationen unter. [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#)

Wenn es sich bei Ihrem Cluster um einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) ohne Shards handelt, führen Sie zunächst die Schritte unter aus. [Um die Replikation zu einem Valkey- oder Redis OSS-Cluster ohne Shards hinzuzufügen](#)

4. Wählen Sie Knoten hinzufügen aus.

5. Geben Sie im Dialogfeld Add Node (Knoten hinzufügen) die erforderlichen Informationen ein.
6. Wählen Sie die Schaltfläche Sofort übernehmen – Ja, um diesen Knoten sofort hinzuzufügen, oder Nein, um das Hinzufügen dieses Knotens bis zum nächsten Wartungsfenster des Clusters aufzuschieben.

Auswirkung von neuen Hinzufügings- und Entfernungsanträgen auf ausstehende Anträge

Szenarien	Anstehende Operation	Neue Anfrage	Ergebnisse
Szenario 1	Delete	Delete	Der neue Löschantrag, ob anstehend oder sofort, ersetzt den anstehenden Löschantrag.. Wenn z. B. die Knoten 0001, 0003 und 0007 zur Löschung anstehen und ein neuer Löschantrag für die Knoten 0002 und 0004 gestellt wird, werden nur die Knoten 0002 und 0004 gelöscht. Die Knoten 0001, 0003 und 0007 werden nicht gelöscht.
Szenario 2	Delete	Create	Der neue Erstellungsauftrag, ob anstehend oder sofort, ersetzt den ausstehenden Löschauftrag. Wenn beispielsweise die Knoten 0001, 0003 und 0007 zur Löschung anstehen und ein neuer Antrag auf Erstellung eines Knotens gestellt wird, wird ein neuer Knoten erstellt und die Knoten 0001, 0003 und 0007 werden nicht gelöscht.
Szenario 3	Create	Delete	Der neue Löschantrag, ob anstehend oder sofort, ersetzt den ausstehenden Erstellungsantrag. Sind beispielsweise zwei Knoten anzulegen und wird ein neuer Antrag auf Löschung des Knotens 0003 gestellt, werden keine neuen Knoten angelegt und der Knoten 0003 wird gelöscht.
Szenario 4	Create	Create	Der neue Erstellungsauftrag wird dem ausstehenden Erstellungsauftrag hinzugefügt.

Szenarien	Anstehende Operation	Neue Anfrage	Ergebnisse
			Wenn z. B. eine ausstehende Anforderung zur Erstellung von zwei Knoten besteht und eine neue Anforderung zur Erstellung von drei Knoten ausgegeben wird, wird die neue Anforderung der ausstehenden Anforderung hinzugefügt und es werden fünf Knoten erstellt.  Important Wenn der neue Erstellungsauftrag auf Sofort anwenden – Ja eingestellt ist, werden alle Erstellungsaufträge sofort ausgeführt. Wenn der neue Erstellungsauftrag auf Sofort anwenden – Nein eingestellt ist, sind alle Erstellungsaufträge in der Schwebe.

Um festzustellen, welche Vorgänge noch nicht abgeschlossen sind, wählen Sie die Registerkarte Beschreibung und prüfen Sie, wie viele ausstehende Erstellungen oder Löschungen angezeigt werden. Sie können nicht sowohl ausstehende Erstellungen als auch ausstehende Löschungen haben.

7. Wählen Sie die Schaltfläche Add.

Nach kurzer Zeit sollten die neuen Knoten mit dem Status `creating` in der Knotenliste erscheinen. Wenn dies nicht der Fall ist, aktualisieren Sie die Browserseite. Sobald sich der Status des Knotens in `verfügbar` ändert, kann der neue Knoten verwendet werden.

Verwenden Sie den AWS CLI ElastiCache

Um einem Cluster mithilfe von Knoten hinzuzufügen AWS CLI, verwenden Sie den AWS CLI Vorgang `modify-cache-cluster` mit den folgenden Parametern:

- `--cache-cluster-id` Die ID des Clusters, zu dem Sie Knoten hinzufügen möchten.

- `--num-cache-nodes` Der Parameter `--num-cache-nodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `--num-cache-nodes` größer als die aktuelle Anzahl von Knoten in diesem Cluster sein. Wenn dieser Wert kleiner als die aktuelle Anzahl von Knoten ist, ElastiCache erwartet, dass der Parameter `cache-node-ids-to-remove` und eine Liste von Knoten aus dem Cluster entfernt werden. Weitere Informationen finden Sie unter [Verwenden AWS CLI Sie mit ElastiCache](#).
- `--apply-immediately` oder `--no-apply-immediately` gibt an, ob diese Knoten umgehend bzw. während des nächsten Wartungsfensters hinzugefügt werden sollen.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --num-cache-nodes 5 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --num-cache-nodes 5 ^  
  --apply-immediately
```

Dieser Vorgang erzeugt eine Ausgabe ähnlich der folgenden (JSON-Format):

```
{  
  "CacheCluster": {  
    "Engine": "memcached",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.memcached1.4",  
      "ParameterApplyStatus": "in-sync"  
    },  
    "CacheClusterId": "my-cluster",  
    "PreferredAvailabilityZone": "us-west-2b",  
    "ConfigurationEndpoint": {  
      "Port": 11211,  
      "Address": "rlh-mem000.7alc7bf-example.cfg.usw2.cache.amazonaws.com"  
    }  
  }  
}
```

```

    },
    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "NumCacheNodes": 2,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {
      "NumCacheNodes": 5
    },
    "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled",
  }
}

```

Weitere Informationen finden Sie im AWS CLI Thema [modify-cache-cluster](#).

Verwenden von AWS CLI with ElastiCache

Wenn Sie Knoten zu einem vorhandenen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) hinzufügen möchten, für den die Replikation nicht aktiviert ist, müssen Sie zuerst die Replikationsgruppe erstellen und dabei den vorhandenen Cluster als primären Cluster angeben. Weitere Informationen finden Sie unter [Erstellen einer Replikationsgruppe mithilfe eines verfügbaren Valkey- oder Redis OSS-Clusters \(\)AWS CLI](#). Wenn die Replikationsgruppe verfügbar ist, können Sie mit dem folgenden Vorgang fortfahren.

Um einem Cluster mithilfe von Knoten hinzuzufügen AWS CLI, verwenden Sie den AWS CLI Vorgang `increase-replica-count` mit den folgenden Parametern:

- `--replication-group-id` Die ID der Replikationsgruppe, aus der Sie Knoten entfernen möchten.

- `--new-replica-count` gibt die Anzahl der Knoten an, die nach Anwendung der Änderung in dieser Replikationsgruppe enthalten sein sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `--new-replica-count` größer als die aktuelle Anzahl von Knoten in diesem Cluster sein.
- `--apply-immediately` oder `--no-apply-immediately` gibt an, ob diese Knoten umgehend bzw. während des nächsten Wartungsfensters hinzugefügt werden sollen.

Für Linux, macOS oder Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 4 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 4 ^  
  --apply-immediately
```

Dieser Vorgang erzeugt eine Ausgabe ähnlich der folgenden (JSON-Format):

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "node-test",  
    "Description": "node-test",  
    "Status": "modifying",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "node-test-001",  
      "node-test-002",  
      "node-test-003",  
      "node-test-004",  
      "node-test-005"  
    ],  
    "NodeGroups": [  
      {  
        "NodeGroupId": "0001",  
        "Status": "modifying",  
        "PrimaryEndpoint": {  
          "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
```

```

        "Port": 6379
    },
    "ReaderEndpoint": {
        "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
        "Port": 6379
    },
    "NodeGroupMembers": [
        {
            "CacheClusterId": "node-test-001",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "primary"
        },
        {
            "CacheClusterId": "node-test-002",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2c",
            "CurrentRole": "replica"
        },
        {
            "CacheClusterId": "node-test-003",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2b",
            "CurrentRole": "replica"
        }
    ]
},
"SnapshottingClusterId": "node-test-002",

```

```
    "AutomaticFailover": "enabled",
    "MultiAZ": "enabled",
    "SnapshotRetentionLimit": 1,
    "SnapshotWindow": "07:30-08:30",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r5.large",
    "DataTiering": "disabled",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false,
    "ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-test"
  }
}
```

Weitere Informationen finden Sie im AWS CLI Thema [increase-replica-count](#).

Verwenden der ElastiCache API

Wenn Sie Knoten zu einem vorhandenen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) hinzufügen möchten, für den die Replikation nicht aktiviert ist, müssen Sie zuerst die Replikationsgruppe erstellen und den vorhandenen Cluster als primären Cluster angeben. Weitere Informationen finden Sie unter [Hinzufügen von Replikaten zu einem eigenständigen Valkey- oder Redis OSS-Cluster \(Cluster Mode Disabled\) \(API\) ElastiCache](#). Wenn die Replikationsgruppe verfügbar ist, können Sie mit dem folgenden Vorgang fortfahren.

Um Knoten zu einem Cluster (ElastiCache API) hinzuzufügen

- Rufen Sie die API-Operation `IncreaseReplicaCount` mit folgenden Parametern auf:
 - `ReplicationGroupId` Die ID des Clusters, zu dem Sie Knoten hinzufügen möchten.
 - `NewReplicaCount` Der Parameter `NewReplicaCount` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `NewReplicaCount` größer als die aktuelle Anzahl von Knoten in diesem Cluster sein. Wenn dieser Wert kleiner ist als die aktuelle Anzahl der Knoten, verwenden Sie die `DecreaseReplicaCount`-API mit der Anzahl der Knoten, die aus dem Cluster entfernt werden sollen.
 - `ApplyImmediately` Gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters hinzugefügt werden sollen.
 - `Region` Gibt die AWS Region des Clusters an, zu der Sie Knoten hinzufügen möchten.

Das folgende Beispiel zeigt einen Aufruf zum Hinzufügen von Knoten zu einem Cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=IncreaseReplicaCount  
  &ApplyImmediately=true  
  &NumCacheNodes=4  
  &ReplicationGroupId=my-replication-group  
  &Region=us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie unter ElastiCache API-Thema [IncreaseReplicaCount](#).

Die ElastiCache API verwenden

Um Knoten zu einem Cluster hinzuzufügen (ElastiCache API)

- Rufen Sie die API-Operation `ModifyCacheCluster` mit folgenden Parametern auf:
 - `CacheClusterId` Die ID des Clusters, zu dem Sie Knoten hinzufügen möchten.
 - `NumCacheNodes` Der Parameter `NumCacheNodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `NumCacheNodes` größer als die aktuelle Anzahl von Knoten in diesem Cluster sein. Wenn dieser Wert unter der aktuellen Anzahl von Knoten liegt, wird ElastiCache erwartet, dass der Parameter `CacheNodeIdsToRemove` mit einer Liste von Knoten aus dem Cluster entfernt wird (siehe [Verwenden der ElastiCache API mit Memcached](#)).
 - `ApplyImmediately` Gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters hinzugefügt werden sollen.

- **Region**Gibt die AWS Region des Clusters an, zu der Sie Knoten hinzufügen möchten.

Das folgende Beispiel zeigt einen Aufruf zum Hinzufügen von Knoten zu einem Cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=true  
  &NumCacheNodes=5  
&CacheClusterId=my-cluster  
&Region=us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie unter ElastiCache API-Thema [ModifyCacheCluster](#).

Knoten aus einem ElastiCache Cluster entfernen

Sie können einen Knoten aus einem Valkey-, Memcached- oder Redis-OSS-Cluster löschen, indem Sie die AWS-Managementkonsole, oder die API AWS CLI verwenden. ElastiCache

Note

Jedes Mal, wenn Sie die Anzahl der Knoten in einem Memcached-Cluster ändern, müssen Sie zumindest einen Teil Ihres Keyspace neu zuordnen, sodass er dem richtigen Knoten zugeordnet ist. Detaillierte Informationen zum Lastausgleich eines Memcached-Clusters finden Sie unter [Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich \(Memcached\)](#).

Mit dem ElastiCache AWS-Managementkonsole

Entfernen von Knoten aus einem Cluster (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region des Clusters aus, aus dem Sie Knoten entfernen möchten.
3. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, von dem Sie einen Knoten entfernen möchten.

Es wird eine Liste der Cluster der ausgewählten Engine angezeigt.

4. Wählen Sie aus der Liste der Cluster den Namen des Clusters aus, aus dem Sie einen Knoten entfernen möchten.

Eine Liste der Knoten des Clusters wird angezeigt.

5. Aktivieren Sie das Kontrollkästchen links neben der Knoten-ID des Knotens, den Sie entfernen möchten. Sie können mithilfe der ElastiCache -Konsole nur jeweils einen Knoten auswählen. Wenn Sie mehrere Knoten auswählen, können Sie die Schaltfläche Delete node (Knoten löschen) daher nicht verwenden.

Die Seite Delete Node (Knoten löschen) wird angezeigt.

6. Um den Knoten zu löschen, machen Sie alle Angaben auf der Seite Delete Node und wählen Sie Delete Node. Um den Knoten beizubehalten, wählen Sie Cancel (Abbrechen).

⚠ Important

Wenn Sie mit Valkey oder Redis OSS die Knotenergebnisse im Cluster löschen, die nicht mehr Multi-AZ-konform sind, stellen Sie sicher, dass Sie zuerst das Multi-AZ-Kontrollkästchen deaktivieren und dann den Knoten löschen. Wenn Sie das Kontrollkästchen Multi-AZ deaktivieren, können Sie Auto failover (Automatisches Failover) aktivieren.

Auswirkung von neuen Hinzufügungs- und Entfernungsanträgen auf ausstehende Anträge

Szenarien	Anstehende Operation	Neue Anfrage	Ergebnisse
Szenario 1	Delete	Delete	Der neue Löschantrag, ob anstehend oder sofort, ersetzt den anstehenden Löschantrag.. Wenn z. B. die Knoten 0001, 0003 und 0007 zur Löschung anstehen und ein neuer Löschantrag für die Knoten 0002 und 0004 gestellt wird, werden nur die Knoten 0002 und 0004 gelöscht. Die Knoten 0001, 0003 und 0007 werden nicht gelöscht.
Szenario 2	Delete	Create	Der neue Erstellungsauftrag, ob anstehend oder sofort, ersetzt den ausstehenden Löschauftrag. Wenn beispielsweise die Knoten 0001, 0003 und 0007 zur Löschung anstehen und ein neuer Antrag auf Erstellung eines Knotens gestellt wird, wird ein neuer Knoten erstellt und die Knoten 0001, 0003 und 0007 werden nicht gelöscht.
Szenario 3	Create	Delete	Der neue Löschantrag, ob anstehend oder sofort, ersetzt den ausstehenden Erstellungsantrag. Sind beispielsweise zwei Knoten anzulegen und wird ein neuer Antrag auf Löschung des Knotens 0003

Szenarien	Anstehende Operation	Neue Anfrage	Ergebnisse
			gestellt, werden keine neuen Knoten angelegt und der Knoten 0003 wird gelöscht.
Szenario 4	Create	Create	Der neue Erstellungsauftrag wird dem ausstehenden Erstellungsauftrag hinzugefügt. Wenn z. B. eine ausstehende Anforderung zur Erstellung von zwei Knoten besteht und eine neue Anforderung zur Erstellung von drei Knoten ausgegeben wird, wird die neue Anforderung der ausstehenden Anforderung hinzugefügt und es werden fünf Knoten erstellt.  Important Wenn der neue Erstellungsauftrag auf Sofort anwenden – Ja eingestellt ist, werden alle Erstellungsaufträge sofort ausgeführt. Wenn der neue Erstellungsauftrag auf Sofort anwenden – Nein eingestellt ist, sind alle Erstellungsaufträge in der Schwebe.

Um festzustellen, welche Vorgänge noch nicht abgeschlossen sind, wählen Sie die Registerkarte Beschreibung und prüfen Sie, wie viele ausstehende Erstellungen oder Löschungen angezeigt werden. Sie können nicht sowohl ausstehende Erstellungen als auch ausstehende Löschungen haben.

Verwenden AWS CLI Sie mit ElastiCache

1. Identifizieren Sie IDs den Knoten, den Sie entfernen möchten. Weitere Informationen finden Sie unter [Details eines ElastiCache Clusters anzeigen](#).
2. Listen Sie mit der CLI-Operation `decrease-replica-count` wie im folgenden Beispiel die zu entfernenden Knoten auf.

Um über die Befehlszeilenschnittstelle Knoten aus einem Cluster zu entfernen, verwenden Sie den Befehl `decrease-replica-count` mit den folgenden Parametern:

- `--replication-group-id` Die ID der Replikationsgruppe, aus der Sie Knoten entfernen möchten.
- `--new-replica-count` Der Parameter `--new-replica-count` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen.
- `--replicas-to-remove` Eine Liste der Knoten IDs , die Sie aus diesem Cluster entfernen möchten.
- `--apply-immediately` oder `--no-apply-immediately` gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters entfernt werden sollen.
- `--region` Gibt die AWS Region des Clusters an, aus dem Sie Knoten entfernen möchten.

 Note

Sie können nur einen von `--replicas-to-remove` oder `--new-replica-count` Parametern übergeben, wenn Sie diesen Vorgang aufrufen.

Für Linux, macOS oder Unix:

```
aws elasticache decrease-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 2 \  
  --region us-east-2 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache decrease-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 3 ^  
  --region us-east-2 ^  
  --apply-immediately
```

Dieser Vorgang erzeugt eine Ausgabe ähnlich der folgenden (JSON-Format):

```

{
  "ReplicationGroup": {
    "ReplicationGroupId": "node-test",
    "Description": "node-test"
  },
  "Status": "modifying",
  "PendingModifiedValues": {},
  "MemberClusters": [
    "node-test-001",
    "node-test-002",
    "node-test-003",
    "node-test-004",
    "node-test-005",
    "node-test-006"
  ],
  "NodeGroups": [
    {
      "NodeGroupId": "0001",
      "Status": "modifying",
      "PrimaryEndpoint": {
        "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
        "Port": 6379
      },
      "ReaderEndpoint": {
        "Address": "node-test-
ro.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
        "Port": 6379
      },
      "NodeGroupMembers": [
        {
          "CacheClusterId": "node-test-001",
          "CacheNodeId": "0001",
          "ReadEndpoint": {
            "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
          },
          "PreferredAvailabilityZone": "us-west-2a",
          "CurrentRole": "primary"
        },
        {
          "CacheClusterId": "node-test-002",
          "CacheNodeId": "0001",

```

```
        "ReadEndpoint": {
            "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2c",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-003",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-004",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-004.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2c",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-005",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-005.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-006",
        "CacheNodeId": "0001",
```

```

        "ReadEndpoint": {
            "Address": "node-
test-006.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    }
]
}
],
"SnapshottingClusterId": "node-test-002",
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.large",
"DataTiering": "disabled",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false,
"ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-
test"
}
}

```

Alternativ können Sie auch `decrease-replica-count` aufrufen und statt des Parameters `--new-replica-count` den Parameter `--replicas-to-remove` übergeben, wie im Folgenden gezeigt:

Für Linux, macOS oder Unix:

```

aws elasticache decrease-replica-count \
  --replication-group-id my-replication-group \
  --replicas-to-remove node-test-003 \
  --region us-east-2 \
  --apply-immediately

```

Für Windows:

```

aws elasticache decrease-replica-count ^
  --replication-group-id my-replication-group ^
  --replicas-to-remove node-test-003 ^

```

```
--region us-east-2 ^  
--apply-immediately
```

Weitere Informationen finden Sie in den AWS CLI Themen [decrease-replica-count](#).

Verwenden der ElastiCache API mit Valkey oder Redis OSS

Um Knoten mithilfe der ElastiCache API zu entfernen, rufen Sie den DecreaseReplicaCount API-Vorgang mit der Replikationsgruppen-ID und einer Liste der zu entfernenden Knoten auf, wie hier gezeigt:

- **ReplicationGroupId** Die ID der Replikationsgruppe, aus der Sie Knoten entfernen möchten.
- **ReplicasToRemove** Der Parameter ReplicasToRemove gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen.
- **ApplyImmediately** gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters entfernt werden sollen.
- **Region** Gibt die AWS Region des Clusters an, aus der Sie einen Knoten entfernen möchten.

Das folgende Beispiel entfernt umgehend Knoten 0004 und 0005 aus dem Cluster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DecreaseReplicaCount  
&ReplicationGroupId=my-replication-group  
&ApplyImmediately=true  
&ReplicasToRemove=node-test-003  
&Region us-east-2  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie unter ElastiCache API-Thema [DecreaseReplicaCount](#).

Verwenden der ElastiCache API mit Memcached

Um Knoten mithilfe der ElastiCache API zu entfernen, rufen Sie den `ModifyCacheCluster` API-Vorgang mit der Cache-Cluster-ID und einer Liste der zu entfernenden Knoten auf, wie hier gezeigt:

- `CacheClusterId` Die ID des Clusters, aus dem Sie Knoten entfernen möchten.
- `NumCacheNodes` Der Parameter `NumCacheNodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen.
- `CacheNodeIdsToRemove.member.n` Die Liste der Knoten IDs , die aus dem Cluster entfernt werden sollen.
 - `CacheNodeIdsToRemove.member.1=0004`
 - `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately` gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters entfernt werden sollen.
- `Region` Gibt die AWS Region des Clusters an, aus dem Sie einen Knoten entfernen möchten.

Das folgende Beispiel entfernt umgehend Knoten 0004 und 0005 aus dem Cluster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&CacheClusterId=my-cluster  
&ApplyImmediately=true  
&CacheNodeIdsToRemove.member.1=0004  
&CacheNodeIdsToRemove.member.2=0005  
&NumCacheNodes=3  
&Region us-east-2  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie unter ElastiCache API-Thema [ModifyCacheCluster](#).

Abbrechen ausstehender Operationen zum Hinzufügen oder Löschen von Knoten in ElastiCache

Wenn Sie sich dafür entschieden haben, eine ElastiCache Clusteränderung nicht sofort anzuwenden, hat der Vorgang den Status Ausstehend, bis er in Ihrem nächsten Wartungsfenster ausgeführt wird. Sie können jeden ausstehenden Vorgang zum Hinzufügen oder Löschen abbrechen.

Verwenden Sie den `modify-cache-cluster` Befehl, um einen ausstehenden Vorgang zum Hinzufügen oder Löschen von Knoten mit dem AWS CLI abzubrechen. Stellen Sie `num-cache-nodes` den Wert auf die aktuelle Anzahl von Cache-Knoten im Cluster ein und fügen Sie dann das `--apply-immediately` Flag hinzu. Dadurch wird die ausstehende Änderung außer Kraft gesetzt.

So stornieren Sie das Hinzufügen oder Löschen eines ausstehenden Knotens:

```
aws elasticache modify-cache-cluster
--cache-cluster-id <your-cluster-id>
--num-cache-nodes <current-number-of-nodes>
--apply-immediately
```

Wenn nicht klar ist, ob das Hinzufügen oder Löschen von Knoten aussteht, können Sie deren Status mit dem `describe-cache-clusters` folgenden Befehl überprüfen:

```
aws elasticache describe-cache-clusters
--cache-cluster-id <your-cluster-id>
```

Alle ausstehenden Knoten sollten in der `PendingModifiedValues` Ausgabe erscheinen. Zum Beispiel:

```
"PendingModifiedValues": {
  "NumCacheNodes": 3
},
```

Löschen eines Clusters in ElastiCache

Solange sich ein ElastiCache Cluster im Status „Verfügbar“ befindet, wird er Ihnen in Rechnung gestellt, unabhängig davon, ob Sie ihn aktiv nutzen oder nicht. Löschen Sie den Cluster, damit keine Gebühren mehr anfallen.

Warning

Wenn Sie einen ElastiCache Cluster löschen, werden Ihre manuellen Snapshots beibehalten. Sie können auch einen letzten Snapshot erstellen, bevor der Cluster gelöscht wird. Automatisch erstellte Cache-Snapshots werden nicht aufbewahrt.

Mit dem AWS-Managementkonsole

Mit dem folgenden Verfahren wird ein einzelner Cluster aus Ihrer Bereitstellung gelöscht. Um mehrere Cache-Cluster zu löschen, wiederholen Sie das Verfahren für jeden Cluster, den Sie löschen möchten. Sie brauchen nicht zu warten, bis ein Cluster fertig gelöscht ist, bevor Sie den Vorgang zum Löschen eines anderen Clusters starten.

So löschen Sie einen Cluster

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Engine-Dashboard die Engine aus, die in dem Cluster läuft, den Sie löschen möchten.

Eine Liste mit allen Clustern, in denen diese Engine läuft, wird angezeigt.

3. Wählen Sie aus der Liste den Namen des zu löschenden Clusters aus.

Important

Sie können jeweils nur einen Cluster von der ElastiCache Konsole aus löschen. Werden mehrere Cluster ausgewählt, wird die Löschoperation deaktiviert.

4. Klicken Sie bei Actions auf Delete.
5. Wählen Sie auf dem Bestätigungsbildschirm Delete Cluster (Cluster löschen) die Option Delete (Löschen), um den Cluster zu löschen, oder Cancel (Abbrechen), um den Cluster beizubehalten.

Wenn Sie Delete auswählen, ändert sich der Status des Clusters zu deleting.

Sobald dieser Cluster nicht mehr in der Cluster-Liste erscheint, fallen dafür keine Gebühren mehr an.

Verwenden Sie den AWS CLI, um einen ElastiCache Cluster zu löschen

Der folgende Code löscht den ElastiCache Cluster `my-cluster`.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

Die `delete-cache-cluster` CLI-Aktion löscht nur einen Cluster. Um mehrere Cluster zu löschen, rufen Sie `delete-cache-cluster` jeden Cache-Cluster auf, den Sie löschen möchten. Sie müssen nicht warten, bis ein Cluster vollständig gelöscht ist, bevor Sie einen anderen löschen.

Für Linux, macOS oder Unix:

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --region us-east-2
```

Für Windows:

```
aws elasticache delete-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --region us-east-2
```

Weitere Informationen finden Sie im AWS CLI ElastiCache For-Thema [delete-cache-cluster](#).

Verwenden der ElastiCache API

Der folgende Code löscht den Cluster `my-cluster`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DeleteCacheCluster  
&CacheClusterId=my-cluster  
&Region us-east-2  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
```

```
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150202T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Der `DeleteCacheCluster` API-Vorgang löscht nur einen Cluster. Um mehrere Cluster zu löschen, rufen Sie `DeleteCacheCluster` für jeden Cluster auf, den Sie löschen möchten. Sie müssen nicht warten, bis ein Cluster vollständig gelöscht ist, bevor Sie einen anderen löschen.

Weitere Informationen finden Sie im ElastiCache API-Referenzthema [DeleteCacheCluster](#).

Zugreifen auf Ihren ElastiCache Cluster oder Ihre Replikationsgruppe

Ihre ElastiCache Amazon-Instances sind für den Zugriff über eine EC2 Amazon-Instance konzipiert.

Wenn Sie Ihre ElastiCache Instance in einer Amazon Virtual Private Cloud (Amazon VPC) gestartet haben, können Sie von einer ElastiCache Amazon-Instance in derselben Amazon VPC aus auf Ihre EC2 Instance zugreifen. Oder Sie können mithilfe von VPC-Peering von einem Amazon EC2 in einer anderen Amazon-VPC auf Ihre ElastiCache Instance zugreifen.

Wenn Sie Ihre ElastiCache Instance in EC2 Classic gestartet haben, erlauben Sie der EC2 Instance den Zugriff auf Ihren Cluster, indem Sie der mit der Instance verknüpften EC2 Amazon-Sicherheitsgruppe Zugriff auf Ihre Cache-Sicherheitsgruppe gewähren. Standardmäßig ist der Zugriff auf einen Cluster auf das Konto beschränkt, mit dem der Cluster gestartet wurde.

Themen

- [Zugriff auf Ihren Cluster oder die Replikationsgruppe erteilen](#)

Zugriff auf Ihren Cluster oder die Replikationsgruppe erteilen

Sie haben Ihren Cluster in VPC EC2 gestartet

Wenn Sie Ihren Cluster in einer Amazon Virtual Private Cloud (Amazon VPC) gestartet haben, können Sie nur von einer EC2 Amazon-Instance aus, die in derselben Amazon VPC läuft, eine Verbindung zu Ihrem ElastiCache Cluster herstellen. In diesem Fall müssen Sie Netzwerkzugang zum Cluster gewähren.

Note

Wenn Sie Local Zones verwenden, vergewissern Sie sich, dass Sie sie aktiviert haben. Weitere Informationen finden Sie unter [Local Zones aktivieren](#). Dadurch wird Ihre VPC auf diese Local Zone ausgedehnt, und Ihre VPC behandelt das Subnetz wie jedes andere Subnetz in jeder anderen Availability Zone, und relevante Gateways, Routentabellen und andere Sicherheitsgruppen werden automatisch angepasst.

So gewähren Sie einem Cluster den Netzwerkeingang aus einer Amazon-VPC-Sicherheitsgruppe

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.

2. Wählen Sie in der Navigationsleiste unter Network & Security die Option Security Groups aus.
3. Wählen Sie aus der Liste der Sicherheitsgruppen die Sicherheitsgruppe Ihrer Amazon VPC aus. Sofern Sie keine Sicherheitsgruppe zur ElastiCache Verwendung erstellt haben, wird diese Sicherheitsgruppe als Standard bezeichnet.
4. Wählen Sie die Registerkarte Inbound und verfahren Sie dann wie folgt:
 - a. Wählen Sie Bearbeiten aus.
 - b. Wählen Sie Regel hinzufügen aus.
 - c. Wählen Sie in der Spalte Typ die Option Benutzerdefinierte TCP-Regel aus.
 - d. Geben Sie in das Feld Port Range die Portnummer Ihres Clusterknotens ein. Diese Nummer muss mit der Nummer übereinstimmen, die Sie beim Starten des Clusters angegeben haben. Der Standardport für Memcached ist **11211** Der Standardport für Valkey und Redis OSS ist. **6379**
 - e. Wählen Sie im Feld Quelle die Option Anywhere mit dem Portbereich (0.0.0.0/0) aus, sodass jede EC2 Amazon-Instance, die Sie in Ihrer Amazon VPC starten, eine Verbindung zu Ihren Knoten herstellen kann. ElastiCache

 Important

Durch das Öffnen des ElastiCache Clusters für 0.0.0.0/0 wird der Cluster nicht dem Internet zugänglich gemacht, da er keine öffentliche IP-Adresse hat und daher nicht von außerhalb der VPC darauf zugegriffen werden kann. Die Standardsicherheitsgruppe kann jedoch auf andere EC2 Amazon-Instances im Kundenkonto angewendet werden, und diese Instances können eine öffentliche IP-Adresse haben. Wenn diese Instances zufälligerweise eine Aktion auf dem Standardport ausführen, dann könnte dieser Service unbeabsichtigt zugänglich gemacht werden. Daher wird empfohlen, eine VPC-Sicherheitsgruppe einzurichten, die ausschließlich von ElastiCache genutzt wird. Weitere Informationen finden Sie unter [Benutzerdefinierte Sicherheitsgruppen](#).

- f. Wählen Sie Speichern.

Wenn Sie eine EC2 Amazon-Instance in Ihrer Amazon VPC starten, kann diese Instance eine Verbindung zu Ihrem ElastiCache Cluster herstellen.

Zugriff auf ElastiCache Ressourcen von außen AWS

Amazon ElastiCache ist ein AWS Service, der einen cloudbasierten In-Memory-Key-Value-Speicher bereitstellt. Der Service ist so konzipiert, dass er ausschließlich von innen abgerufen werden kann. AWS Wenn der ElastiCache Cluster jedoch in einer VPC gehostet wird, können Sie eine Network Address Translation (NAT) -Instance verwenden, um externen Zugriff zu gewähren.

Voraussetzungen

Die folgenden Anforderungen müssen erfüllt sein, damit Sie von außerhalb AWS auf Ihre ElastiCache Ressourcen zugreifen können:

- Der Cluster muss sich innerhalb einer VPC befinden und über eine Network Address Translation (NAT)-Instance zugänglich sein. Für diese Anforderung gibt es keine Ausnahmen.
- Die NAT-Instance muss in der gleichen VPC gestartet werden wie der Cluster.
- Die NAT-Instance muss in einem vom Cluster getrennten, öffentlichen Subnetz gestartet werden.
- Eine Elastic IP-Adresse (EIP) muss der NAT-Instance zugeordnet sein. Die Portweiterleitungsfunktion von iptables wird verwendet, um einen Port auf der NAT-Instance an den Cache-Knotenport innerhalb der VPC weiterzuleiten.

Überlegungen

Beachten Sie beim Zugriff auf Ihre ElastiCache -Ressourcen von außerhalb von ElastiCache die folgenden Punkte.

- Clients stellen eine Verbindung mit der EIP und dem Cache-Port der NAT-Instance her. Die Portweiterleitung auf der NAT-Instance leitet den Datenverkehr an den entsprechenden Clusterknoten weiter.
- Wenn ein Cluster-Knoten hinzugefügt oder ersetzt wird, müssen die iptables-Regeln entsprechend aktualisiert werden.

Einschränkungen

Dieser Ansatz sollte nur zu Test- und Entwicklungszwecken verwendet werden. Er wird aufgrund der folgenden Einschränkungen nicht für die Produktion empfohlen:

- Die NAT-Instance fungiert als Proxy zwischen Clients und mehreren Clustern. Das Hinzufügen eines Proxys wirkt sich auf die Leistung des Clusters aus. Die Auswirkung nimmt mit der Anzahl der Cluster zu, auf die Sie über die NAT-Instance zugreifen.
- Der Datenverkehr von den Clients zur NAT-Instance ist nicht verschlüsselt. Daher sollten Sie keine sensiblen Daten über die NAT-Instance senden.
- Die NAT-Instance bedeutet zusätzlichen Aufwand durch Verwalten einer weiteren Instance.
- Die NAT-Instance dient als einzelne Fehlerquelle. Informationen zum Einrichten von NAT mit hoher Verfügbarkeit in VPC finden Sie unter [Hohe Verfügbarkeit für Amazon VPC NAT-Instances: Beispiel](#).

Wie greife ich von außen auf ElastiCache Ressourcen zu AWS

Das folgende Verfahren zeigt, wie Sie mithilfe einer NAT-Instanz eine Verbindung zu Ihren ElastiCache Ressourcen herstellen.

Diese Schritte setzen Folgendes voraus:

- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to 10.0.1.231:6379`
- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379`

Als Nächstes benötigen Sie NAT in die entgegengesetzte Richtung:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

Sie müssen auch die IP-Weiterleitung aktivieren, die standardmäßig deaktiviert ist:

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/sysctl.conf sudo sysctl --system
```

- Ihr Zugriff auf einen Memcached-Cluster erfolgt mit:
 - IP-Adresse – 10.0.1.230
 - Memcached-Standardport – 11211
 - Sicherheitsgruppe – *10\0\0\55*
- Sie greifen auf einen Valkey- oder Redis OSS-Cluster zu mit:

- IP-Adresse – 10.0.1.230
- Standardport — 6379
- Sicherheitsgruppe – sg-bd56b7da
- AWSIP-Adresse der Instanz — 198.99.100.27
- Die IP-Adresse Ihres vertrauenswürdigen Clients lautet 198.51.100.27.
- Ihre NAT-Instance hat die Elastic IP-Adresse 203.0.113.73.
- Die Sicherheitsgruppe Ihrer NAT-Instance lautet sg-ce56b7a9.

So stellen Sie mithilfe einer NAT-Instanz eine Verbindung zu Ihren ElastiCache Ressourcen her

1. Erstellen Sie eine NAT-Instanz in derselben VPC wie Ihr Cluster, jedoch in einem öffentlichen Subnetz.

Standardmäßig startet der VPC-Assistent einen `cache.m1.small`-Knotentyp. Wählen Sie eine Knotengröße aus, die Ihren Anforderungen entspricht. Sie müssen EC2 NAT AMI verwenden, um ElastiCache von außen darauf zugreifen zu können AWS.

Informationen zum Erstellen einer NAT-Instance finden Sie unter [NAT-Instances](#) im AWS VPC-Benutzerhandbuch.

2. Erstellen Sie Sicherheitsgruppenregeln für den Cluster und die NAT-Instance.

Die Sicherheitsgruppe der NAT-Instance und die Cluster-Instance sollten folgende Regeln besitzen:

- Zwei Regeln für eingehenden Datenverkehr
 - Bei Memcached besteht die erste Regel darin, TCP-Verbindungen von vertrauenswürdigen Clients zu jedem Cache-Port zuzulassen, der von der NAT-Instanz weitergeleitet wird (11211 — 11213).
 - Bei Valkey und Redis OSS besteht die erste Regel darin, TCP-Verbindungen von vertrauenswürdigen Clients zu jedem Cache-Port zuzulassen, der von der NAT-Instanz weitergeleitet wird (6379 — 6381).
- Eine zweite Regel ermöglicht den SSH-Zugriff auf vertrauenswürdige Clients.

Sicherheitsgruppe für NAT-Instanzen — Regeln für eingehenden Datenverkehr mit Memcached

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
Zielbereich	TCP	11211 –11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

NAT-Instanz-Sicherheitsgruppe — Regeln für eingehenden Datenverkehr mit Valkey oder Redis OSS

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
Zielbereich	TCP	6379-6380	198.51.100.27/32
SSH	TCP	22	203.0.113.73/32

- Bei Memcached handelt es sich um eine ausgehende Regel, die TCP-Verbindungen zum Cache-Port (11211) zulässt.

Sicherheitsgruppe der NAT-Instance – Regeln für ausgehenden Datenverkehr

Typ	Protocol (Protokoll)	Port-Bereich	Ziel
Zielbereich	TCP	11211	sg-ce56b7a9 (NAT-Sicherheitsgruppe)

- Mit Valkey oder Redis OSS, einer ausgehenden Regel, die TCP-Verbindungen zum Cache-Port (6379) zulässt.

Sicherheitsgruppe der NAT-Instance – Regeln für ausgehenden Datenverkehr

Typ	Protocol (Protokoll)	Port-Bereich	Ziel
Zielbereich	TCP	6379	sg-ce56b7a9 (NAT-Sicherheitsgruppe)

- Mit Memcached, einer Regel für eingehende Nachrichten für die Sicherheitsgruppe des Clusters, die TCP-Verbindungen von der NAT-Instance zum Cache-Port (11211) zulässt.

Sicherheitsgruppe der Cluster-Instance – Regel für eingehenden Datenverkehr

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
Zielbereich	TCP	11211	sg-ce56b7a9 (NAT-Sicherheitsgruppe)

- Mit Valkey oder Redis OSS, einer eingehenden Regel für die Sicherheitsgruppe des Clusters, die TCP-Verbindungen von der NAT-Instance zum Cache-Port (6379) zulässt.

Sicherheitsgruppe der Cluster-Instance – Regel für eingehenden Datenverkehr

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
Zielbereich	TCP	6379	sg-ce56b7a9 (NAT-Sicherheitsgruppe)

3. Validieren Sie die Regeln.

- Bestätigen Sie, dass der vertrauenswürdige Client eine SSH-Verbindung mit der NAT-Instance herstellen kann.
- Bestätigen Sie, dass der vertrauenswürdige Client eine Verbindung mit dem Cluster über die NAT-Instance herstellen kann.

4. Memcached

Fügen Sie der NAT-Instance eine iptables-Regel hinzu.

Eine iptables-Regel muss der NAT-Tabelle für jeden Knoten im Cluster hinzugefügt werden, um den Cache-Port von der NAT-Instance an den Cluster-Knoten weiterzuleiten. Ein Beispiel könnte folgendermaßen aussehen:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to 10.0.1.230:11211
```

Die Portnummer muss für jeden Knoten im Cluster eindeutig sein. Wenn Sie z. B. mit einem aus drei Knoten bestehenden Memcached-Cluster unter Verwendung der Ports 11211 – 11213 arbeiten, sehen die Regeln wie folgt aus:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to  
10.0.1.230:11211  
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11212 -j DNAT --to  
10.0.1.231:11211  
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11213 -j DNAT --to  
10.0.1.232:11211
```

Bestätigen Sie, dass der vertrauenswürdige Client eine Verbindung mit dem Cluster herstellen kann.

Der vertrauenswürdige Client sollte eine Verbindung mit dem EIP herstellen, der der NAT-Instance zugewiesen ist, und dem Cluster-Port, der dem entsprechenden Cluster-Knoten entspricht. Die Verbindungszeichenfolge für PHP sieht beispielsweise wie folgt aus:

```
$memcached->connect( '203.0.113.73', 11211 );  
$memcached->connect( '203.0.113.73', 11212 );  
$memcached->connect( '203.0.113.73', 11213 );
```

Zum Überprüfen der Verbindung kann auch ein Telnet-Client verwendet werden. Beispiel:

```
telnet 203.0.113.73 11211  
telnet 203.0.113.73 11212  
telnet 203.0.113.73 11213
```

Valkey oder Redis OSS

Fügen Sie der NAT-Instance eine iptables-Regel hinzu.

Eine iptables-Regel muss der NAT-Tabelle für jeden Knoten im Cluster hinzugefügt werden, um den Cache-Port von der NAT-Instance an den Cluster-Knoten weiterzuleiten. Ein Beispiel könnte folgendermaßen aussehen:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to  
10.0.1.230:6379
```

Die Portnummer muss für jeden Knoten im Cluster eindeutig sein. Wenn Sie beispielsweise mit einem Redis-OSS-Cluster mit drei Knoten arbeiten und die Ports 6379 — 6381 verwenden, würden die Regeln wie folgt aussehen:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to 10.0.1.230:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to 10.0.1.231:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379
```

Bestätigen Sie, dass der vertrauenswürdige Client eine Verbindung mit dem Cluster herstellen kann.

Der vertrauenswürdige Client sollte eine Verbindung mit dem EIP herstellen, der der NAT-Instance zugewiesen ist, und dem Cluster-Port, der dem entsprechenden Cluster-Knoten entspricht. Die Verbindungszeichenfolge für PHP sieht beispielsweise wie folgt aus:

```
redis->connect( '203.0.113.73', 6379 );
redis->connect( '203.0.113.73', 6380 );
redis->connect( '203.0.113.73', 6381 );
```

Zum Überprüfen der Verbindung kann auch ein Telnet-Client verwendet werden. Beispiel:

```
telnet 203.0.113.73 6379
telnet 203.0.113.73 6380
telnet 203.0.113.73 6381
```

5. Speichern Sie die iptables-Konfiguration.

Speichern Sie die Regeln, nachdem Sie sie getestet und überprüft haben. Wenn Sie eine Redhat-Linux-Verteilung (wie Amazon Linux) verwenden, führen Sie den folgenden Befehl aus:

```
service iptables save
```

Verwandte Themen

Die folgenden Themen können von zusätzlichem Interesse sein.

- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#)
- [Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden ausgeführt wird](#)

- [NAT-Instances](#)
- [Konfiguration von Clients ElastiCache](#)
- [Hohe Verfügbarkeit für Amazon VPC NAT-Instances: Beispiel](#)

Verbindungsendpunkte finden in ElastiCache

Ihre Anwendung stellt über Endpunkte eine Verbindung zu Ihrem ElastiCache Cluster her. Ein Endpunkt ist die eindeutige Adresse eines Knotens oder Clusters.

Sie können auch eine private Verbindung zwischen Ihrer VPC und Ihren ElastiCache API-Endpunkten herstellen, indem Sie einen VPC-Schnittstellen-Endpoint erstellen. AWS PrivateLink Weitere Informationen finden Sie unter [ElastiCache API- und Schnittstellen-VPC-Endpunkte \(\)AWS PrivateLink](#).

Welche Endpunkte sollen mit Valkey oder Redis OSS verwendet werden?

- Verwenden Sie bei einem eigenständigen Knoten den Endpunkt des Knotens sowohl für Lese- als auch für Schreibvorgänge.
- Verwenden Sie für Valkey- oder Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) den primären Endpunkt für alle Schreibvorgänge. Verwenden Sie Leser-Endpoint, um am Endpunkt ankommende Verbindungen auf alle Lesereplikate zu verteilen. Verwenden Sie die einzelnen Knotenendpunkte für Lesevorgänge (in der API/CLI werden diese als Leseendpunkte bezeichnet).
- Verwenden Sie für Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) den Konfigurationsendpunkt des Clusters für alle Operationen, die Befehle mit aktiviertem Clustermodus unterstützen. Sie müssen einen Client verwenden, der entweder Valkey Cluster oder Redis OSS Cluster auf Redis OSS 3.2 und höher unterstützt. Sie können weiterhin von einzelnen Knotenendpunkten lesen (im Folgenden werden API/CLI diese als Lese-Endpunkte bezeichnet).

Die folgenden Abschnitte führen Sie durch die Erkennung der Endpunkte, die Sie für die von Ihnen ausgeführte Engine benötigen.

Welche Endpunkte sollen mit Memcached verwendet werden?

Für ElastiCache serverlosen Cache für Memcached rufen Sie einfach den DNS und den Port des Cluster-Endpunkts von der Konsole ab.

Verwenden Sie von der aus den `describe-serverless-caches` Befehl AWS CLI, um die Endpunktinformationen abzurufen.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Die Ausgabe der obigen Operation sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format):

```
{
  "ServerlessCaches": [
    {
      "ServerlessCacheName": "serverless-memcached",
      "Description": "test",
      "CreateTime": 1697659642.136,
      "Status": "available",
      "Engine": "memcached",
      "MajorEngineVersion": "1.6",
      "FullEngineVersion": "21",
      "SecurityGroupIds": [
        "sg-083eda453e1e51310"
      ],
      "Endpoint": {
        "Address": "serverless-memcached-01.amazonaws.com",
        "Port": 11211
      },
      "ARN": "<the ARN>",
      "SubnetIds": [
        "subnet-0cf759df15bd4dc65",
        "subnet-09e1307e8f1560d17"
      ],
      "SnapshotRetentionLimit": 0,
      "DailySnapshotTime": "03:00"
    }
  ]
}
```



```
}
```

Wenn Sie bei einem instanzbasierten Memcached-Cluster Automatic Discovery verwenden, können Sie den Konfigurationsendpunkt des Clusters verwenden, um Ihren Memcached-Client zu konfigurieren. Dies bedeutet, dass Sie einen Client verwenden müssen, der Automatic Discovery unterstützt.

Wenn Sie Automatic Discovery nicht verwenden, müssen Sie den Client so konfigurieren, dass er die einzelnen Knotenendpunkte für Lese- und Schreibvorgänge verwendet. Auch müssen Sie sie beim Hinzufügen und Entfernen von Knoten verfolgen.

Finden Sie die Endpunkte eines Valkey- oder Redis OSS-Clusters (Cluster-Modus deaktiviert) (Konsole)

Wenn ein Valkey-, Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) nur einen Knoten hat, wird der Endpunkt des Knotens sowohl für Lese- als auch für Schreibvorgänge verwendet. Wenn ein Valkey- oder Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) mehrere Knoten hat, gibt es drei Arten von Endpunkten: den primären Endpunkt, den Reader-Endpunkt und die Knotenendpunkte.

Der primäre Endpunkt ist ein DNS-Name, der immer auf den primären Knoten im Cluster aufgelöst wird. Der primäre Endpunkt ist Änderungen an Ihrem Cluster, wie Heraufstufen einer Read Replica in die Rolle des primären Knotens, gegenüber immun. Für Schreibvorgänge empfehlen wir, dass Anwendungen sich ausschließlich mit dem primären Endpunkt verbinden.

Ein Reader-Endpunkt verteilt eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Read Replicas in einem für Redis OSS bestimmten Cluster. ElastiCache Zusätzliche Faktoren, z. B. wenn die Anwendung die Verbindungen erstellt oder wie die Anwendung die Verbindungen (erneut) verwendet, bestimmen die Verteilung des Datenverkehrs. Reader-Endpunkte bleiben hinsichtlich der Cluster-Änderungen beim Hinzufügen oder Entfernen von Replicas in Echtzeit auf dem aktuellen Stand. Sie können die mehreren Read Replicas Ihres ElastiCache Redis OSS-Clusters in verschiedenen AWS Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Leser-Endpunkte sicherzustellen.

Note

Ein Leser-Endpunkt ist kein Load Balancer. Es handelt sich um einen DNS-Eintrag, der nach dem Rotationsprinzip zu einer IP-Adresse eines der Replikationsknoten aufgelöst wird.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpunkt werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren.

So finden Sie die Endpunkte eines Valkey-, Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>

- Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.

Der Cluster-Bildschirm wird mit einer Liste von Valkey- oder Valkey- oder Redis OSS-Clustern (Clustermodus deaktiviert) und Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert) angezeigt.

- Um die Primary and/or Reader-Endpunkte des Clusters zu finden, wählen Sie den Namen des Clusters (nicht die Schaltfläche links neben dem Cluster).

▼ Cluster details			
Cluster name [redacted]	Description [redacted]	Node type cache.r6g.large	Status Available
Engine Redis OSS	Engine version 6.0.5	Global datastore -	Global datastore role -
Update status Update available	Cluster mode Off	Shards 1	Number of nodes 3
Data tiering Disabled	Multi-AZ Enabled	Auto-failover Enabled	Encryption in transit Disabled
Encryption at rest Disabled	Parameter group default.redis6.x	Outpost ARN -	Configuration endpoint -
Primary endpoint [redacted]-encrypted.llru6f.ng.0001.use1.cache.ama zonaws.com:6379	Reader endpoint [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.a mazonsaws.com:6379	ARN [redacted]	

Primäre Endpunkte und Reader-Endpunkte für einen Valkey-, Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert)

Wenn sich im Cluster nur ein Knoten befindet, ist kein primärer Endpunkt vorhanden. Sie können mit dem nächsten Schritt fortfahren.

- Wenn der Valkey-, Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) über Replikatknoten verfügt, können Sie die Replikatknotenendpunkte des Clusters finden, indem Sie den Namen des Clusters und dann die Registerkarte Knoten auswählen.

Auf dem angezeigten Knotenbildschirm wird jeder Knoten im Cluster, –primärer Knoten und Replikate, – mit seinem Endpunkt aufgelistet.

☐	Node Name	Status	Current Role	Port	Endpoint
☐	test-no-001	available	primary	6379	test-no-001.usw2.cache.amazonaws.com
☐	test-no-002	available	replica	6379	test-no-002.usw2.cache.amazonaws.com
☐	test-no-003	available	replica	6379	test-no-003.usw2.cache.amazonaws.com

Knotenendpunkte für einen Valkey-, Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert)

5. So kopieren Sie einen Endpunkt in Ihre Zwischenablage:

- Suchen Sie einen Endpunkt nach dem anderen nach dem zu kopierenden Endpunkt.
- Wählen Sie das Kopiersymbol direkt vor dem Endpunkt aus.

Der Endpunkt wird nun in Ihre Zwischenablage kopiert. Informationen zur Verwendung des Endpunkts zur Verbindung mit einem Knoten finden Sie unter [Verbindung zu Memcached-Knoten herstellen](#).

Ein primärer Endpunkt von Valkey, Valkey oder Redis OSS (Clustermodus deaktiviert) sieht etwa wie folgt aus. Abhängig davon, ob die Verschlüsselung während der Übertragung aktiviert ist oder nicht, unterscheidet sich dies.

Verschlüsselung während der Übertragung nicht aktiviert

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port
```

```
redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Verschlüsselung während der Übertragung aktiviert

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port
```

```
master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Finden von Endpunkten für einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) (Konsole)

Ein Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) hat einen einzigen Konfigurationsendpunkt. Über die Verbindung zum Konfigurationsendpunkt findet die Anwendung für jede Shard im Cluster die primären Endpunkte sowie jene für Schreibvorgänge.

Um den Endpunkt eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) zu finden

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.

Der Cluster-Bildschirm mit einer Liste von Clustern wird angezeigt. Wählen Sie den Cluster aus, zu dem Sie eine Verbindung herstellen möchten.

3. Um den Konfigurationsendpunkt des Clusters zu suchen, wählen Sie den Namen des Clusters (nicht das Optionsfeld) aus.
4. Der Configuration endpoint (Konfigurationsendpunkt) wird unter Cluster details (Cluster-Details) angezeigt. Wählen Sie zum Kopieren das Symbol Copy (Kopieren) links vom Endpunkt aus.

Die Endpunkte eines Clusters finden (Konsole) (Memcached)

Alle Memcached-Endpunkte sind Endpunkte. read/write Zum Herstellen einer Verbindung mit Knoten in einem Memcached-Cluster kann Ihre Anwendung entweder die Endpunkte für jeden Knoten oder den Konfigurationsendpunkt des Clusters zusammen mit Automatic Discovery verwenden. Um Automatic Discovery verwenden zu können, benötigen Sie einen Client, der Automatic Discovery unterstützt.

Wenn Automatic Discovery verwendet wird, stellt Ihre Client-Anwendung über den Konfigurationsendpunkt eine Verbindung mit Ihrem Memcached-Cluster her. Während Sie Ihren Cluster durch Hinzufügen oder Entfernen von Knoten skalieren, „kennt“ Ihre Anwendung automatisch alle Knoten im Cluster und kann zu beliebigen von ihnen eine Verbindung herstellen. Ohne Automatic Discovery wäre das Ihre Aufgabe oder Sie müssten die Endpunkte Ihrer Anwendung jedes Mal manuell aktualisieren, wenn Sie einen Knoten hinzufügen oder löschen würden.

Wenn Sie einen Endpunkt kopieren möchten, wählen Sie das Kopiersymbol direkt vor der Endpunktadresse aus. Informationen zur Verwendung des Endpunkts zur Verbindung mit einem Knoten finden Sie unter [Verbindung zu Memcached-Knoten herstellen](#).

Die Konfiguration und die Knotenendpunkte sehen sehr ähnlich aus. Es folgen die mit Fettdruck hervorgehobenen Unterschiede.

```
myclustername. .... .cfg.usw2.cache.amazonaws.com:port # configuration endpoint
contains "cfg"
myclustername. .... .0001.usw2.cache.amazonaws.com:port # node endpoint for node 0001
```

 Important

Wenn Sie für Ihren Memcached-Konfigurationsendpunkt einen CNAME erstellen möchten, müssen Sie im CNAME `.cfg.` einschließen, damit Ihr Automatic Discovery-Client den CNAME als Konfigurationsendpunkt erkennen kann.

Suchen von Endpunkten (AWS CLI)

Für Memcached können Sie AWS CLI for Amazon verwenden, ElastiCache um die Endpunkte für Knoten und Cluster zu ermitteln.

Für Redis OSS können Sie AWS CLI for Amazon verwenden, ElastiCache um die Endpunkte für Knoten, Cluster und auch Replikationsgruppen zu ermitteln.

Themen

- [Suchen von Endpunkten für Knoten und Cluster \(AWS CLI\)](#)
- [Finden der Endpunkte für Valkey- oder Redis OSS-Replikationsgruppen \(\)AWS CLI](#)

Suchen von Endpunkten für Knoten und Cluster (AWS CLI)

Sie können den verwenden AWS CLI , um die Endpunkte für einen Cluster und seine Knoten mit dem Befehl zu ermitteln. `describe-cache-clusters` Für Valkey- oder Redis OSS-Cluster gibt der Befehl den Cluster-Endpunkt zurück. Für Memcached-Cluster gibt der Befehl den Konfigurationsendpunkt zurück. Wenn Sie den optionalen Parameter `--show-cache-node-info` einschließen, gibt der Befehl auch die Endpunkte der einzelnen Knoten im Cluster zurück.

Example

Der folgende Befehl ruft den Konfigurationsendpunkt (ConfigurationEndpoint) und einzelne Knotenendpunkte (Endpoint) für den Memcached-Cluster `mycluster` ab.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-clusters \
  --cache-cluster-id mycluster \
  --show-cache-node-info
```

Für Windows:

```
aws elasticache describe-cache-clusters ^
  --cache-cluster-id mycluster ^
  --show-cache-node-info
```

Die Ausgabe der obigen Operation sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
```

```
"CacheClusters": [
{
  "Engine": "memcached",
  "CacheNodes": [
    {
      "CacheNodeId": "0001",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
      "CustomerAvailabilityZone": "us-west-2b"
    },
    {
      "CacheNodeId": "0002",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
      "CustomerAvailabilityZone": "us-west-2b"
    },
    {
      "CacheNodeId": "0003",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
      "CustomerAvailabilityZone": "us-west-2b"
    }
  ],
  "CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "CacheParameterGroupName": "default.memcached1.4",
    "ParameterApplyStatus": "in-sync"
  },
  "CacheClusterId": "mycluster",
```



```

    "PreferredAvailabilityZone": "us-west-2b",
    "ConfigurationEndpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
    },
    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-22T21:30:29.967Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "available",
    "NumCacheNodes": 3,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {},
    "PreferredMaintenanceWindow": "mon:09:00-mon:10:00",
    "CacheNodeType": "cache.m4.large",
    "DataTiering": "disabled"
}
]
}

```

Important

Wenn Sie sich dafür entscheiden, einen CNAME für Ihren Memcached-Konfigurationsendpunkt zu erstellen, müssen Sie den `.cfg.` in den CNAME aufnehmen, damit Ihr Auto-Discovery-Client den CNAME als Konfigurationsendpunkt erkennen kann. Zum Beispiel `mycluster.cfg.local` in Ihrer `php.ini`-Datei für den Parameter `session.save_path`.

Example

Für Valkey und Redis OSS ruft der folgende Befehl die Clusterinformationen für den Einzelknotencluster `mycluster` ab.

Important

Der Parameter `--cache-cluster-id` kann mit der Cluster-ID Valkey oder Redis OSS (Clustermodus deaktiviert) mit einem Knoten oder mit bestimmten Knoten-IDs in Replikationsgruppen verwendet werden. Der Wert `--cache-cluster-id` einer

Replikationsgruppe ist ein vierstelliger Wert wie. 0001 Wenn --cache-cluster-id es sich um die ID eines Clusters (Knotens) in einer Replikationsgruppe handelt, replication-group-id ist der in der Ausgabe enthalten.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id redis-cluster \  
  --show-cache-node-info
```

Für Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id redis-cluster ^  
  --show-cache-node-info
```

Die Ausgabe der obigen Operation sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{  
  "CacheClusters": [  
    {  
      "CacheClusterStatus": "available",  
      "SecurityGroups": [  
        {  
          "SecurityGroupId": "sg-77186e0d",  
          "Status": "active"  
        }  
      ],  
      "CacheNodes": [  
        {  
          "CustomerAvailabilityZone": "us-east-1b",  
          "CacheNodeCreateTime": "2018-04-25T18:19:28.241Z",  
          "CacheNodeStatus": "available",  
          "CacheNodeId": "0001",  
          "Endpoint": {  
            "Address": "redis-cluster.amazonaws.com",  
            "Port": 6379  
          },  
          "ParameterGroupStatus": "in-sync"  
        }  
      ]  
    }  
  ]  
}
```

```

    ],
    "AtRestEncryptionEnabled": false,
    "CacheClusterId": "redis-cluster",
    "TransitEncryptionEnabled": false,
    "CacheParameterGroup": {
        "ParameterApplyStatus": "in-sync",
        "CacheNodeIdsToReboot": [],
        "CacheParameterGroupName": "default.redis3.2"
    },
    "NumCacheNodes": 1,
    "PreferredAvailabilityZone": "us-east-1b",
    "AutoMinorVersionUpgrade": true,
    "Engine": "redis",
    "AuthTokenEnabled": false,
    "PendingModifiedValues": {},
    "PreferredMaintenanceWindow": "tue:08:30-tue:09:30",
    "CacheSecurityGroups": [],
    "CacheSubnetGroupName": "default",
    "CacheNodeType": "cache.t2.small",
    "DataTiering": "disabled",
    "EngineVersion": "3.2.10",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheClusterCreateTime": "2018-04-25T18:19:28.241Z"
  }
]
}

```

Weitere Informationen finden Sie im Thema [describe-cache-clusters](#) .

Finden der Endpunkte für Valkey- oder Redis OSS-Replikationsgruppen ()AWS CLI

Sie können den verwenden AWS CLI , um die Endpunkte für eine Replikationsgruppe und ihre Cluster mit dem Befehl zu ermitteln. `describe-replication-groups` Der Befehl gibt den primären Endpunkt der Replikationsgruppe und eine Liste aller Cluster (Knoten) in der Replikationsgruppe mit ihren Endpunkten sowie den Reader-Endpunkt zurück.

Die folgende Operation ruft den primären Endpunkt und den Reader-Endpunkt der Replikationsgruppe `myreplgroup` ab. Verwenden Sie den primären Endpunkt für alle Schreibvorgänge.

```
aws elasticache describe-replication-groups \
```

```
--replication-group-id myreplgroup
```

Für Windows:

```
aws elasticache describe-replication-groups ^  
--replication-group-id myreplgroup
```

Die Ausgabe dieser Operation sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{  
  "ReplicationGroups": [  
    {  
      "Status": "available",  
      "Description": "test",  
      "NodeGroups": [  
        {  
          "Status": "available",  
          "NodeGroupMembers": [  
            {  
              "CurrentRole": "primary",  
              "PreferredAvailabilityZone": "us-west-2a",  
              "CacheNodeId": "0001",  
              "ReadEndpoint": {  
                "Port": 6379,  
                "Address": "myreplgroup-001.amazonaws.com"  
              },  
              "CacheClusterId": "myreplgroup-001"  
            },  
            {  
              "CurrentRole": "replica",  
              "PreferredAvailabilityZone": "us-west-2b",  
              "CacheNodeId": "0001",  
              "ReadEndpoint": {  
                "Port": 6379,  
                "Address": "myreplgroup-002.amazonaws.com"  
              },  
              "CacheClusterId": "myreplgroup-002"  
            },  
            {  
              "CurrentRole": "replica",  
              "PreferredAvailabilityZone": "us-west-2c",  
              "CacheNodeId": "0001",  
              "ReadEndpoint": {
```

```
        "Port": 6379,
        "Address": "myreplgroup-003.amazonaws.com"
    },
    "CacheClusterId": "myreplgroup-003"
  }
],
"NodeGroupId": "0001",
"PrimaryEndpoint": {
  "Port": 6379,
  "Address": "myreplgroup.amazonaws.com"
},
"ReaderEndpoint": {
  "Port": 6379,
  "Address": "myreplgroup-ro.amazonaws.com"
}
}
],
"ReplicationGroupId": "myreplgroup",
"AutomaticFailover": "enabled",
"SnapshottingClusterId": "myreplgroup-002",
"MemberClusters": [
  "myreplgroup-001",
  "myreplgroup-002",
  "myreplgroup-003"
],
"PendingModifiedValues": {}
}
]
```

Weitere Informationen finden Sie unter [describe-replication-groups](#) in der Referenz zum AWS CLI - Befehl.

Endpunkte finden (API) ElastiCache

Für Memcached können Sie die ElastiCache Amazon-API verwenden, um die Endpunkte für Knoten und Cluster zu ermitteln.

Für Redis OSS können Sie die ElastiCache Amazon-API verwenden, um die Endpunkte für Knoten, Cluster und auch Replikationsgruppen zu ermitteln.

Themen

- [Suche nach Endpunkten für Knoten und Cluster \(API\) ElastiCache](#)
- [Suche nach Endpunkten für Valkey- oder Redis OSS-Replikationsgruppen \(API\) ElastiCache](#)

Suche nach Endpunkten für Knoten und Cluster (API) ElastiCache

Sie können die ElastiCache API verwenden, um mit der Aktion die Endpunkte für einen Cluster und seine Knoten zu ermitteln. `DescribeCacheClusters` Für Valkey- oder Redis OSS-Cluster gibt der Befehl den Cluster-Endpoint zurück. Für Memcached-Cluster gibt der Befehl den Konfigurationsendpoint zurück. Wenn Sie den optionalen Parameter `ShowCacheNodeInfo` einschließen, gibt die Aktion auch die Endpunkte der einzelnen Knoten im Cluster zurück.

Example

Für Memcached ruft der folgende Befehl den Konfigurationsendpoint (*ConfigurationEndpoint*) und die einzelnen Knotenendpunkte (*Endpoint*) für den Memcached-Cluster mycluster ab.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&CacheClusterId=mycluster  
&ShowCacheNodeInfo=true  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Important

Wenn Sie sich dafür entscheiden, einen CNAME für Ihren Memcached-Konfigurationsendpoint zu erstellen, müssen Sie den `.cfg.` in den CNAME aufnehmen,

damit Ihr Auto-Discovery-Client den CNAME als Konfigurationsendpunkt erkennen kann. Zum Beispiel `mycluster.cfg.local` in Ihrer `php.ini`-Datei für den Parameter `session.save_path`.

Suche nach Endpunkten für Valkey- oder Redis OSS-Replikationsgruppen (API) ElastiCache

Sie können die ElastiCache API verwenden, um mit der Aktion die Endpunkte für eine Replikationsgruppe und ihre Cluster zu ermitteln. `DescribeReplicationGroups` Die Aktion gibt den primären Endpunkt der Replikationsgruppe und eine Liste aller Cluster in der Replikationsgruppe mit ihren Endpunkten sowie den Reader-Endpunkt zurück.

Der folgende Vorgang ruft den primären Endpunkt (`PrimaryEndpoint`), den Leser-Endpunkt (`ReaderEndpoint`) und die einzelnen Knotenendpunkte (`ReadEndpoint`) für die Replikationsgruppe ab. `myreplgroup` Verwenden Sie den primären Endpunkt für alle Schreibvorgänge.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&ReplicationGroupId=myreplgroup  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie unter [DescribeReplicationGroups](#).

Arbeiten mit Shards in ElastiCache

Ein Shard (API/CLI: Node Group) ist eine Sammlung von eins bis sechs ElastiCache für Valkey- oder Redis-OSS-Knoten. Ein Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) wird niemals mehr als einen Shard haben. Mit Shards können Sie große Datenbanken in kleinere, schnellere und einfacher zu verwaltende Teile aufteilen, die als Daten-Shards bezeichnet werden. Dadurch kann die Datenbankeffizienz erhöht werden, indem die Operationen auf mehrere separate Abschnitte verteilt werden. Die Verwendung von Shards kann viele Vorteile bieten, darunter eine verbesserte Leistung, Skalierbarkeit und Kosteneffizienz.

Sie können einen Cluster mit einer höheren Anzahl an Shards und einer geringeren Anzahl an Replikaten mit bis zu 90 Knoten pro Cluster erstellen. Diese Clusterkonfiguration reicht von 90 Shards

und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die Anzahl erlaubter Replikate entspricht. Die Daten des Clusters werden über die Shards des Clusters hinweg partitioniert. Wenn ein Shard mehr als einen Knoten enthält, implementiert der Shard die Replikation, wobei ein Knoten der read/write primäre Knoten und die anderen Knoten schreibgeschützte Replikatknoten sind.

Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Engine-Version Valkey 7.2 und höher oder Redis OSS 5.0.6 bis 7.1 ist. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

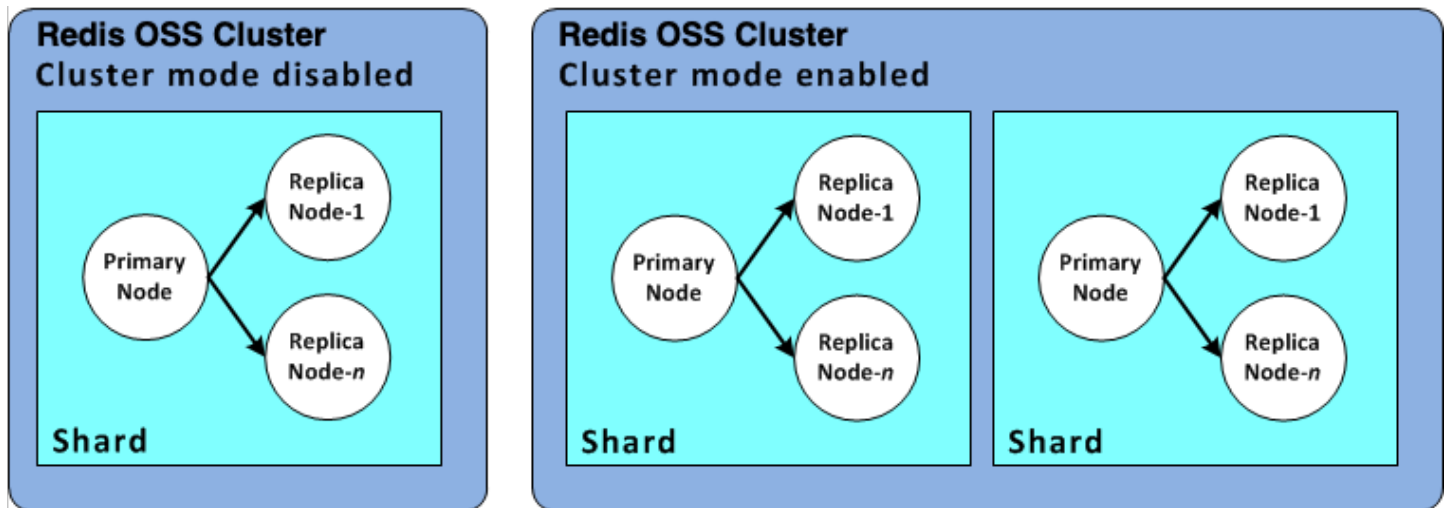
Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, siehe [AWS -Service-Limits](#) und wählen Sie den Limittyp Knoten pro Cluster pro Instance-Typ.

Wenn Sie mithilfe der ElastiCache Konsole einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) erstellen, geben Sie die Anzahl der Shards im Cluster und die Anzahl der Knoten in den Shards an. Weitere Informationen finden Sie unter [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#). Wenn Sie die ElastiCache API verwenden oder AWS CLI um einen Cluster (in der KnotengruppeAPI/CLI), you can configure the number of nodes in a shard (API/CLI: Replikationsgruppe genannt) unabhängig zu erstellen. Weitere Informationen finden Sie hier:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Für jeden Knoten in einem Shard gelten dieselben Rechner-, Arbeitsspeicher- und Festspeicherspezifikationen. Mit der ElastiCache API können Sie Shard-weite Attribute wie die Anzahl der Knoten, Sicherheitseinstellungen und Systemwartungsfenster steuern.



Valkey- oder Redis OSS-Shard-Konfigurationen

Weitere Informationen erhalten Sie unter [Offline-Resharding für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#) und [Online-Resharding für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#).

Auffinden der ID eines Shards

Sie können die ID eines Shards mithilfe der AWS-Managementkonsole, der oder der API finden.
AWS CLI ElastiCache

Unter Verwendung der AWS-Managementkonsole

Themen

- [Für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#)
- [Für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)

Für Valkey oder Redis OSS (Clustermodus deaktiviert)

Für Valkey oder Redis OSS (Cluster-Modus deaktiviert) sind Replikationsgruppen-Shards immer verfügbar. IDs 0001

Für Valkey oder Redis OSS (Clustermodus aktiviert)

Das folgende Verfahren verwendet die AWS-Managementkonsole, um die Shard-ID einer Replikationsgruppe von Valkey oder Redis OSS (Clustermodus aktiviert) zu ermitteln.

So finden Sie die Shard-ID in einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS und dann den Namen der Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) aus, für die Sie den Shard suchen möchten. IDs
3. In der Spalte Shard-Name ist die Shard-ID die letzten vier Ziffern des Shard-Namens.

Verwenden Sie den AWS CLI

Um Shard-IDs (Knotengruppen) für entweder Valkey oder Redis OSS (Clustermodus deaktiviert) oder Valkey oder Redis OSS (Clustermodus aktiviert) zu finden, verwenden Sie den AWS CLI Vorgang `describe-replication-groups` mit dem folgenden optionalen Parameter.

- **--replication-group-id**—Ein optionaler Parameter, der bei Verwendung die Ausgabe auf die Details der angegebenen Replikationsgruppe beschränkt. Wenn dieser Parameter weggelassen wird, werden die Details von bis zu 100 Replikationsgruppen zurückgegeben.

Example

Dieser Befehl gibt die Details für `sample-repl-group` zurück.

Für Linux, macOS oder Unix:

```
aws elasticache describe-replication-groups \
  --replication-group-id sample-repl-group
```

Für Windows:

```
aws elasticache describe-replication-groups ^
--replication-group-id sample-repl-group
```

Die Ausgabe dieses Befehls sieht in etwa wie folgt aus: Die Shard-IDs (Knotengruppen-IDs) sollen das Auffinden erleichtern *highlighted*.

```
{
  "ReplicationGroups": [
```

```
{
  "Status": "available",
  "Description": "2 shards, 2 nodes (1 + 1 replica)",
  "NodeGroups": [
    {
      "Status": "available",
      "Slots": "0-8191",
      "NodeGroupId": "0001",
      "NodeGroupMembers": [
        {
          "PreferredAvailabilityZone": "us-west-2c",
          "CacheNodeId": "0001",
          "CacheClusterId": "sample-repl-group-0001-001"
        },
        {
          "PreferredAvailabilityZone": "us-west-2a",
          "CacheNodeId": "0001",
          "CacheClusterId": "sample-repl-group-0001-002"
        }
      ]
    },
    {
      "Status": "available",
      "Slots": "8192-16383",
      "NodeGroupId": "0002",
      "NodeGroupMembers": [
        {
          "PreferredAvailabilityZone": "us-west-2b",
          "CacheNodeId": "0001",
          "CacheClusterId": "sample-repl-group-0002-001"
        },
        {
          "PreferredAvailabilityZone": "us-west-2a",
          "CacheNodeId": "0001",
          "CacheClusterId": "sample-repl-group-0002-002"
        }
      ]
    }
  ],
  "ConfigurationEndpoint": {
    "Port": 6379,
    "Address": "sample-repl-
group.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
  },
}
```

```

    "ClusterEnabled": true,
    "ReplicationGroupId": "sample-repl-group",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "13:00-14:00",
    "MemberClusters": [
        "sample-repl-group-0001-001",
        "sample-repl-group-0001-002",
        "sample-repl-group-0002-001",
        "sample-repl-group-0002-002"
    ],
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled",
    "PendingModifiedValues": {}
  }
]
}

```

Verwenden der API ElastiCache

Verwenden Sie den AWS CLI Vorgang mit dem folgenden optionalen Parameter, um Shard-IDs (Knotengruppen) für Valkey oder Redis OSS (Clustermodus deaktiviert) oder Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus aktiviert) `describe-replication-groups` zu finden.

- **ReplicationGroupId**—Ein optionaler Parameter, der bei Verwendung die Ausgabe auf die Details der angegebenen Replikationsgruppe beschränkt. Wenn dieser Parameter weggelassen wird, werden die Details von bis zu **xxx** Replikationsgruppen zurückgegeben.

Example

Dieser Befehl gibt die Details für `sample-repl-group` zurück.

Für Linux, macOS oder Unix:

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroup
&ReplicationGroupId=sample-repl-group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>

```

Vergleich knotenbasierter Valkey-, Memcached- und Redis OSS-Cluster

Amazon ElastiCache unterstützt die OSS-Cache-Engines Valkey, Memcached und Redis. Jede Engine hat ihre Vorteile. Anhand der Informationen in diesem Thema können Sie die Engine und Version auswählen, die Ihren Anforderungen am besten entspricht.

Important

Nachdem Sie einen Cache, einen knotenbasierten Cluster oder eine Replikationsgruppe erstellt haben, können Sie ein Upgrade auf eine neuere Engine-Version durchführen, aber Sie können kein Downgrade auf eine ältere Engine-Version durchführen. Wenn Sie eine ältere Engine-Version verwenden möchten, müssen Sie den vorhandenen Cache, den knotenbasierten Cluster oder die Replikationsgruppe löschen und sie mit der früheren Engine-Version erneut erstellen.

Auf den ersten Blick ähneln sich die Engines. Jede Engine befindet sich in einem als Hauptspeicher-basierten Key-Value Store. In der Praxis bestehen jedoch erhebliche Unterschiede.

Wählen Sie Memcached aus, wenn Folgendes auf Sie zutrifft:

- Sie benötigen ein möglichst einfaches Modell.
- Sie müssen große Knoten mit mehreren Kernen oder Threads ausführen.
- Sie müssen Knoten skalieren, hinzufügen und entfernen können, wenn die Anforderungen an Ihr System steigen oder sinken.
- Sie müssen Objekte zwischenspeichern.

Wählen Sie Valkey oder Redis OSS with, ElastiCache wenn Folgendes auf Sie zutrifft:

- ElastiCache Version 7.2 für Valkey oder Version 7.0 (erweitert) für Redis OSS

[Sie möchten Functions, Sharded Pub/Sub oder ACL-Verbesserungen verwenden.](#) Weitere Informationen finden Sie unter [Redis OSS Version 7.0](#) (erweitert).

- ElastiCache Version 6.2 (erweitert) für Redis OSS

Sie möchten die Möglichkeit haben, Daten zwischen Speicher und SSD mit dem R6gd-Knotentyp zu staffeln. Weitere Informationen finden Sie unter [Daten-Tiering](#).

- ElastiCache Version 6.0 (erweitert) für Redis OSS

Sie möchten Benutzer mit rollenbasierter Zugriffskontrolle authentifizieren.

Weitere Informationen finden Sie unter [Redis OSS Version 6.0 \(erweitert\)](#).

- ElastiCache Version 5.0.0 (erweitert) für Redis OSS

Sie möchten [Redis OSS-Streams](#) verwenden, eine Protokolldatenstruktur, die es Produzenten ermöglicht, neue Elemente in Echtzeit anzuhängen, und es Verbrauchern ermöglicht, Nachrichten entweder blockierend oder nicht blockierend zu konsumieren.

Weitere Informationen finden Sie unter [Redis OSS Version 5.0.0 \(erweitert\)](#).

- ElastiCache Version 4.0.10 (erweitert) für Redis OSS

Unterstützt sowohl die Verschlüsselung als auch das dynamische Hinzufügen oder Entfernen von Shards zu Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert).

Weitere Informationen finden Sie unter [Redis OSS Version 4.0.10 \(erweitert\)](#).

Die folgenden Versionen sind veraltet, haben das Ende ihrer Lebensdauer erreicht oder stehen kurz vor dem Ende.

- ElastiCache Version 3.2.10 (erweitert) für Redis OSS

Unterstützt die Fähigkeit, Shards dynamisch zu Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) hinzuzufügen oder zu entfernen.

 Important

Derzeit unterstützt ElastiCache 3.2.10 für Redis OSS keine Verschlüsselung.

Weitere Informationen finden Sie hier:

- [Redis OSS Version 3.2.10 \(erweitert\)](#)

- Bewährte Methoden für das Online-Resharding für Redis OSS. Weitere Informationen finden Sie im Folgenden:
 - [Bewährte Methoden: Online-Resharding](#)
 - [Online-Resharding und Shard-Rebalancing für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
 - [Weitere Informationen zur Skalierung von Redis OSS-Clustern finden Sie unter Skalierung.](#)
-
- ElastiCache Version 3.2.6 (erweitert) für Redis OSS

Wenn Sie die Funktionalität früherer Redis OSS-Versionen sowie die folgenden Funktionen benötigen, wählen Sie 3.2.6:

- Verschlüsselung während der Übertragung. Weitere Informationen finden Sie unter [Amazon ElastiCache for Redis OSS In-Transit Encryption](#).
 - Verschlüsselung im Ruhezustand. Weitere Informationen finden Sie unter [Amazon ElastiCache for Redis OSS At-Rest Encryption](#).
-
- ElastiCache (Clustermodus aktiviert) Version 3.2.4 für Redis OSS

Wenn Sie die Funktionalität von 2.8.x sowie die folgenden Funktionen benötigen, wählen Sie 3.2.4 (Clustermodus):

- Sie müssen Ihre Daten auf zwei bis 500 Knotengruppen verteilen (nur im Cluster-Modus).
 - Sie benötigen räumliche Indizierung (Cluster-Modus oder Nicht-Cluster-Modus).
 - Sie müssen nicht mehrere Datenbanken unterstützen.
-
- ElastiCache (nicht geclusterter Modus) 2.8.x und 3.2.4 (erweitert) für Redis OSS

Falls Folgendes auf Sie zutrifft, 2.8.x oder 3.2.4 (nicht geclusterter Modus):

- Sie benötigen komplexe Datentypen, wie Zeichenfolgen, Hashes, Listen, Sets, Sorted Sets und Bitmaps.
- Sie müssen Datensätze im Speicher sortieren oder ihnen ein Rang zuweisen.
- Sie benötigen Persistenz für Ihren Schlüsselspeicher.
- Sie müssen Ihre Daten für leseintensive Anwendungen vom Primärknoten in eine oder mehrere Lesereplikate replizieren.
- Sie benötigen ein automatisches Failover, wenn Ihr primärer Knoten ausfällt.
- Sie benötigen Veröffentlichungs- und Subskriptionsfunktionen (Pub/Sub), um Clients über

- Sie benötigen Sicherungs- und Wiederherstellungsfunktionen für knotenbasierte Cluster sowie für serverlose Caches.
- Sie müssen mehrere Datenbanken unterstützen.

Zusammenfassung des Vergleichs von Memcached, Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert)

	Memcached	Valkey oder Redis OSS (Clustermodus deaktiviert)	Valkey oder Redis OSS (Clustermodus aktiviert)
Engine-Versionen +	1.4.5 und höher	4.0.10 und höher	4.0.10 und höher
Datentypen	Einfach	2.8.x – Complex * Komplex	3.2.x und höher – Komplex
Datenpartitionierung	Ja	Nein	Ja
Modifizierbarer Cluster	Ja	Ja	3.2.10 und höher – Beschränkt
Online-Resharding	Nein	Nein	3.2.10 und höher
Verschlüsselung	während der Übertragung 1.6.12 und höher	4.0.10 und höher	4.0.10 und höher
Daten-Tiering	Nein	6.2 und später	6.2 und später
Compliance-Zertifizierungen			
Compliance-Zertifizierung			
FedRAMP	Ja – 1.6.12 und höher	4.0.10 und höher	4.0.10 und höher
HIPAA	Ja – 1.6.12 und höher	4.0.10 und höher	4.0.10 und höher
PCI DSS	Ja	4.0.10 und höher	4.0.10 und höher
Multi-Thread	Ja	Nein	Nein
Upgrade des Knotentyps	Nein	Ja	Ja

	Memcached	Valkey oder Redis OSS (Clustermodus deaktiviert)	Valkey oder Redis OSS (Clustermodus aktiviert)
Engine-Upgrade	Ja	Ja	Ja
Hohe Verfügbarkeit (Replikation)	Nein	Ja	Ja
Automatisches Failover	Nein	Optional	Erforderlich
Pub-/Sub-Funktionen	Nein	Ja	Ja
Sortierte Sätze	Nein	Ja	Ja
Backup und Wiederherstellung	Nur für serverlose Caches, gilt nicht für knotenbasierte Cluster	Ja	Ja
Räumliche Indizierung	Nein	4.0.10 und höher	Ja

Hinweise:

Zeichenfolge, Objekte (wie Datenbanken)

* Zeichenfolge, Sets, Sorted Sets, Listen, Hashes, Bitmaps, HyperLogLog

Zeichenfolge, Sets, Sorted Sets, Listen, Hashes, Bitmaps, HyperLogLog, räumliche Indizes

+ Schließt Versionen aus, die veraltet sind, das Ende ihrer Nutzungsdauer erreicht haben oder bald ablaufen werden.

Nachdem Sie die Engine für Ihren Cluster ausgewählt haben, verwenden Sie am besten die neueste Version dieser Engine. Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

Online-Migration für Valkey oder Redis OSS

Mithilfe der Online-Migration können Sie Ihre Daten von Ihrem selbst gehosteten Open-Source-Betriebssystem Valkey oder Redis OSS auf Amazon zu Amazon migrieren. EC2 ElastiCache

Dies bezieht sich auf die Migration von einer selbst gehosteten Instanz zum Service. ElastiCache Informationen zum Upgrade von Redis OSS auf Valkey finden Sie unter. ElastiCache [Aufrüstung von Motorversionen, einschließlich motorübergreifender Upgrades](#)

Note

Die Online-Migration zu ElastiCache serverlosen Caches oder Clustern, die auf dem Knotentyp r6gd ausgeführt werden, wird nicht unterstützt.

Übersicht

Um Ihre Daten von Valkey oder Redis Open Source OSS, das auf Amazon läuft, EC2 zu Amazon zu migrieren, ElastiCache ist eine bestehende oder neu erstellte Amazon-Bereitstellung erforderlich. ElastiCache Die Bereitstellung muss eine migrationsbereite Konfiguration haben. Sie sollte außerdem der gewünschten Konfiguration entsprechen, mit Attributen wie Instance-Typ, Anzahl der Shards und Anzahl der Replikate.

Die Online-Migration ist für die Datenmigration von selbst gehostetem Open-Source-Betriebssystem Valkey oder Redis OSS auf Amazon konzipiert und nicht für EC2 die ElastiCache Übertragung von Daten zwischen Clustern. ElastiCache

Important

Wir empfehlen dringend, die folgenden Abschnitte vollständig zu lesen, bevor Sie den Online-Migrationsprozess starten.

Die Migration beginnt, sobald Sie die API-Operation oder den AWS CLI -Befehl `StartMigration` aufrufen. Bei der Migration deaktivierter Cluster im Valkey- oder Redis OSS-Clustermodus macht der Migrationsprozess den primären Knoten des Valkey- oder Redis OSS-Clusters zu einer Replik Ihres primären ElastiCache Valkey- oder Redis OSS-Quellclusters. Bei der Migration von Clustern mit aktiviertem Valkey- oder Redis OSS-Clustermodus macht der Migrationsprozess den Primärknoten jedes ElastiCache Shards zu einem Replikat des entsprechenden Shards Ihres Quellclusters, der dieselben Slots besitzt.

Sobald die Änderungen auf der Client-Seite durchgeführt wurden, rufen Sie die `CompleteMigration`-API-Operation auf. Durch diesen API-Vorgang wird Ihre ElastiCache Bereitstellung auf Ihre primäre Valkey- oder Redis OSS-Bereitstellung mit Primär- und Replikatknoten

(falls zutreffend) umgestellt. Nun können Sie die Client-Anwendung veranlassen, Daten ab jetzt nach ElastiCache zu schreiben. Während der Migration können Sie den Status der Replikation überprüfen, indem Sie den Befehl [valkey-cli INFO auf Ihren Valkey-Knoten](#) und auf den Primärknoten ausführen. ElastiCache

Schritte zur Migration

Die folgenden Themen beschreiben den Prozess zur Migration der Daten:

- [Quelle und Ziel für die Migration vorbereiten](#)
- [Testen der Datenmigration](#)
- [Migration starten](#)
- [Verifizieren des Fortschritts der Datenmigration](#)
- [Datenmigration abschließen](#)

Quelle und Ziel für die Migration vorbereiten

Mit diesen Schritten können Sie sich darauf vorbereiten, Ihre Daten von einer selbst gehosteten Valkey- oder Redis-Quelle auf oder von einem Redis OSS-Cluster EC2 zu ElastiCache einem Valkey-Cluster zu migrieren. ElastiCache

Dies bezieht sich auf die Migration von einer selbst gehosteten Instanz zum Service. ElastiCache Informationen zum Upgrade von Redis OSS auf Valkey finden Sie unter. ElastiCache [Aufrüstung von Motorversionen, einschließlich motorübergreifender Upgrades](#)

Sie müssen sicherstellen, dass alle vier der unten genannten Voraussetzungen erfüllt sind, bevor Sie die Migration über die ElastiCache Konsole, API oder AWS CLI starten.

Um Ihre Quell- und Ziel-OSS-Knoten von Valkey oder Redis für die Migration vorzubereiten

1. Identifizieren Sie die ElastiCache Zielbereitstellung und stellen Sie sicher, dass Sie Daten dorthin migrieren können.

Eine bestehende oder neu erstellte ElastiCache Bereitstellung sollte die folgenden Anforderungen für die Migration erfüllen:

- Es verwendet Valkey oder Redis OSS 5.0.6 oder höher.
- Die Verschlüsselung bei der Übertragung ist nicht aktiviert.

- Es hat Multi-AZ aktiviert.
 - Es steht ausreichend Speicherplatz für die Daten aus Ihrem Valkey- oder Redis OSS-Cluster zur Verfügung. Zum Konfigurieren der richtigen Einstellungen für reservierten Speicher siehe [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).
 - Wenn der Clustermodus deaktiviert ist, können Sie direkt von Valkey oder Redis OSS Version 2.8.21 auf Valkey oder Redis OSS Version 5.0.6 und höher migrieren, wenn Sie die CLI oder Valkey oder Redis OSS Versionen 5.0.6 und höher über die CLI oder Konsole verwenden. Wenn der Clustermodus aktiviert ist, können Sie direkt von jeder Valkey- oder Redis OSS-Version mit aktiviertem Clustermodus auf Redis OSS Version 5.0.6 und höher migrieren, wenn Sie die CLI- oder Redis OSS-Versionen 5.0.6 über die CLI oder die Konsole verwenden.
 - Die Anzahl der Shards in Quell- und Zieldatei stimmt überein.
 - Ist nicht Teil eines globalen Datenspeichers.
 - Datenklassifizierung ist deaktiviert.
2. Stellen Sie sicher, dass die Konfigurationen Ihres Open-Source-Betriebssystems Valkey oder Redis OSS und die Bereitstellung kompatibel sind. ElastiCache

In der ElastiCache Zielbereitstellung sollten mindestens alle folgenden Komponenten mit Ihrer Valkey- oder Redis OSS-Konfiguration für die Replikation kompatibel sein:

- In Ihrem Cluster sollte AUTH nicht aktiviert sein.
- Die Konfiguration `protected-mode` sollte auf `eingestellt` sein.
- Wenn Sie eine `bind` Konfiguration in Ihrer Valkey- oder Redis OSS-Konfiguration haben, sollte diese aktualisiert werden, um Anfragen von ElastiCache Knoten zuzulassen.
- Die Anzahl der logischen Datenbanken auf dem ElastiCache Knoten und Ihrem Valkey- oder Redis OSS-Cluster sollte identisch sein. Dieser Wert wird `databases` in der Valkey- oder Redis OSS-Konfiguration festgelegt.
- Valkey- oder Redis OSS-Befehle, die Datenänderungen vornehmen, sollten nicht umbenannt werden, damit die Daten erfolgreich repliziert werden können. Zum Beispielsync,,, undpsync. info config command cluster
- Um die Daten aus Ihrem Valkey- oder Redis-OSS-Cluster zu replizieren ElastiCache, stellen Sie sicher, dass ausreichend CPU und Arbeitsspeicher vorhanden sind, um diese zusätzliche Last zu bewältigen. Diese Last stammt aus der RDB-Datei, die von Ihrem Valkey- oder Redis-OSS-Cluster erstellt und über das Netzwerk an den Knoten übertragen wurde. ElastiCache
- Alle Valkey- oder Redis OSS-Instanzen im Quellcluster sollten auf demselben Port laufen.

3. Stellen Sie sicher, dass Ihre Instances eine Verbindung herstellen können, ElastiCache indem Sie wie folgt vorgehen:
 - Stellen Sie sicher, dass die IP-Adresse jeder Instance privat ist.
 - Weisen Sie die ElastiCache Bereitstellung derselben Virtual Private Cloud (VPC) zu oder erstellen Sie sie in derselben virtuellen privaten Cloud (VPC) wie Ihr Valkey- oder Redis-OSS auf Ihrer Instanz (empfohlen).
 - Wenn sie unterschiedlich VPCs sind, richten Sie VPC-Peering ein, um den Zugriff zwischen den Knoten zu ermöglichen. Weitere Informationen zum VPC-Peering siehe [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#).
 - Die mit Ihren Valkey- oder Redis OSS-Instances verknüpfte Sicherheitsgruppe sollte eingehenden Datenverkehr von Knoten zulassen. ElastiCache
4. Stellen Sie sicher, dass Ihre Anwendung den Datenverkehr nach Abschluss der Datenmigration an die ElastiCache Knoten weiterleiten kann. Weitere Informationen finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#).

Testen der Datenmigration

Nachdem alle Voraussetzungen erfüllt sind, können Sie die Migrationseinrichtung mithilfe der AWS-Managementkonsole ElastiCache API oder überprüfen AWS CLI. Das folgende Beispiel illustriert die Verwendung der CLI.

Testen Sie die Migration, indem Sie den Befehl `test-migration` mit den folgenden Parametern aufrufen:

- `--replication-group-id` – Die ID der Replikationsgruppe, in die Daten migriert werden sollen.
- `--customer-node-endpoint-list` – Die Liste der Endpunkte, von denen Daten migriert werden sollen. Die Liste sollte nur ein Element enthalten.

Nachstehend finden Sie ein Beispiel für die Verwendung der CLI.

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

ElastiCache validiert das Migrations-Setup ohne tatsächliche Datenmigration.

Migration starten

Nachdem alle Voraussetzungen erfüllt sind, können Sie mit der Datenmigration mithilfe der AWS-Managementkonsole ElastiCache API oder beginnen AWS CLI. Bei aktiviertem Clustermodus und einer unterschiedlichen Slot-Migration wird vor der Live-Migration ein Resharding durchgeführt. Das folgende Beispiel illustriert die Verwendung der CLI.

Note

Wir empfehlen die Verwendung der `TestMigration`-API zur Validierung der Migrationseinrichtung. Dies ist allerdings völlig optional.

Starten Sie die Migration, indem Sie den Befehl `start-migration` mit den folgenden Parametern aufrufen:

- `--replication-group-id`— ID der ElastiCache Zielreplikationsgruppe
- `--customer-node-endpoint-list`— Eine Liste von Endpunkten mit entweder DNS- oder IP-Adressen und dem Port, auf dem Ihr Valkey- oder Redis-OSS-Quellcluster läuft. Die Liste kann sowohl bei deaktiviertem Clustermodus als auch bei aktiviertem Clustermodus nur ein Element enthalten. Wenn Sie die verkettete Replikation aktiviert haben, kann der Endpunkt auf ein Replikat statt auf den primären Knoten in Ihrem Valkey- oder Redis OSS-Cluster verweisen.

Nachstehend finden Sie ein Beispiel für die Verwendung der CLI.

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

Wenn Sie diesen Befehl ausführen, konfiguriert sich der ElastiCache primäre Knoten (in jedem Shard) so, dass er ein Replikat Ihrer Valkey- oder Redis-OSS-Instanz wird (in einem entsprechenden Shard, der dieselben Steckplätze in einem clusterfähigen Redis besitzt). Der Status des ElastiCache Clusters ändert sich in Migration und die Datenmigration von Ihrer Valkey - oder Redis OSS-Instanz zum primären Knoten beginnt. ElastiCache Abhängig von der Größe der Daten und der Auslastung Ihrer Valkey- oder Redis OSS-Instanz kann es eine Weile dauern, bis die Migration abgeschlossen ist. Sie können den Fortschritt der Migration überprüfen, indem Sie den Befehl [valkey-cli INFO auf Ihrer Valkey-Instanz](#) und Ihrem Primärknoten ausführen. ElastiCache

Nach erfolgreicher Replikation werden alle Schreibvorgänge auf Ihre Valkey- oder Redis OSS-Instanzen auf den Cluster übertragen. ElastiCache Sie können ElastiCache Knoten für Lesevorgänge verwenden. Sie können jedoch nicht in den ElastiCache-Cluster schreiben. Wenn mit einem ElastiCache Primärknoten andere Replikatknoten verbunden sind, replizieren diese Replikatknoten weiterhin vom ElastiCache Primärknoten aus. Auf diese Weise werden alle Daten aus Ihrem Valkey- oder Redis OSS-Cluster auf alle Knoten im Cluster repliziert. ElastiCache

Wenn ein ElastiCache primärer Knoten kein Replikat Ihrer Valkey- oder Redis OSS-Instanz werden kann, versucht er es mehrmals, bevor er sich schließlich wieder zum primären Knoten hochstuft. Der Status des ElastiCache-Clusters wechselt dann zu `available` (Verfügbar) und ein Replikationsgruppenereignis zum Fehlschlagen des Initiierens der Migration wird gesendet. Prüfen Sie Folgendes, um einen solchen Fehlschlag zu diagnostizieren:

- Betrachten Sie das Replikationsgruppenereignis. Verwenden Sie die spezifische Informationen aus dem Ereignis, um den Migrationsfehler zu beheben.
- Wenn das Ereignis keine spezifischen Informationen bereitstellt, müssen Sie sicherstellen, dass die Leitlinien in [Quelle und Ziel für die Migration vorbereiten](#) befolgt wurden.
- Stellen Sie sicher, dass die Routing-Konfiguration für Ihre VPC und Subnetze den Verkehr zwischen ElastiCache Knoten und Ihren Valkey- oder Redis OSS-Instances zulässt.
- Stellen Sie sicher, dass die mit Ihren Valkey- oder Redis OSS-Instances verknüpfte Sicherheitsgruppe eingehenden Datenverkehr von Knoten zulässt. ElastiCache
- Weitere Informationen zu replikationsspezifischen Fehlern finden Sie in den Valkey- oder Redis OSS-Protokollen für Ihre Instances.

Verifizieren des Fortschritts der Datenmigration

Nach Beginn der Datenmigration können Sie den Verlauf folgendermaßen überwachen:

- Stellen Sie sicher, dass Valkey oder Redis OSS up im `INFO` Befehl auf den ElastiCache primären Knoten enthalten `master_link_status` ist. Sie finden diese Informationen auch in der ElastiCache Konsole. Wählen Sie den Cluster aus und beobachten Sie unter CloudWatch Metriken den Integritätsstatus des primären Links. Die Daten sind synchronisiert, sobald der Wert 1 erreicht hat.
- Sie können überprüfen, ob das ElastiCache Replikat online ist, indem Sie den `INFO` Befehl auf Ihren Valkey- oder Redis OSS-Instances ausführen. Dadurch werden auch Informationen zur Replikationsverzögerung bereitgestellt.

- Überprüfen Sie, ob der Client-Ausgabepuffer niedrig ist, indem Sie den Befehl [CLIENT LIST](#) auf Ihren Valkey- oder Redis OSS-Instances verwenden.

Nach Abschluss der Datenmigration sind die Daten mit allen neuen Schreibvorgängen synchronisiert, die auf die primären Knoten Ihres Valkey- oder Redis OSS-Clusters eingehen.

Datenmigration abschließen

Wenn Sie bereit sind, zum ElastiCache Cluster überzugehen, verwenden Sie den `complete-migration` CLI-Befehl mit den folgenden Parametern:

- `--replication-group-id` – Die ID für die Replikationsgruppe.
- `--force` – Ein Wert, der zwingt, dass die Migration beendet wird, ohne sicherzustellen, dass die Daten synchron sind.

Im Folgenden wird ein Beispiel gezeigt.

```
aws elasticache complete-migration --replication-group-id test-cluster
```

Während Sie diesen Befehl ausführen, beendet der ElastiCache primäre Knoten (in jedem Shard) die Replikation von Ihrer Valkey- oder Redis OSS-Instance und stuft ihn zum primären Knoten herauf. Diese Übergabe wird normalerweise innerhalb von Minuten abgeschlossen. Prüfen Sie zum Bestätigen der Übergabe an den primären Knoten auf das Ereignis `Complete Migration successful for test-cluster`. An diesem Punkt können Sie Ihre Anwendung auf Schreib- und Lesevorgänge ausrichten. ElastiCache ElastiCache Der Clusterstatus sollte von `migriert` zu `verfügbar` geändert werden.

Wenn die Heraufstufung zum Primärknoten fehlschlägt, repliziert der ElastiCache primäre Knoten weiterhin von Ihrer Valkey- oder Redis OSS-Instanz aus. Der ElastiCache Cluster befindet sich weiterhin im Migrationsstatus, und es wird eine Ereignismeldung über den Fehler an die Replikationsgruppe gesendet. Beachten Sie zum Beheben dieses Fehlers Folgendes:

- Prüfen Sie das Replikationsgruppenereignis. Verwenden Sie spezifische Informationen aus dem Ereignis, um den Fehler zu beheben.
- Unter Umständen wird eine Ereignismeldung angezeigt, weil die Daten nicht synchronisiert sind. Wenn ja, stellen Sie sicher, dass die ElastiCache primäre Instanz von Ihrer Valkey- oder Redis

OSS-Instanz aus replizieren kann und beide synchron sind. Wenn Sie die Migration weiterhin stoppen möchten, können Sie den vorhergehenden Befehl mit der Option `–force` ausführen.

- Möglicherweise erhalten Sie eine Ereignismeldung, wenn einer der ElastiCache Knoten ausgetauscht wird. Sie können den Schritt zum Abschließen der Migration wiederholen, nachdem der Austausch abgeschlossen wurde.

Durchführen einer Online-Datenmigration mithilfe der Konsole

Sie können die verwendete AWS-Managementkonsole, um Ihre Daten von Ihrem Cluster auf Ihren Valkey- oder Redis OSS-Cluster zu migrieren.

So führen Sie eine Online-Datenmigration mit der Konsole durch

1. Melden Sie sich bei der Konsole an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Erstellen Sie entweder einen neuen Valkey- oder Redis OSS-Cluster oder wählen Sie einen vorhandenen Cluster aus. Stellen Sie sicher, dass der Cluster die folgenden Anforderungen erfüllt:
 - Ihre Engine-Version sollte Valkey 7.2 und höher oder Redis OSS 5.0.6 oder höher sein.
 - In Ihrem Cluster sollte AUTH nicht aktiviert sein.
 - Die Konfiguration `protected-mode` sollte auf `eingestellt` sein.
 - Wenn Sie eine `bind` Konfiguration in Ihrer Valkey- oder Redis OSS-Konfiguration haben, sollte diese aktualisiert werden, um Anfragen von ElastiCache Knoten zuzulassen.
 - Die Anzahl der Datenbanken sollte zwischen dem ElastiCache Knoten und Ihrem Valkey- oder Redis OSS-Cluster gleich sein. Dieser Wert wird `databases` in der Engine-Konfiguration festgelegt.
 - Valkey- oder Redis OSS-Befehle, die Datenänderungen vornehmen, sollten nicht umbenannt werden, damit die Daten erfolgreich repliziert werden können.
 - Um die Daten aus Ihrem Valkey- oder Redis OSS-Cluster zu replizieren ElastiCache, stellen Sie sicher, dass genügend CPU und Arbeitsspeicher vorhanden sind, um diese zusätzliche Last zu bewältigen. Diese Last stammt aus der RDB-Datei, die von Ihrem Valkey- oder Redis-OSS-Cluster erstellt und über das Netzwerk an den Knoten übertragen wurde. ElastiCache
 - Der Cluster hat den Status `available` (Verfügbar).
3. Wählen Sie bei ausgewähltem Cluster Migrieren von Daten von Endpoint für Aktionen aus.

4. Geben Sie im Dialogfeld Daten vom Endpunkt migrieren die IP-Adresse und den Port ein, an dem Ihr Valkey- oder Redis OSS-Cluster verfügbar ist.

 Important

Die IP-Adresse muss exakt übereinstimmen. Wenn Sie die Adresse falsch eingeben, schlägt die Migration fehl.

5. Wählen Sie Start Migration (Migration starten) aus.

Sobald der Cluster die Migration beginnt, wechselt sein Status zu Modifying (Änderung läuft) und dann zu Migrating (Migration läuft).

6. Überwachen Sie den Migrationsfortschritt, indem Sie im Navigationsbereich Events (Ereignisse) auswählen.

Sie können die laufende Migration jederzeit stoppen. Wählen Sie zu diesem Zweck den Cluster und anschließend Stop Data Migration (Datenmigration stoppen) für Actions (Aktionen) aus. Der Cluster wechselt dann zum Status Available (Verfügbar).

Wenn die Migration erfolgreich verläuft, wechselt der Cluster zum Status Available (Verfügbar) und das Ereignisprotokoll enthält folgenden Eintrag:

Migration operation succeeded for replication group *ElastiCacheClusterName*.

Wenn die Migration fehlschlägt, wechselt der Cluster zum Status Available (Verfügbar) und das Ereignisprotokoll enthält folgenden Eintrag:

Migration operation failed for replication group *ElastiCacheClusterName*.

Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache

Sie können Ihren ElastiCache Clustern zusätzliche Skalierbarkeit und Zuverlässigkeit bieten, indem Sie Regionen und Availability Zones mithilfe des entsprechenden Endpunkts festlegen.

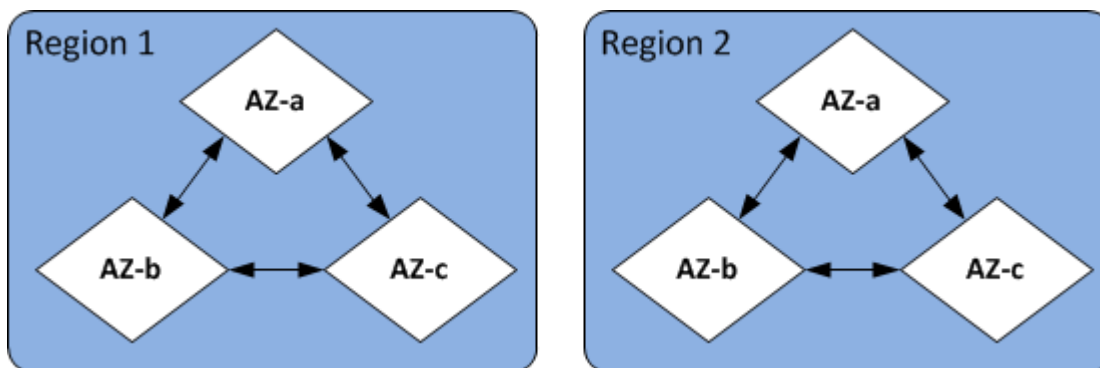
AWS Cloud-Computing-Ressourcen sind in hochverfügbaren Rechenzentrumseinrichtungen untergebracht. Die Einrichtungen dieser Rechenzentren befinden sich an verschiedenen Standorten, um die Skalierbarkeit und Zuverlässigkeit zu erhöhen. Diese Standorte sind nach Regionen und Availability Zones kategorisiert.

AWS Die Regionen sind groß und weit über verschiedene geografische Standorte verteilt. Availability Zones sind unterschiedliche Standorte innerhalb einer AWS Region, die so konzipiert sind, dass sie von Ausfällen in anderen Availability Zones isoliert sind. Sie bieten kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben AWS Region.

⚠ Important

Jede Region ist komplett eigenständig. Jede ElastiCache Aktivität, die Sie initiieren (z. B. das Erstellen von Clustern), wird nur in Ihrer aktuellen Standardregion ausgeführt.

Wenn Sie einen Cluster in einer bestimmten Region erstellen oder damit arbeiten möchten, müssen Sie den entsprechenden regionalen Service-Endpunkt wählen. Informationen zu Service-Endpunkten finden Sie unter [Unterstützte Regionen und Endpunkte](#).



Regionen und Availability Zones

Themen

- [Überlegungen zur Availability Zone bei Memcached](#)
- [Lokalisieren Ihrer Knoten](#)
- [Unterstützte Regionen und Endpunkte](#)
- [Verwenden von lokalen Zonen mit ElastiCache](#)
- [Outposts verwenden mit ElastiCache](#)

Überlegungen zur Availability Zone bei Memcached

Die Verteilung Ihrer Memcached-Knoten auf mehrere Availability Zones innerhalb einer Region schützt Sie vor den Auswirkungen eines katastrophalen Ausfalls, z. B. eines Stromausfalls innerhalb einer Availability Zone.

Serverless-Caching

ElastiCache Beim serverlosen Caching wird ein hochverfügbarer Cache erstellt, der sich über mehrere Availability Zones erstreckt. Sie können bei der Erstellung Ihres serverlosen Clusters Subnetze aus verschiedenen Availability Zones und derselben VPC angeben oder ElastiCache Subnetze automatisch aus Ihrer Standard-VPC auswählen.

Entwerfen Sie Ihren eigenen Cluster für Memcached ElastiCache

Ein Memcached-Cluster kann bis zu 300 Knoten haben. Wenn Sie Knoten erstellen oder zu Ihrem Memcached-Cluster hinzufügen, können Sie eine einzige Availability Zone für all Ihre Knoten angeben, die Auswahl einer einzigen Availability Zone für alle Ihre Knoten zulassen ElastiCache , die Availability Zones für jeden Knoten angeben oder die Auswahl einer Availability Zone für jeden Knoten ermöglichen ElastiCache . Wenn Sie neue Knoten zu einem bestehenden Memcached-Cluster hinzufügen, können diese in verschiedenen Availability Zones erstellt werden. Sobald ein Cache-Knoten erstellt wurde, kann seine Availability Zone nicht mehr geändert werden.

Wenn Sie möchten, dass die Knoten eines Clusters in einem einzelnen Availability Zone-Cluster auf mehrere Availability Zones verteilt sind, ElastiCache können Sie neue Knoten in den verschiedenen Availability Zones erstellen. Anschließend können Sie einige oder alle ursprünglichen Cache-Knoten löschen. Wir empfehlen diesen Ansatz.

Migration von Memcached-Knoten aus einer einzelnen Availability Zone in mehrere Availability Zones

1. Ändern Sie Ihren Cluster, indem Sie neue Cache-Knoten in den Availability Zones erstellen, in denen Sie sie benötigen. Führen Sie in Ihrer Anforderung folgende Schritte aus:
 - Setzen Sie AZMode (CLI: `- -az-mode`) zu `cross-az`.
 - Setzen Sie NumCacheNodes (CLI: `- -num-cache-nodes`) auf die Anzahl der aktuell aktiven Cache-Knoten plus die Anzahl der neuen Cache-Knoten, die Sie erstellen möchten.
 - Setzen Sie NewAvailabilityZones (CLI: `- -new-availability-zones`) in eine Liste der Zonen, in denen Sie die neuen Cache-Knoten erstellen wollen. Um die Availability Zone für jeden neuen Knoten ElastiCache bestimmen zu lassen, geben Sie keine Liste an.
 - Setzen Sie ApplyImmediately (CLI: `- -apply-immediately`) auf `true`.

 Note

Wenn Sie keine automatische Erkennung verwenden, stellen Sie sicher, dass Sie Ihre Client-Anwendung mit den neuen Cache-Knoten-Endpunkten aktualisieren.

Bevor Sie mit dem nächsten Schritt fortfahren, müssen Sie sicherstellen, dass die Memcached-Knoten vollständig erstellt und verfügbar sind.

2. Ändern Sie den Cluster, indem Sie die Knoten entfernen, die Sie in der ursprünglichen Availability Zone nicht mehr benötigen. Führen Sie in Ihrer Anforderung folgende Schritte aus:
 - Setzen Sie NumCacheNodes (CLI: `- -num-cache-nodes`) auf die Anzahl der aktiven Cache-Knoten, die Sie nach der Anwendung dieser Änderung möchten.
 - Setzen Sie CacheNodeIdsToRemove (CLI: `- -nodes-to-remove`) in eine Liste der Cache-Knoten, die Sie aus dem Cluster entfernen möchten.

Die Anzahl der IDs aufgeführten Cache-Knoten muss der Anzahl der derzeit aktiven Knoten abzüglich des Werts in `entsprechenNumCacheNodes`.

- (Optional) Setzen Sie ApplyImmediately (CLI: `- -apply-immediately`) auf `true`.

Wenn Sie ApplyImmediately (CLI: `- -apply-immediately`) nicht auf `true` gesetzt ist, werden die Knotenlöschungen in Ihrem nächsten Wartungsfenster stattfinden.

Lokalisieren Ihrer Knoten

Amazon ElastiCache unterstützt die Lokalisierung aller Knoten eines Clusters in einer oder mehreren Availability Zones (AZs). Wenn Sie sich außerdem dafür entscheiden, Ihre Knoten in mehreren zu lokalisieren AZs (empfohlen ElastiCache), können Sie entweder die AZ für jeden Knoten auswählen oder zulassenElastiCache , dass sie für Sie ausgewählt werden.

Indem Sie die Knoten an verschiedenen Orten platzieren AZs, verhindern Sie, dass ein Ausfall, z. B. ein Stromausfall, in einer AZ zum Ausfall Ihres gesamten Systems führt. Tests haben gezeigt, dass es keinen signifikanten Latenzunterschied zwischen der Lokalisierung aller Knoten in einer AZ oder deren Verteilung auf mehrere AZs gibt.

Sie können eine AZ für jeden Knoten angeben, wenn Sie einen Cluster erstellen, oder indem Sie Knoten hinzufügen, wenn Sie einen vorhandenen Cluster ändern. Wenn Sie bei der Erstellung eines Clusters für jeden Knoten eine AZ angeben, muss die AZ in dieser Subnetzgruppe verfügbar sein. Weitere Informationen finden Sie hier:

- [Einen Cluster für Memcached erstellen](#)
- [Einen Cluster für Valkey oder Redis OSS erstellen](#)
- [Einen ElastiCache Cluster ändern](#)
- [Hinzufügen von Knoten zu einem ElastiCache Cluster](#)

Unterstützte Regionen und Endpunkte

Amazon ElastiCache ist in mehreren AWS Regionen verfügbar. Das bedeutet, dass Sie ElastiCache Cluster an Standorten starten können, die Ihren Anforderungen entsprechen. Sie können beispielsweise in der AWS Region starten, die Ihren Kunden am nächsten ist, oder in einer bestimmten AWS Region, um bestimmte gesetzliche Anforderungen zu erfüllen.

Jede -Region ist darauf ausgelegt, vollständig von den anderen -Regionen getrennt zu sein. Innerhalb jeder Region gibt es mehrere Availability Zones (AZ). ElastiCache Serverlose Caches replizieren Daten automatisch über mehrere Availability Zones hinweg (außer wenn Daten in zwei Availability Zones repliziert werden)us-west-1, um eine hohe Verfügbarkeit zu gewährleisten. Beim Entwerfen Ihres eigenen ElastiCache Clusters können Sie wählen, ob Sie Ihre Knoten auf unterschiedlichen Wegen starten möchten, um Fehlertoleranz AZs zu erreichen. Weitere Informationen zu Regionen und Availability Zones erhalten Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#) oben in diesem Thema.

Regionen, in denen ElastiCache es unterstützt wird

Regionsname/Region	Endpunkt	Protokoll	
Region USA Ost (Ohio) us-east-2	elasticache.us-east-2.amazonaws.com	HTTPS	
Region USA Ost (Nord-Virginia) us-east-1	elasticache.us-east-1.amazonaws.com	HTTPS	
Region USA West (Nordkalifornien) us-west-1	elasticache.us-west-1.amazonaws.com	HTTPS	
Region USA West (Oregon) us-west-2	elasticache.us-west-2.amazonaws.com	HTTPS	
Region Kanada (Zentral) ca-central-1	elasticache.ca-central-1.amazonaws.com	HTTPS	
Region Kanada (West) ca-west-1	elasticache.ca-west-1.amazonaws.com	HTTPS	
Asien-Pazifik (Jakarta) ap-southeast-3	elasticache.ap-southeast-3.amazonaws.com	HTTPS	
Region Asien-Pazifik (Mumbai)	elasticache.ap-south-1.amazonaws.com	HTTPS	

Regionsname/Region	Endpunkt	Protokoll	
ap-south-1			
Region Asien-Pazifik (Hyderabad)	elasticache.ap-south-2.amazonaws.com	HTTPS	
ap-south-2			
Region Asien-Pazifik (Tokio)	elasticache.ap-northeast-1.amazonaws.com	HTTPS	
ap-northeast-1			
Region Asien-Pazifik (Seoul)	elasticache.ap-northeast-2.amazonaws.com	HTTPS	
ap-northeast-2			
Region Asien-Pazifik (Osaka)	elasticache.ap-northeast-3.amazonaws.com	HTTPS	
ap-northeast-3			
Region Asien-Pazifik (Singapur)	elasticache.ap-southeast-1.amazonaws.com	HTTPS	
ap-southeast-1			
Region Asien-Pazifik (Sydney)	elasticache.ap-southeast-2.amazonaws.com	HTTPS	
ap-southeast-2			
Region Europa (Frankfurt)	elasticache.eu-central-1.amazonaws.com	HTTPS	
eu-central-1			

Regionsname/Region	Endpunkt	Protokoll	
Region Europa (Zürich) eu-central-2	elasticache.eu- central-2.am azonaws.com	HTTPS	
Region Europa (Stockholm) eu-north-1	elasticache.eu- north-1.amaz onaws.com	HTTPS	
Region Naher Osten (Bahrain) me-south-1	elasticache.me- south-1.amaz onaws.com	HTTPS	
Region „Naher Osten (VAE)“ me-central-1	elasticache.me- central-1.am azonaws.com	HTTPS	
Region Europa (Irland) eu-west-1	elasticache.eu- west-1.amazo naws.com	HTTPS	
Region Europa (London) eu-west-2	elasticache.eu- west-2.amazo naws.com	HTTPS	
Region Europa (Paris) eu-west-3	elasticache.eu- west-3.amazo naws.com	HTTPS	
Region Europa (Mailand) eu-south-1	elasticache.eu- south-1.amaz onaws.com	HTTPS	

Regionsname/Region	Endpunkt	Protokoll	
Region Europa (Spanien) eu-south-2	elasticache.eu- south-2.amaz onaws.com	HTTPS	
Region Südamerika (São Paulo) sa-east-1	elasticache.sa- east-1.amazo naws.com	HTTPS	
Region China (Peking) cn-north-1	elasticache.cn- north-1.amaz onaws.com.cn	HTTPS	
Region China (Ningxia) cn-northwest-1	elasticache.cn- northwest-1. amazonaws .com.cn	HTTPS	
Region Asien-Pazifik (Hongkong) ap-east-1	elasticache.ap- east-1.amazo naws.com	HTTPS	
Region Afrika (Kapstadt) af-south-1	elasticache.af- south-1.amaz onaws.com	HTTPS	
Region Israel (Tel Aviv) il-central-1	elasticache.il- central-1.am azonaws.com	HTTPS	

Regionsname/Region	Endpunkt	Protokoll
AWS GovCloud (US-West) us-gov-west-1	elasticache.us-gov-west-1.amazonaws.com	HTTPS
AWS GovCloud (US-Ost) us-gov-east-1	elasticache.us-gov-east-1.amazonaws.com	HTTPS

Informationen zur Nutzung von AWS GovCloud (US) mit ElastiCache finden Sie unter [Dienste in der Region AWS GovCloud \(USA\): ElastiCache](#).

Einige Regionen unterstützen eine Teilmenge von Knotentypen. Eine Tabelle der unterstützten Knotentypen nach AWS Regionen finden Sie unter [Unterstützte Knotentypen nach AWS-Region](#).

Die meisten Regionen unterstützen die Einrichtung einer privaten Verbindung zwischen Ihrer VPC und Ihren ElastiCache API-Endpunkten, indem Sie einen VPC-Schnittstellen-Endpunkt erstellen. AWS PrivateLink Weitere Informationen finden Sie unter [ElastiCache API- und Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#).

Eine Tabelle mit AWS Produkten und Dienstleistungen nach Regionen finden Sie unter [Produkte und Dienstleistungen](#) nach Regionen.

Verwenden von lokalen Zonen mit ElastiCache

Eine lokale Zone ist eine Erweiterung einer AWS Region, die sich geografisch in der Nähe Ihrer Benutzer befindet. Sie können jede Virtual Private Cloud (VPC) von einer übergeordneten AWS Region auf eine Local Zones erweitern, indem Sie ein neues Subnetz erstellen und es der lokalen Zone zuweisen. Wenn Sie ein Subnetz in einer lokalen Zone erstellen, wird Ihre VPC auf diese lokale Zone erweitert. Das Subnetz in der lokalen Zone funktioniert genauso wie andere Subnetze in Ihrer VPC.

Mithilfe von Local Zones können Sie Ressourcen wie einen ElastiCache Cluster an mehreren Standorten in der Nähe Ihrer Benutzer platzieren.

Wenn Sie einen ElastiCache Cluster erstellen, können Sie ein Subnetz in einer lokalen Zone auswählen. Local Zones haben ihre eigenen Verbindungen mit dem Internet und unterstützen Direct Connect. Daher können Ressourcen, die in einer lokalen Zone erstellt wurden, von lokalen Benutzern mit Kommunikationen mit sehr geringer Latenz genutzt werden. Weitere Informationen finden Sie unter [AWS -Local-Zones](#).

Eine lokale Zone wird beispielsweise us-west-2-lax-1a durch einen AWS Regionalcode gefolgt von einer Kennung dargestellt, die den Standort angibt.

Zu diesem Zeitpunkt sind die verfügbaren Local Zones us-west-2-lax-1a und us-west-2-lax-1b.

Die folgenden Einschränkungen gelten ElastiCache für Local Zones:

- Globale Datenspeicher werden nicht unterstützt.
- Online-Migration wird nicht unterstützt.
- Die folgenden Knotentypen werden von Local Zones zur Zeit unterstützt:
 - Aktuelle Generation:

M5-Knotentypen: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`, `cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

R5-Knotentypen: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

T3-Knotentypen: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Aktivieren einer Local Zone

1. Aktivieren Sie die lokale Zone in der EC2 Amazon-Konsole.

Weitere Informationen finden Sie unter [Enabling Local Zones](#) im EC2 Amazon-Benutzerhandbuch.

2. Erstellen Sie ein Subnetz in der Local Zone.

Weitere Informationen finden Sie unter [Erstellen eines Subnetzes in Ihrer VPC](#) im Amazon-VPC-Benutzerhandbuch.

3. Erstellen Sie eine ElastiCache Subnetzgruppe in der lokalen Zone.

Wenn Sie eine ElastiCache Subnetzgruppe erstellen, wählen Sie die Availability Zone-Gruppe für die Lokale Zone aus.

Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

4. Erstellen Sie einen Cluster ElastiCache für Memcached, der das ElastiCache Subnetz in der lokalen Zone verwendet.

Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

5. Erstellen Sie einen OSS-Cluster ElastiCache für Redis, der das ElastiCache Subnetz in der lokalen Zone verwendet. Weitere Informationen finden Sie in den folgenden Themen:

- [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Outposts verwenden mit ElastiCache

Sie können AWS Outposts mit ElastiCache verwenden. Outposts ist ein vollständig verwalteter Service, der AWS Infrastruktur APIs, Dienste und Tools auf Kundenstandorte ausdehnt. Durch den lokalen Zugriff auf die AWS verwaltete Infrastruktur ermöglicht AWS Outposts seinen Kunden, Anwendungen vor Ort mit denselben Programmierschnittstellen wie in AWS Regionen zu erstellen und auszuführen und gleichzeitig lokale Rechen- und Speicherressourcen für geringere Latenz und lokale Datenverarbeitungsanforderungen zu nutzen. Ein Outpost ist ein Pool von AWS Rechen- und Speicherkapazität, der am Standort eines Kunden bereitgestellt wird. AWS betreibt, überwacht und verwaltet diese Kapazität als Teil einer AWS Region. Sie können Subnetze in Ihrem Outpost erstellen und diese angeben, wenn Sie AWS Ressourcen wie ElastiCache Cluster erstellen.

Note

In dieser Version gelten folgende Einschränkungen:

- ElastiCache for Outposts unterstützt nur M5- und R5-Knotenfamilien.
- Multi-AZ (Cross-Outpost-Replikation) wird nicht unterstützt.
- Live-Migration wird nicht unterstützt.
- Lokale Snapshots werden nicht unterstützt.
- Engine-Logs und Slow-Logs können nicht aktiviert werden.
- ElastiCache on Outposts unterstützt CoIP nicht.

- ElastiCache for Outposts wird in den folgenden Regionen nicht unterstützt: cn-north-1, cn-northwest-1 und ap-northeast-3.

Outposts mit der ElastiCache Konsole verwenden

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich entweder Valkey-Caches, Redis OSS-Caches oder Memcached-Caches aus.
3. Wenn Sie Valkey-Caches ausgewählt haben, wählen Sie Valkey-Cache erstellen aus. Wenn Sie Redis OSS-Caches ausgewählt haben, wählen Sie Redis OSS-Cache erstellen aus. Wenn Sie Memcached-Caches ausgewählt haben, wählen Sie Memcached-Cache erstellen aus.
4. Wählen Sie unter Cluster-Einstellungen die Optionen Eigenen Cache entwerfen und Cluster-Cache aus. Lassen Sie den Clustermodus auf Deaktiviert eingestellt. Erstellen Sie dann einen Namen und eine optionale Beschreibung für den Cache.
5. Wählen Sie als Standort die Option Lokal aus.
6. Im Bereich On-premises sehen Sie das Feld Outpost ID. Geben Sie die ID ein, unter der der Cluster ausgeführt werden soll.

Alle weiteren Einstellungen unter Cluster-Einstellungen können die Standardeinstellungen beibehalten.

7. Wählen Sie unter Konnektivität die Option Neue Subnetzgruppe erstellen aus und geben Sie die VPC-ID ein. Belassen Sie den Rest als Standard und wählen Sie Weiter aus.

Konfiguration von lokalen Optionen

Sie können entweder einen verfügbaren Outpost auswählen, um Ihren Cluster hinzuzufügen, oder, falls keine Outposts verfügbar sind, mithilfe der folgenden Schritte einen neuen erstellen:

Unter Lokale Optionen:

1. Unter Valkey-Einstellungen, Redis OSS-Einstellungen oder Memcached-Einstellungen, je nach Ihrer bevorzugten Engine:
 - a. Name: Geben Sie einen Namen für den Cluster ein

- b. Beschreibung: Geben Sie eine Beschreibung für den Cluster ein.
- c. Kompatibilität der Engine-Version: Die Engine-Version basiert auf der AWS Outpost-Region
- d. Port: Akzeptieren Sie für Valkey oder Redis OSS den Standardport 6379. Akzeptieren Sie für Memcached den Standardport 11211. Wenn Sie lieber einen anderen Port verwenden möchten, geben Sie die Portnummer ein.
- e. Parametergruppe: Verwenden Sie das Dropdown-Menü, um eine Standard- oder benutzerdefinierte Parametergruppe auszuwählen.
- f. Knoten-Typ: Die verfügbaren Instances basieren auf der Verfügbarkeit von Outposts. Wenn Sie Valkey oder Redis OSS verwenden, unterstützt Porting Assistant for .NET for Outposts nur die M5- und R5-Knotenfamilien. Wählen Sie in der Dropdown-Liste die Option Outposts und wählen Sie dann einen verfügbaren Knotentyp aus, den Sie für diesen Cluster verwenden möchten. Wählen Sie dann Speichern aus.
- g. Anzahl der Replikate: Geben Sie die Anzahl der Lesereplikate ein, die für diese Replikationsgruppe erstellt werden sollen. Sie müssen mindestens über ein und nicht mehr als fünf Lesereplikate verfügen. Der Standardwert lautet 2.

Die automatisch generierten Namen der Lesereplikate folgen dem gleichen Muster wie der Name des primären Clusters, wobei ein Bindestrich und eine sequenzielle dreistellige Zahl am Ende hinzugefügt werden, beginnend mit -002. Wenn Ihre Replikationsgruppe beispielsweise den Namen MyGroup hat, dann würden die Namen der Secondaries MyGroup-002, MyGroup-003, MyGroup-004, MyGroup-005, MyGroup-006 lauten.

2. Unter Konnektivität:

- a. Subnetzgruppe: Wählen Sie in der Liste die Option Neu erstellen aus.
 - Name: Geben Sie einen Namen für die Subnetzgruppe ein
 - Beschreibung: Geben Sie eine Beschreibung für die Subnetzgruppe ein
 - VPC-ID: Die VPC-ID sollte mit der Outpost-VPC übereinstimmen. Wenn Sie eine VPC auswählen, die kein Subnetz IDs in den Outposts hat, wird die Liste leer zurückgegeben.
 - Availability Zone oder Outpost: Wählen Sie den Outpost aus, den Sie verwenden.
 - Subnetz-ID: Wählen Sie eine Subnetz-ID aus, die für den Outpost verfügbar ist. Wenn kein Subnetz IDs verfügbar ist, müssen Sie es erstellen. Weitere Informationen finden Sie unter [Ein Subnetz erstellen](#).
- b. Wählen Sie Erstellen aus.

Anzeige der Details des Outpost-Clusters

Wählen Sie auf der Listenseite einen Cluster aus, der zu einem AWS Outpost gehört, und beachten Sie Folgendes, wenn Sie sich die Cluster-Details ansehen:

- **Availability Zone:** Dies stellt den Outpost dar, wobei ein ARN (Amazon Resource Name) und die AWS Ressourcennummer verwendet werden.
- **Name des Außenpostens:** Der Name des Außenpostens AWS.

Outposts mit der AWS CLI verwenden

Sie können die AWS Command Line Interface(AWS CLI) verwenden, um mehrere AWS Dienste von der Befehlszeile aus zu steuern und sie mithilfe von Skripten zu automatisieren. Sie können die AWS CLI für (einmalige) Ad-hoc-Operationen verwenden.

Herunterladen und Konfigurieren der AWS CLI

Das AWS CLI läuft unter Windows, MacOS oder Linux. Gehen Sie folgendermaßen vor, um sie herunterzuladen und zu konfigurieren.

So laden Sie den CLI herunter, installieren und konfigurieren ihn

1. Laden Sie die AWS CLI von der [AWS Command Line Interface-Webseite](#) herunter.
2. Folgen Sie den Anweisungen zur [Installation der AWS CLI](#) und [zur Konfiguration der AWS CLI](#) im AWS Command Line Interface Benutzerhandbuch.

Verwenden der AWS CLI mit Outposts

Verwenden Sie den folgenden CLI-Vorgang, um einen Cluster zu erstellen, der Outposts verwendet:

- [create-cache-cluster](#)— Bei dieser Operation akzeptiert der `outpost-mode` Parameter einen Wert, der angibt, ob die Knoten im Cluster in einem einzelnen Outpost oder in mehreren Outposts erstellt werden.

Note

Zurzeit wird nur der `single-outpost`-Modus unterstützt.

```
aws elasticache create-cache-cluster \
  --cache-cluster-id cluster id \
  --outpost-mode single-outpost \
```

Arbeiten mit ElastiCache

In diesem Abschnitt finden Sie Einzelheiten zur Verwaltung der verschiedenen Komponenten Ihrer ElastiCache Implementierung.

Themen

- [Snapshot und Wiederherstellung](#)
- [Motorversionen und Aufrüstung in ElastiCache](#)
- [ElastiCache Best Practices und Caching-Strategien](#)
- [Verwaltung Ihres knotenbasierten Clusters in ElastiCache](#)
- [Automatisches Verbinden einer EC2 Instanz und eines ElastiCache Caches](#)
- [Skalierung ElastiCache](#)
- [Erste Schritte mit Bloom-Filtern](#)
- [Erste Schritte mit Watch in Serverless](#)
- [Erste Schritte mit Vector Search](#)
- [Erste Schritte mit JSON für Valkey und Redis OSS](#)
- [Verschlagworten Sie Ihre Ressourcen ElastiCache](#)
- [Verwenden des Amazon ElastiCache Well-Architected-Objektivs](#)
- [Allgemeine Schritte zur Fehlerbehebung und bewährte Methoden mit ElastiCache](#)

Snapshot und Wiederherstellung

ElastiCache Amazon-Caches, auf denen Valkey, Redis OSS oder Serverless Memcached ausgeführt werden, können ihre Daten sichern, indem sie einen Snapshot erstellen. Sie können das Backup verwenden, um einen Cache wiederherzustellen oder Daten in einem neuen Cache durch Seeding zu speichern. Ein Backup besteht aus den Metadaten des Caches zusammen mit allen Daten im Cache. Alle Sicherungen werden in Amazon Simple Storage Service (Amazon S3) geschrieben, der einen dauerhaften Speicher bereitstellt. Sie können Ihre Daten jederzeit wiederherstellen, indem Sie einen neuen Valkey-, Redis OSS- oder Serverless Memcached-Cache erstellen und ihn mit Daten aus einem Backup füllen. Mit ElastiCache können Sie Backups mithilfe der AWS-Managementkonsole, AWS Command Line Interface (AWS CLI) und der API verwalten. ElastiCache

Wenn Sie vorhaben, den Cache zu löschen, und es wichtig ist, die Daten beizubehalten, können Sie eine zusätzliche Vorsichtsmaßnahme ergreifen. Erstellen Sie dazu zuerst ein manuelles Backup,

überprüfen Sie, dass der Status `available` lautet und löschen Sie dann den Cache. Dadurch wird sichergestellt, dass die Cache-Daten weiterhin verfügbar sind, falls das Backup fehlschlagen sollte. Sie können nach den oben beschriebenen bewährten Methoden erneut versuchen, eine Sicherung anzufertigen.

Themen

- [Sicherungseinschränkungen](#)
- [Auswirkungen von Backups knotenbasierter Cluster auf die Leistung](#)
- [Planen automatischer Backups](#)
- [Erstellen manueller Backups](#)
- [Erstellen einer endgültigen Sicherung](#)
- [Beschreiben von Sicherungen](#)
- [Kopieren eines Backups](#)
- [Exportieren einer Sicherung](#)
- [Wiederherstellen aus einem Backup in einen neuen Cache](#)
- [Löschen einer Sicherung](#)
- [Markieren von Sicherungen](#)
- [Tutorial: Seeding eines neuen knotenbasierten Clusters mit einem extern erstellten Backup](#)

Sicherungseinschränkungen

Berücksichtigen Sie die folgenden Einschränkungen bei der Planung und Erstellung von Sicherungen:

- Backup und Wiederherstellung werden nur für Caches unterstützt, die auf Valkey, Redis OSS oder Serverless Memcached ausgeführt werden.
- Bei Valkey- oder Redis OSS-Clustern (Clustermodus deaktiviert) werden Sicherung und Wiederherstellung auf Knoten nicht unterstützt. `cache.t1.micro` Alle anderen Cache-Knotentypen werden unterstützt.
- Bei Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert) werden Sicherung und Wiederherstellung für alle Knotentypen unterstützt.
- In einem zusammenhängenden Zeitraum von 24 Stunden können Sie nicht mehr als 24 manuelle Backups pro serverlosem Cache erstellen. Für knotenbasierte Valkey- und Redis OSS-Cluster können Sie nicht mehr als 20 manuelle Backups pro Knoten im Cluster erstellen.

- Valkey oder Redis OSS (Clustermodus aktiviert) unterstützen nur das Erstellen von Backups auf Clusterebene (für die API oder CLI, die Replikationsgruppenebene). Valkey oder Redis OSS (Clustermodus aktiviert) unterstützt nicht das Erstellen von Backups auf Shard-Ebene (für die API oder CLI, auf Knotengruppenebene).
- Während des Backup-Vorgangs können Sie keine anderen API- oder CLI-Operationen im serverlosen Cache ausführen. Sie können während des Backups API- oder CLI-Operationen auf einem knotenbasierten Cluster ausführen.
- Wenn Sie Valkey- oder Redis OSS-Caches mit Data Tiering verwenden, können Sie kein Backup nach Amazon S3 exportieren.
- Sie können ein Backup eines Clusters mit dem R6gd-Knotentyp nur für Cluster wiederherstellen, die den R6gd-Knotentyp verwenden.

Auswirkungen von Backups knotenbasierter Cluster auf die Leistung

Backups auf Serverless-Caches sind für die Anwendung transparent, ohne dass sich dies auf die Leistung auswirkt. Bei der Erstellung von Backups für knotenbasierte Cluster kann es jedoch je nach verfügbarem reservierten Speicher zu Leistungseinbußen kommen. Backups für knotenbasierte Cluster sind ElastiCache für Memcached nicht verfügbar, aber für Redis OSS. ElastiCache

Im Folgenden finden Sie Richtlinien zur Verbesserung der Backup-Leistung für knotenbasierte Cluster.

- `reserved-memory-percent` Parameter festlegen — Um übermäßiges Paging zu vermeiden, empfehlen wir Ihnen, den Parameter festzulegen. `reserved-memory-percent` Dieser Parameter verhindert, dass Valkey und Redis OSS den gesamten verfügbaren Speicher des Knotens verbrauchen, und kann dazu beitragen, den Umfang des Paging zu reduzieren. Die Leistung lässt sich möglicherweise einfach durch Verwenden eines größeren Knotens verbessern. Weitere Hinweise zum reservierten Speicher und zu den Parametern finden Sie unter [reserved-memory-percent](#) [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#)
- Backups aus einer Read Replica erstellen — Wenn Sie Valkey oder Redis OSS in einer Knotengruppe mit mehr als einem Knoten ausführen, können Sie ein Backup vom primären Knoten oder einer der Read Replica erstellen. Aufgrund der während BGSAVE erforderlichen Systemressourcen wird empfohlen, Sicherungen von einer der Read Replicas zu erstellen. Während die Sicherung von der Replikation erstellt wird, wirken sich die BGSAVE-

Ressourcenanforderungen nicht auf den primären Knoten aus. Der primäre Knoten kann weiterhin Anfragen verarbeiten, ohne langsamer zu werden.

Siehe dazu [Erstellen einer manuellen Sicherung \(Konsole\)](#) und wählen Sie im Fenster Sicherung erstellen im Feld Clustername ein Replikat anstelle des standardmäßigen Primärknotens aus.

Wenn Sie eine Replikationsgruppe löschen und ein letztes Backup anfordern, wird das Backup ElastiCache immer vom primären Knoten erstellt. Dadurch wird sichergestellt, dass Sie die neuesten Valkey- oder Redis-OSS-Daten erfassen, bevor die Replikationsgruppe gelöscht wird.

Planen automatischer Backups

Sie können automatische Backups für jeden serverlosen Cache oder knotenbasierten Cluster von Valkey oder Redis OSS aktivieren. Wenn automatische Backups aktiviert sind, ElastiCache erstellt täglich eine Sicherungskopie des Caches. Es gibt keine Auswirkungen auf den Cache und die Änderung erfolgt sofort. Automatische Backups schützen vor Datenverlust. Bei einem Ausfall können Sie einen neuen Cache erstellen, indem Sie Ihre Daten aus dem aktuellen Backup wiederherstellen. Das Ergebnis ist ein warm gestarteter Cache, auf dem Ihre Daten vorab geladen und einsatzbereit sind. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Sie können automatische Backups für jeden Memcached Serverless Cache aktivieren. Wenn automatische Backups aktiviert sind, ElastiCache erstellt täglich eine Sicherungskopie des Caches. Es gibt keine Auswirkungen auf den Cache und die Änderung erfolgt sofort. Automatische Backups schützen vor Datenverlust. Bei einem Ausfall können Sie einen neuen Cache erstellen, indem Sie Ihre Daten aus dem aktuellen Backup wiederherstellen. Das Ergebnis ist ein warm gestarteter Cache, auf dem Ihre Daten vorab geladen und einsatzbereit sind. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Wenn Sie automatische Sicherungen planen, sollten Sie die folgenden Einstellungen vornehmen:

- **Backup-Startzeit** — Die Tageszeit, zu der mit der Erstellung eines Backups ElastiCache begonnen wird. Sie können das Backup-Fenster für jeden beliebigen Zeitpunkt festlegen, zu dem es am sinnvollsten ist. Wenn Sie kein Backup-Fenster angeben, wird automatisch eines ElastiCache zugewiesen.
- **Aufbewahrungsfrist für Sicherungen** – Die Anzahl von Tagen, für die die Sicherung in Amazon S3 aufbewahrt wird. Wenn Sie z. B. als Aufbewahrungsfrist 5 festlegen, dann wird die heute erstellte Sicherung 5 Tage lang aufbewahrt. Bei Ablauf der Aufbewahrungsfrist wird die Sicherung automatisch gelöscht.

Der maximale Aufbewahrungsfrist für Sicherungen ist 35 Tage. Wenn als Backup-Aufbewahrungsfrist „0“ festgelegt wird, werden automatische Backups für den Cache deaktiviert.

Wenn Sie automatische Backups planen, ElastiCache wird mit der Erstellung des Backups begonnen. Sie können das Backup-Fenster für jeden beliebigen Zeitpunkt festlegen, zu dem es am sinnvollsten ist. Wenn Sie kein Backup-Fenster angeben, wird automatisch eines ElastiCache zugewiesen.

Sie können automatische Backups aktivieren oder deaktivieren, wenn Sie entweder einen neuen Cache erstellen oder einen vorhandenen Cache aktualisieren, indem Sie die ElastiCache Konsole, AWS CLI, oder die ElastiCache API verwenden. Für Valkey und Redis OSS erfolgt dies, indem Sie das Kontrollkästchen Automatische Backups aktivieren im Abschnitt Erweiterte Valkey-Einstellungen oder Erweiterte Redis OSS-Einstellungen aktivieren. Für Memcached erfolgt dies, indem Sie im Abschnitt Erweiterte Memcached-Einstellungen das Kontrollkästchen Automatische Backups aktivieren aktivieren aktivieren.

Erstellen manueller Backups

Zusätzlich zu automatischen Sicherungen können Sie jederzeit eine manuelle Sicherung erstellen. Im Gegensatz zu automatischen Sicherungen, die nach einem angegebenen Aufbewahrungszeitraum automatisch gelöscht werden, ist für manuelle Sicherungen kein Aufbewahrungszeitraum vorhanden, nachdem sie automatisch gelöscht werden. Selbst wenn Sie den Cache löschen, werden alle manuellen Backups aus diesem Cache beibehalten. Wenn Sie eine manuelle Sicherung nicht mehr aufbewahren möchten, müssen Sie sie selbst explizit löschen.

Eine manuelle Sicherung kann nicht nur direkt, sondern auch auf eine der folgenden Arten erstellt werden:

- [Kopieren eines Backups](#). Es spielt keine Rolle, ob die Quellsicherung automatisch oder manuell erstellt wurde.
- [Erstellen einer endgültigen Sicherung](#). Erstellen Sie ein Backup unmittelbar vor dem Löschen eines Clusters oder Knotens.

Sie können ein manuelles Backup eines Caches mithilfe der AWS-Managementkonsole AWS CLI, der oder der ElastiCache API erstellen.

Sie können manuelle Backups von Replikaten erstellen, bei denen der Clustermodus aktiviert und der Clustermodus deaktiviert ist.

Erstellen einer manuellen Sicherung (Konsole)

So erstellen Sie ein manuelles Backup eines Caches (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich je nach Ihren Vorlieben Valkey-Caches, Redis OSS-Caches oder Memcached-Caches aus.
3. Wählen Sie das Feld links neben dem Namen des Caches aus, den Sie sichern möchten.
4. Wählen Sie Backup.
5. Geben Sie im Dialog Create Backup einen Namen für Ihre Sicherung im Feld Backup Name ein. Es ist empfehlenswert, dass aus dem Namen der gesicherte Cluster und das Datum sowie die Uhrzeit der Sicherung ersichtlich ist.

Für die Benennung von Clustern gelten die folgenden Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

6. Wählen Sie Create Bucket.

Der Status des Clusters ändert sich in snapshotting.

Erstellen einer manuellen Sicherung (AWS CLI)

Manuelles Backup eines serverlosen Caches mit dem AWS CLI

Um eine manuelle Sicherung eines Caches mit dem zu erstellen AWS CLI, verwenden Sie den `create-serverless-snapshot` AWS CLI Vorgang mit den folgenden Parametern:

- `--serverless-cache-name` – der Name des Serverless-Caches, für den Sie ein Backup erstellen.
- `--serverless-cache-snapshot-name` – der Name des zu erstellenden Snapshots.

Für Linux, macOS oder Unix:

- ```
aws elasticache create-serverless-snapshot \
 --serverless-cache-name CacheName \
 --serverless-cache-snapshot-name bkup-20231127
```

Für Windows:

- ```
aws elasticache create-serverless-snapshot ^
    --serverless-cache-name CacheName ^
    --serverless-cache-snapshot-name bkup-20231127
```

Manuelles Backup eines knotenbasierten Clusters mit dem AWS CLI

Um ein manuelles Backup eines knotenbasierten Clusters mithilfe von zu erstellen AWS CLI, verwenden Sie den `create-snapshot` AWS CLI Vorgang mit den folgenden Parametern:

- `--cache-cluster-id`
 - Wenn der Cluster, den Sie sichern, keine Replikatknoten hat, `--cache-cluster-id` ist dies beispielsweise der Name des Clusters, den Sie sichern. *mycluster*
 - Wenn der von Ihnen gesicherte Cluster über mindestens einen Replikatknoten verfügt, lautet der Name des Knotens im Clusters, den Sie für die Sicherung verwenden möchten `--cache-cluster-id`. Der Name könnte beispielsweise lauten. *mycluster-002*

Verwenden Sie diesen Parameter nur, wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) sichern.

- `--replication-group-id`— Name des Valkey- oder Redis OSS-Clusters (Cluster-Modus aktiviert) (CLI/API: eine Replikationsgruppe), der als Quelle für das Backup verwendet werden soll. Verwenden Sie diesen Parameter, wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) sichern.
- `--snapshot-name` Name des zu erstellenden Snapshots.

Für die Benennung von Clustern gelten die folgenden Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

Beispiel 1: Sicherung eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert), der keine Replikatknoten hat

Mit dem folgenden AWS CLI Vorgang wird das Backup `bkup-20150515` aus dem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) erstellt `myNonClusteredRedis`, der keine Read Replicas enthält.

Für Linux, macOS oder Unix:

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis \  
  --snapshot-name bkup-20150515
```

Für Windows:

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis ^  
  --snapshot-name bkup-20150515
```

Beispiel 2: Sicherung eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) mit Replikatknoten

Der folgende AWS CLI Vorgang erstellt das Backup `bkup-20150515` aus dem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert). `myNonClusteredRedis` Diese Sicherung hat eine oder mehrere Lesereplikate.

Für Linux, macOS oder Unix:

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis-001 \  
  --snapshot-name bkup-20150515
```

Für Windows:

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis-001 ^  
  --snapshot-name bkup-20150515
```

Beispielausgabe: Sicherung eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) mit Replikatknoten

Die Ausgabe der Operation sieht in etwa folgendermaßen aus.

```
{  
  "Snapshot": {  
    "Engine": "redis",  
    "CacheParameterGroupName": "default.redis6.x",  
    "VpcId": "vpc-91280df6",
```

```

    "CacheClusterId": "myNonClusteredRedis-001",
    "SnapshotRetentionLimit": 0,
    "NumCacheNodes": 1,
    "SnapshotName": "bkup-20150515",
    "CacheClusterCreateTime": "2017-01-12T18:59:48.048Z",
    "AutoMinorVersionUpgrade": true,
    "PreferredAvailabilityZone": "us-east-1c",
    "SnapshotStatus": "creating",
    "SnapshotSource": "manual",
    "SnapshotWindow": "08:30-09:30",
    "EngineVersion": "6.0",
    "NodeSnapshots": [
      {
        "CacheSize": "",
        "CacheNodeId": "0001",
        "CacheNodeCreateTime": "2017-01-12T18:59:48.048Z"
      }
    ],
    "CacheSubnetGroupName": "default",
    "Port": 6379,
    "PreferredMaintenanceWindow": "wed:07:30-wed:08:30",
    "CacheNodeType": "cache.m3.2xlarge",
    "DataTiering": "disabled"
  }
}

```

Beispiel 3: Sicherung eines Clusters für Valkey oder Redis OSS (Clustermodus aktiviert)

Der folgende AWS CLI Vorgang erstellt das Backup `bkup-20150515` aus dem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert). `myClusteredRedis` Beachten Sie die Verwendung von `--replication-group-id` anstelle von `--cache-cluster-id` zum Identifizieren der Quelle. Beachten Sie außerdem, dass ElastiCache für die Sicherung der Replikatknoten verwendet wird, sofern vorhanden, und standardmäßig der primäre Knoten verwendet wird, wenn ein Replikatknoten nicht verfügbar ist.

Für Linux, macOS oder Unix:

```

aws elasticache create-snapshot \
  --replication-group-id myClusteredRedis \
  --snapshot-name bkup-20150515

```

Für Windows:

```
aws elasticache create-snapshot ^  
  --replication-group-id myClusteredRedis ^  
  --snapshot-name bkup-20150515
```

Beispielausgabe: Sicherung eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert)

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{  
  "Snapshot": {  
    "Engine": "redis",  
    "CacheParameterGroupName": "default.redis6.x.cluster.on",  
    "VpcId": "vpc-91280df6",  
    "NodeSnapshots": [  
      {  
        "CacheSize": "",  
        "NodeGroupId": "0001"  
      },  
      {  
        "CacheSize": "",  
        "NodeGroupId": "0002"  
      }  
    ],  
    "NumNodeGroups": 2,  
    "SnapshotName": "bkup-20150515",  
    "ReplicationGroupId": "myClusteredRedis",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 1,  
    "AutomaticFailover": "enabled",  
    "SnapshotStatus": "creating",  
    "SnapshotSource": "manual",  
    "SnapshotWindow": "10:00-11:00",  
    "EngineVersion": "6.0",  
    "CacheSubnetGroupName": "default",  
    "ReplicationGroupDescription": "2 shards 2 nodes each",  
    "Port": 6379,  
    "PreferredMaintenanceWindow": "sat:03:30-sat:04:30",  
    "CacheNodeType": "cache.r3.large",  
    "DataTiering": "disabled"  
  }  
}
```

Verwandte Themen

Weitere Informationen finden Sie unter [create-bucket](#) in der AWS CLI-Befehlsreferenz.

Erstellen eines Backups mit CloudFormation

Sie können CloudFormation es verwenden, um mithilfe der Eigenschaften oder eine Sicherungskopie Ihres ElastiCache Redis OSS- oder Valkey-Caches zu erstellen.

`AWS::ElastiCache::ServerlessCache` `AWS::ElastiCache::ReplicationGroup`

Verwenden der Ressource **AWS::ElastiCache::ServerlessCache**

Verwenden Sie dies, um mithilfe der `AWS::ElastiCache::ServerlessCache` Ressource ein Backup zu erstellen:

```
Resources:
    iotCatalog:
        Type: AWS::ElastiCache::ServerlessCache
        Properties:
            ...
            ServerlessCacheName: "your-cache-name"
            Engine: "redis"
            CacheUsageLimits
```

Verwendung der `AWS::ElastiCache::ReplicationGroup` Ressource

Verwenden Sie die `AWS::ElastiCache::ReplicationGroup` Ressource:

```
Resources:
    iotCatalog:
        Type: AWS::ElastiCache::ReplicationGroup
        Properties:
            ...
            ReplicationGroupDescription: "Description of your
replication group"

            Engine: "redis"
            CacheNodeType
            NumCacheClusters
            AutomaticFailoverEnabled
            AtRestEncryptionEnabled
```

Erstellen einer endgültigen Sicherung

Sie können mit der ElastiCache Konsole, der oder der AWS CLI ElastiCache API ein endgültiges Backup erstellen.

Erstellen einer endgültigen Sicherung (Konsole)

Sie können eine endgültige Sicherung erstellen, wenn Sie mithilfe der Konsole einen serverlosen Valkey-, Memcached- oder Redis OSS-Cache oder einen knotenbasierten Valkey- oder Redis OSS-Cluster löschen. ElastiCache

Um beim Löschen eines Caches eine endgültige Sicherung zu erstellen, wählen Sie im Löschdialogfeld unter Backup erstellen die Option Ja aus und geben Sie der Sicherung einen Namen.

Verwandte Themen

- [Mit dem AWS-Managementkonsole](#)
- [Löschen einer Replikationsgruppe \(Konsole\)](#)

Erstellen einer endgültigen Sicherung (AWS CLI)

Sie können ein endgültiges Backup erstellen, wenn Sie einen Cache löschen, indem Sie den AWS CLI

Themen

- [Beim Löschen eines Valkey-Caches, eines Memcached Serverless Caches oder eines Redis OSS-Caches](#)
- [Beim Löschen eines Valkey- oder Redis OSS-Clusters ohne Read Replicas](#)
- [Wenn Sie einen Valkey- oder Redis OSS-Cluster mit Read Replicas löschen](#)

Beim Löschen eines Valkey-Caches, eines Memcached Serverless Caches oder eines Redis OSS-Caches

Verwenden Sie den `delete-serverless-cache` AWS CLI Vorgang mit den folgenden Parametern, um ein endgültiges Backup zu erstellen.

- `--serverless-cache-name` – Name des zu löschenden Clusters.
- `--final-snapshot-name` – Name des Backups.

Der folgende Code erstellt das endgültige Backup `bkup-20231127-final` beim Löschen des Caches `myserverlesscache`.

Für Linux, macOS oder Unix:

```
aws elasticache delete-serverless-cache \  
    --serverless-cache-name myserverlesscache \  
    --final-snapshot-name bkup-20231127-final
```

Für Windows:

```
aws elasticache delete-serverless-cache ^  
    --serverless-cache-name myserverlesscache ^  
    --final-snapshot-name bkup-20231127-final
```

Weitere Informationen finden Sie unter [delete-serverless-cache](#) in der Referenz zum AWS CLI-Befehl.

Beim Löschen eines Valkey- oder Redis OSS-Clusters ohne Read Replicas

Verwenden Sie den Vorgang mit den folgenden Parametern, um ein endgültiges Backup für einen knotenbasierten Cluster ohne Read Replicas zu erstellen. `delete-cache-cluster` AWS CLI

- `--cache-cluster-id` Name des zu löschenden Clusters.
- `--final-snapshot-identifizier` Name der Sicherung.

Der folgende Code erstellt die endgültige Sicherung `bkup-20150515-final` beim Löschen des Clusters `myRedisCluster`.

Für Linux, macOS oder Unix:

```
aws elasticache delete-cache-cluster \  
    --cache-cluster-id myRedisCluster \  
    --final-snapshot-identifizier bkup-20150515-final
```

Für Windows:

```
aws elasticache delete-cache-cluster ^  
    --cache-cluster-id myRedisCluster ^
```

```
--final-snapshot-identifizier bkup-20150515-final
```

Weitere Informationen finden Sie unter [delete-cache-cluster](#) in der Referenz zum AWS CLI-Befehl.

Wenn Sie einen Valkey- oder Redis OSS-Cluster mit Read Replicas löschen

Verwenden Sie den `delete-replication-group` AWS CLI Vorgang mit den folgenden Parametern, um beim Löschen einer Replikationsgruppe ein letztes Backup zu erstellen:

- `--replication-group-id` Name der zu löschenden Replikationsgruppe.
- `--final-snapshot-identifizier` Name der endgültigen Sicherung.

Der folgende Code erstellt die endgültige Sicherung `bkup-20150515-final` beim Löschen der Replikationsgruppe `myReplGroup`.

Für Linux, macOS oder Unix:

```
aws elasticache delete-replication-group \  
    --replication-group-id myReplGroup \  
    --final-snapshot-identifizier bkup-20150515-final
```

Für Windows:

```
aws elasticache delete-replication-group ^  
    --replication-group-id myReplGroup ^  
    --final-snapshot-identifizier bkup-20150515-final
```

Weitere Informationen finden Sie unter [delete-replication-group](#) in der Referenz zum AWS CLI-Befehl.

Beschreiben von Sicherungen

Die folgenden Verfahren zeigen, wie Sie eine Liste Ihrer Sicherungen anzeigen. Wenn Sie möchten, können Sie auch die Details zu einer bestimmten Sicherung anzeigen.

Beschreibung von Backups (Konsole)

Um Backups anzuzeigen, verwenden Sie AWS-Managementkonsole

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Backups aus.
3. Um die Details zu einer bestimmten Sicherung anzuzeigen, aktivieren Sie das Kontrollkästchen links neben dem Namen der betreffenden Sicherung.

Beschreiben von Serverless-Backups (AWS CLI)

Verwenden Sie die CLI-Operation `describe-serverless-cache-snapshots`, um eine Liste von Serverless-Backups und optional Details zu einem bestimmten Backup anzuzeigen.

Beispiele

Die folgende Operation listet mit dem Parameter `--max-records` bis zu 20 zum Konto gehörende Sicherungen auf. Wenn der Parameter `--max-records` weggelassen wird, werden bis zu 50 Sicherungen aufgelistet.

```
aws elasticache describe-serverless-cache-snapshots --max-records 20
```

Die folgende Operation listet mit dem Parameter `--serverless-cache-name` nur die zum Cache `my-cache` gehörenden Backups auf.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-name my-cache
```

Die folgende Operation listet mit dem Parameter `--serverless-cache-snapshot-name` die Details der Sicherung `my-backup` auf.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-snapshot-name my-backup
```

Weitere Informationen finden Sie [describe-serverless-cache-snapshots](#) in der AWS CLI Befehlsreferenz.

Beschreibung knotenbasierter Cluster-Backups ()AWS CLI

Verwenden Sie den `describe-snapshots` CLI-Vorgang, um eine Liste von knotenbasierten Cluster-Backups und optional Details zu einem bestimmten Backup anzuzeigen.

Beispiele

Die folgende Operation listet mit dem Parameter `--max-records` bis zu 20 zum Konto gehörende Sicherungen auf. Wenn der Parameter `--max-records` weggelassen wird, werden bis zu 50 Sicherungen aufgelistet.

```
aws elasticache describe-snapshots --max-records 20
```

Die folgende Operation listet mit dem Parameter `--cache-cluster-id` nur die zum Cluster `my-cluster` gehörenden Sicherungen auf.

```
aws elasticache describe-snapshots --cache-cluster-id my-cluster
```

Die folgende Operation listet mit dem Parameter `--snapshot-name` die Details der Sicherung `my-backup` auf.

```
aws elasticache describe-snapshots --snapshot-name my-backup
```

Weitere Informationen finden Sie unter [describe-snapshots](#) in der Befehlsreferenz.AWS CLI

Kopieren eines Backups

Sie können eine Kopie jedes beliebigen Backups erstellen. Dabei spielt es keine Rolle, ob es automatisch oder manuell erstellt wurde. Sie können Ihr Backup auch exportieren, sodass Sie von außen darauf zugreifen können ElastiCache. Eine Anleitung zum Exportieren Ihrer Sicherung finden Sie unter [Exportieren einer Sicherung](#).

Das folgende Verfahren zeigt, wie Sie ein Backup kopieren.

Kopieren von Backups (Konsole)

So kopieren Sie ein Backup (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste Ihrer Sicherungen anzuzeigen, wählen Sie im linken Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Sicherungen das Kontrollkästchen links neben dem Namen der Sicherung, die Sie kopieren möchten.
4. Wählen Sie Aktionen und dann Kopieren aus.
5. Geben Sie in das Feld New backup name einen Namen für die neue Sicherung ein.
6. Wählen Sie die Option Kopieren aus.

Kopieren eines Serverless-Backups (AWS CLI)

Verwenden Sie die Operation `copy-serverless-cache-snapshot`, um ein Backup eines Serverless-Caches zu kopieren.

Parameters

- `--source-serverless-cache-snapshot-name` Name der zu kopierenden Sicherung.
- `--target-serverless-cache-snapshot-name` Name der Sicherungskopie.

Das folgende Beispiel erstellt eine Kopie einer automatischen Sicherung.

Für Linux, macOS oder Unix:

```
aws elasticache copy-serverless-cache-snapshot \
```

```
--source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 \  
--target-serverless-cache-snapshot-name my-backup-copy
```

Für Windows:

```
aws elasticache copy-serverless-cache-snapshot ^  
--source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 ^  
--target-serverless-cache-snapshot-name my-backup-copy
```

Weitere Informationen finden Sie unter [copy-serverless-cache-snapshot](#) im AWS CLI.

Ein knotenbasiertes Cluster-Backup wird kopiert ()AWS CLI

Verwenden Sie den Vorgang, um eine Sicherung eines knotenbasierten Clusters zu kopieren. copy-snapshot

Parameters

- `--source-snapshot-name` Name der zu kopierenden Sicherung.
- `--target-snapshot-name` Name der Sicherungskopie.
- `--target-bucket` Reserviert zum Exportieren einer Sicherung. Verwenden Sie den Parameter nicht, um eine Kopie einer Sicherung anzufertigen. Weitere Informationen finden Sie unter [Exportieren einer Sicherung](#).

Das folgende Beispiel erstellt eine Kopie einer automatischen Sicherung.

Für Linux, macOS oder Unix:

```
aws elasticache copy-snapshot \  
--source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 \  
--target-snapshot-name amzn-s3-demo-bucket
```

Für Windows:

```
aws elasticache copy-snapshot ^  
--source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 ^  
--target-snapshot-name amzn-s3-demo-bucket
```

Weitere Informationen finden Sie unter [copy-snapshot](#) im AWS CLI.

Exportieren einer Sicherung

Amazon ElastiCache unterstützt den Export Ihres OSS-Backups ElastiCache für Redis in einen Amazon Simple Storage Service (Amazon S3) -Bucket, sodass Sie von außen ElastiCache darauf zugreifen können. Sie können ein Backup mithilfe der ElastiCache Konsole AWS CLI, der oder der ElastiCache API exportieren.

Das Exportieren eines Backups kann hilfreich sein, wenn Sie einen Cluster in einer anderen AWS Region starten müssen. Sie können Ihre Daten in eine AWS Region exportieren, die RDB-Datei in die neue AWS Region kopieren und dann diese RDB-Datei verwenden, um den neuen Cache zu speichern, anstatt darauf zu warten, dass der neue Cluster durch Use gefüllt wird. Informationen zum Erstellen eines neuen Clusters durch Seeding finden Sie unter [Tutorial: Seeding eines neuen knotenbasierten Clusters mit einem extern erstellten Backup](#). Ein weiterer Grund, warum Sie die Daten Ihres Caches exportieren möchten, ist die Verwendung der RDB-Datei für die Offline-Verarbeitung.

Important

- Das ElastiCache Backup und der Amazon S3 S3-Bucket, in den Sie es kopieren möchten, müssen sich in derselben AWS Region befinden.

Obwohl Sicherungen, die in einen Amazon-S3-Bucket kopiert werden, verschlüsselt sind, empfehlen wir dringend, dass Sie anderen Personen keinen Zugriff auf den Amazon-S3-Bucket mit Ihren gespeicherten Sicherungen gewähren.

- Das Exportieren eines Backups nach Amazon S3 wird für Cluster, die Daten-Tiering verwenden, nicht unterstützt. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).
- Das Exportieren eines Backups ist verfügbar für: knotenbasierte Valkey-Cluster, knotenbasierte Redis-OSS-Cluster und serverlose Valkey-, Memcached- und Redis OSS-Caches. Das Exportieren eines Backups ist für knotenbasierte Memcached-Cluster nicht verfügbar.

Bevor Sie ein Backup in einen Amazon S3 S3-Bucket exportieren können, müssen Sie über einen Amazon S3 S3-Bucket in derselben AWS Region wie das Backup verfügen. Gewähren ElastiCache Sie Zugriff auf den Bucket. Die ersten beiden Schritte zeigen, wie Sie dabei vorgehen.

Erstellen Sie einen Amazon S3 S3-Bucket

Die folgenden Schritte verwenden die Amazon S3 S3-Konsole, um einen Amazon S3 S3-Bucket zu erstellen, in den Sie Ihr ElastiCache Backup exportieren und speichern.

So erstellen Sie einen Amazon-S3-Bucket

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie Create Bucket (Bucket erstellen) aus.
3. Verfahren Sie unter Create a Bucket – Select a Bucket Name and Region wie folgt:

- a. Geben Sie für Bucket-Name einen Namen für Ihren Amazon-S3-Bucket ein.

Der Name Ihres Amazon-S3-Buckets muss DNS-konform sein. Andernfalls ElastiCache kann nicht auf Ihre Backup-Datei zugreifen. Die Regeln für die DNS-Konformität lauten:

- Bucket-Namen müssen mindestens 3 und dürfen höchstens 63 Zeichen umfassen.
 - Die Namen müssen eine Folge aus einer oder mehreren Beschriftungen darstellen, die durch einen Punkt (.) getrennt sind, wobei jede Beschriftung:
 - mit einem Kleinbuchstaben oder einer Zahl beginnen.
 - mit einem Kleinbuchstaben oder einer Zahl beginnen.
 - Enthält nur Kleinbuchstaben, Zahlen und Bindestriche.
 - Er darf nicht als IP-Adresse (z. B. 192.0.2.0) formatiert sein.
- b. Wählen Sie aus der Regionsliste eine AWS Region für Ihren Amazon S3 S3-Bucket aus. Diese AWS Region muss dieselbe AWS Region sein wie das ElastiCache Backup, das Sie exportieren möchten.
 - c. Wählen Sie Erstellen aus.

Weitere Informationen zum Erstellen eines Amazon-S3-Buckets finden Sie unter [Erstellen von Buckets](#) im Handbuch für Amazon Simple Storage Service.

Gewähren Sie ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket

ElastiCache Um einen Snapshot in einen Amazon S3 S3-Bucket kopieren zu können, müssen Sie Ihre IAM-Bucket-Richtlinie aktualisieren, um ElastiCache Zugriff auf den Bucket zu gewähren.

 Warning

Obwohl Sicherungen, die in einen Amazon-S3-Bucket kopiert werden, verschlüsselt sind, kann jede Person mit Zugriff auf Ihren Amazon-S3-Bucket auf Ihre Daten zugreifen. Daher wird sehr dazu geraten, IAM-Richtlinien einzurichten, um unbefugten Zugriff auf diesen Amazon-S3-Bucket zu verhindern. Weitere Informationen finden Sie unter [Verwalten des Zugriffs](#) im Benutzerhandbuch zu Amazon S3.

Führen Sie die folgenden Schritte aus, um die korrekten Berechtigungen für einen Amazon S3 Bucket zu erstellen.

Um ElastiCache Zugriff auf einen S3-Bucket zu gewähren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Amazon-S3-Buckets aus, in den Sie die Sicherung kopieren möchten. Dies sollte der in [Erstellen Sie einen Amazon S3 S3-Bucket](#) erstellte S3-Bucket sein.
3. Wählen Sie die Registerkarte Permissions (Berechtigungen) und unter Permissions (Berechtigungen) Access control list (ACL) (Zugriffssteuerungsliste (ACL)) und dann Edit (Bearbeiten) aus.
4. Fügen Sie die Fügen Sie die Kanonische Empfänger-ID
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353 mit den folgenden Optionen hinzu:
 - Objekte – Auflisten oder Schreiben
 - Bucket ACL: Lesen oder Schreiben

 Note

- Für die GovCloud PDT-Region lautet die Canonical ID.
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
- Für die GovCloud OSU-Region lautet die kanonische ID.
c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c

5. Wählen Sie Speichern.

Exportieren Sie ein Backup ElastiCache

Jetzt haben Sie Ihren S3-Bucket erstellt und ElastiCache Zugriffsberechtigungen erteilt. Als Nächstes können Sie die ElastiCache Konsole, die AWS CLI oder die ElastiCache API verwenden, um Ihren Snapshot dorthin zu exportieren.

Nachfolgend finden Sie ein Beispiel dafür, wie die aktualisierte Richtlinie aussehen könnte.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399484",
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::hkg-elasticache-backup",
        "arn:aws:s3:::hkg-elasticache-backup/*"
      ]
    }
  ]
}
```

Für Opt-in-Regionen finden Sie im Folgenden ein Beispiel dafür, wie die aktualisierte IAM-Richtlinie für den S3-Bucket aussehen könnte. (In diesem Beispiel wird die Region Asien-Pazifik (Hongkong) verwendet.)

JSON

```
{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399483",
```

```
    "Effect": "Allow",
    "Action": "s3:*",
    "Resource": [
        "arn:aws:s3:::hkg-elasticache-backup",
        "arn:aws:s3:::hkg-elasticache-backup/*"
    ]
},
{
    "Sid": "Stmt15399484",
    "Effect": "Allow",
    "Action": "s3:*",
    "Resource": [
        "arn:aws:s3:::hkg-elasticache-backup",
        "arn:aws:s3:::hkg-elasticache-backup/*"
    ]
}
]
```

Ein ElastiCache Backup exportieren (Konsole)

Die folgenden Schritte verwenden die ElastiCache Konsole, um ein Backup in einen Amazon S3 S3-Bucket zu exportieren, sodass Sie von außerhalb darauf zugreifen können ElastiCache. Der Amazon S3 S3-Bucket muss sich in derselben AWS Region wie das ElastiCache Backup befinden.

Um ein ElastiCache Backup in einen Amazon S3 S3-Bucket zu exportieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste Ihrer Sicherungen anzuzeigen, wählen Sie im linken Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Sicherungen das Kontrollkästchen links neben dem Namen der Sicherung, die Sie exportieren möchten.
4. Wählen Sie die Option Kopieren aus.
5. Gehen Sie im Dialogfeld Create a Copy of the Backup (Eine Kopie der Sicherung erstellen?) wie folgt vor:
 - a. Geben Sie in das Feld New backup name einen Namen für die neue Sicherung ein.

Der Name muss zwischen 1 und 1.000 Zeichen lang sein und UTF-8-codierbar sein.

ElastiCache fügt eine Instanz-ID und `.rdb` zu dem Wert hinzu, den Sie hier eingeben. Wenn Sie z. B. `my-exported-backup` eingeben, erstellt ElastiCache `my-exported-backup-0001.rdb`.

- b. Wählen Sie aus der Liste S3-Zielspeicherort den Namen des Amazon-S3-Buckets aus, in den Sie Ihre Sicherung kopieren möchten (der von Ihnen unter [Erstellen Sie einen Amazon S3 S3-Bucket](#) erstellte Bucket).

Der Ziel-S3-Standort muss ein Amazon S3 S3-Bucket in der AWS Region des Backups mit den folgenden Berechtigungen sein, damit der Exportvorgang erfolgreich ist.

- Objektzugriff – Lesen und Schreiben.
- Berechtigungszugriff – Lesen.

Weitere Informationen finden Sie unter [Gewähren Sie ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket](#).

- c. Wählen Sie die Option Kopieren aus.

Note

Wenn Ihr S3-Bucket nicht über die erforderlichen Berechtigungen verfügt, ElastiCache um ein Backup dorthin zu exportieren, erhalten Sie eine der folgenden Fehlermeldungen. Kehren Sie zu [Gewähren Sie ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket](#) zurück, um die angegebenen Berechtigungen hinzuzufügen, und exportieren Sie Ihre Sicherung erneut.

- ElastiCache hat keine LESE-Rechte %s für den S3-Bucket erhalten.

Lösung: Fügen Sie Leseberechtigungen für den Bucket hinzu.

- ElastiCache hat keine SCHREIBBERECHTIGUNG %s für den S3-Bucket erhalten.

Lösung: Fügen Sie Schreibberechtigungen für den Bucket hinzu.

- ElastiCache hat keine READ_ACP-Berechtigungen %s für den S3-Bucket erhalten.

Lösung: Fügen Sie Read-Zugriff für Berechtigungen für den Bucket hinzu.

Wenn Sie Ihr Backup in eine andere AWS Region kopieren möchten, verwenden Sie Amazon S3, um es zu kopieren. Weitere Informationen finden Sie unter [Kopieren von Objekten](#) im Entwicklerhandbuch zu Amazon Simple Storage Service.

Exportieren eines ElastiCache serverlosen Backups (AWS CLI)

Exportieren des Backups eines Serverless-Caches

Exportieren Sie die Sicherung mit der CLI-Operation `export-serverless-cache-snapshot` mit den folgenden Parametern zu einem Amazon-S3-Bucket:

Parameters

- `--serverless-cache-snapshot-name` Name der zu kopierenden Sicherung.
- `--s3-bucket-name` – Name des Amazon-S3-Buckets, zu dem die Sicherung exportiert werden soll. Im angegebenen Bucket wird eine Kopie der Sicherung erstellt.

Damit der Exportvorgang erfolgreich ist, `--s3-bucket-name` muss es sich um einen Amazon S3 S3-Bucket in der AWS Region des Backups mit den folgenden Berechtigungen handeln.

- Objektzugriff – Lesen und Schreiben.
- Berechtigungszugriff – Lesen.

Die folgende Operation kopiert eine Sicherung zu `my-s3-bucket`.

Für Linux, macOS oder Unix:

```
aws elasticache export-serverless-cache-snapshot \
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 \
  --s3-bucket-name my-s3-bucket
```

Für Windows:

```
aws elasticache export-serverless-cache-snapshot ^
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 ^
  --s3-bucket-name my-s3-bucket
```

Exportieren eines knotenbasierten ElastiCache Cluster-Backups (AWS CLI)

Exportieren eines Backups eines knotenbasierten Clusters

Exportieren Sie die Sicherung mit der CLI-Operation `copy-snapshot` mit den folgenden Parametern zu einem Amazon-S3-Bucket:

Parameters

- `--source-snapshot-name` Name der zu kopierenden Sicherung.
- `--target-snapshot-name` Name der Sicherungskopie.

Der Name muss zwischen 1 und 1.000 Zeichen lang sein und UTF-8-codierbar sein.

ElastiCache fügt dem hier eingegebenen Wert eine Instanz-ID und `.rdb` hinzu. Wenn Sie z. B. `my-exported-backup` eingeben, erstellt ElastiCache `my-exported-backup-0001.rdb`.

- `--target-bucket` – Name des Amazon-S3-Buckets, zu dem die Sicherung exportiert werden soll. Im angegebenen Bucket wird eine Kopie der Sicherung erstellt.

Damit der Exportvorgang erfolgreich ist, `--target-bucket` muss es sich um einen Amazon S3 S3-Bucket in der AWS Region des Backups mit den folgenden Berechtigungen handeln.

- Objektzugriff – Lesen und Schreiben.
- Berechtigungszugriff – Lesen.

Weitere Informationen finden Sie unter [Gewähren Sie ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket](#).

Die folgende Operation kopiert eine Sicherung zu `my-s3-bucket`.

Für Linux, macOS oder Unix:

```
aws elasticache copy-snapshot \  
  --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 \  
  --target-snapshot-name my-exported-backup \  
  --target-bucket my-s3-bucket
```

Für Windows:

```
aws elasticache copy-snapshot ^  
  --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 ^  
  --target-snapshot-name my-exported-backup ^  
  --target-bucket my-s3-bucket
```

Wiederherstellen aus einem Backup in einen neuen Cache

Sie können ein vorhandenes Backup von Valkey in einem neuen Valkey-Cache oder einem knotenbasierten Cluster wiederherstellen und ein vorhandenes Redis OSS-Backup in einem neuen Redis OSS-Cache oder einem knotenbasierten Cluster wiederherstellen. Sie können auch ein vorhandenes Memcached-Cache-Backup in einem neuen serverlosen Memcached-Cache wiederherstellen.

Wiederherstellen eines Backups in einem Serverless-Cache (Konsole)

Note

ElastiCache Serverless unterstützt RDB-Dateien, die mit Valkey 7.2 und höher kompatibel sind, sowie Redis OSS-Versionen zwischen 5.0 und der neuesten verfügbaren Version.

So stellen Sie ein Backup in einem Serverless-Cache wieder her (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Backups das Kontrollkästchen links neben dem Namen des Backups, das Sie wiederherstellen möchten.
4. Wählen Sie Aktionen und anschließend Wiederherstellen aus.
5. Geben Sie einen Namen und eine optionale Beschreibung für den neuen Serverless-Cache ein.
6. Klicken Sie auf Erstellen, um Ihren neuen Cache zu erstellen und Daten aus Ihrem Backup zu importieren.

Wiederherstellung eines Backups in einem knotenbasierten Cluster (Konsole)

So stellen Sie ein Backup auf einem knotenbasierten Cluster (Konsole) wieder her

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Sicherungen das Kontrollkästchen links neben dem Namen der Sicherung, aus der Sie wiederherstellen möchten.

4. Wählen Sie Aktionen und anschließend Wiederherstellen aus.
5. Wählen Sie Node-based Cache und passen Sie die Cluster-Einstellungen wie Knotentyp, Größe, Anzahl der Shards, Replikate, AZ-Platzierung und Sicherheitseinstellungen an.
6. Wählen Sie Erstellen, um Ihren neuen knotenbasierten Cluster zu erstellen und Daten aus Ihrem Backup zu importieren.

Wiederherstellen eines Backups in einem Serverless-Cache (AWS CLI)

 Note

ElastiCache Serverless unterstützt RDB-Dateien, die mit Valkey 7.2 und höher kompatibel sind, sowie Redis OSS-Versionen zwischen 5.0 und der neuesten verfügbaren Version.

So stellen Sie ein Backup in einem neuen Serverless-Cache wieder her (AWS CLI)

Im folgenden AWS CLI Beispiel wird ein neuer Cache erstellt, der Daten aus einer Sicherung verwendet `create-serverless-cache` und aus dieser importiert.

Für Linux, macOS oder Unix:

```
aws elasticache create-serverless-cache \  
  
    --serverless-cache-name CacheName \  
    --engine redis  
    --snapshot-arns-to-restore Snapshot-ARN
```

Für Windows:

```
aws elasticache create-serverless-cache ^  
  
    --serverless-cache-name CacheName ^  
    --engine redis ^  
    --snapshot-arns-to-restore Snapshot-ARN
```

Löschen einer Sicherung

Eine automatische Sicherung wird automatisch gelöscht, wenn ihre Aufbewahrungsfrist abläuft. Wenn Sie einen Cluster löschen, werden alle seine automatischen Sicherungen ebenfalls gelöscht. Wenn

Sie eine Replikationsgruppe löschen, werden alle automatischen Sicherungen von den Clustern in der betreffenden Gruppe ebenfalls gelöscht.

ElastiCache bietet einen API-Löschvorgang, mit dem Sie ein Backup jederzeit löschen können, unabhängig davon, ob das Backup automatisch oder manuell erstellt wurde. Da für manuelle Sicherungen keine Aufbewahrungsfrist gilt, können sie nur durch manuelles Löschen entfernt werden.

Sie können ein Backup mithilfe der ElastiCache Konsole AWS CLI, der oder der ElastiCache API löschen.

Löschen einer Sicherung (Konsole)

Das folgende Verfahren löscht ein Backup mithilfe der ElastiCache Konsole.

So löschen Sie ein Backup

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Sicherungen aus.

Der Bildschirm „Backups“ wird mit einer Liste Ihrer Sicherungen angezeigt.

3. Aktivieren Sie das Kontrollkästchen links neben dem Namen der Sicherung, die Sie löschen möchten.
4. Wählen Sie Löschen aus.
5. Wenn Sie diese Sicherung löschen möchten, wählen Sie auf dem Bestätigungsbildschirm Delete Backup die Option Delete aus. Der Status wird in deleting geändert.

Löschen eines Serverless-Backups (AWS CLI)

Verwenden Sie den AWS CLI Vorgang delete-snapshot mit dem folgenden Parameter, um ein serverloses Backup zu löschen.

- `--serverless-cache-snapshot-name` Name der zu löschenden Sicherung.

Der folgende Code löscht die Sicherung myBackup.

```
aws elasticache delete-serverless-cache-snapshot --serverless-cache-snapshot-name myBackup
```

Weitere Informationen finden Sie unter [delete-serverless-cache-snapshot](#) in der Referenz zum AWS CLI-Befehl.

Löschen einer knotenbasierten Clustersicherung ()AWS CLI

Verwenden Sie den AWS CLI Vorgang delete-snapshot mit dem folgenden Parameter, um eine knotenbasierte Clustersicherung zu löschen.

- `--snapshot-name` Name der zu löschenden Sicherung.

Der folgende Code löscht die Sicherung myBackup.

```
aws elasticache delete-snapshot --snapshot-name myBackup
```

Weitere Informationen finden Sie unter [delete-snapshot](#) in der AWS CLI-Befehlsreferenz.

Markieren von Sicherungen

Sie können -Tresoren Metadaten in Form von Tags zuweisen. Mit Tags (Markierungen) können Sie Ihre -Ressourcen auf unterschiedliche Weise kategorisieren (z. B. nach Zweck, Eigentümer oder Umgebung). Dies ist nützlich, wenn Sie viele Ressourcen desselben Typs haben — In diesem Fall können Sie schnell bestimmte Ressourcen basierend auf den zugewiesenen Tags (Markierungen) bestimmen. Weitere Informationen finden Sie unter [Ressourcen, die markiert werden können](#).

Mithilfe von Tags zur Kostenzuweisung können Sie Ihre Kosten für mehrere AWS Dienste verfolgen, indem Sie Ihre Ausgaben auf Rechnungen nach Tagwerten gruppieren. Weitere Informationen zu Kostenzuordnungs-Tags finden Sie unter [Verwenden von Kostenzuordnungs-Tags](#).

Mithilfe der ElastiCache Konsole AWS CLI, der oder der ElastiCache API können Sie Kostenzuordnungs-Tags zu Ihren Backups hinzufügen, auflisten, ändern, entfernen oder kopieren. Weitere Informationen finden Sie unter [Überwachung von Kosten mit Kostenzuordnungs-Tags](#).

Tutorial: Seeding eines neuen knotenbasierten Clusters mit einem extern erstellten Backup

Wenn Sie einen neuen knotenbasierten Valkey- oder Redis OSS-Cluster erstellen, können Sie ihn mit Daten aus einer Valkey- oder Redis OSS .rdb-Backup-Datei versorgen. Das Seeding des Clusters ist nützlich, wenn Sie derzeit eine Valkey- oder Redis OSS-Instanz außerhalb von verwalten und Ihren neuen knotenbasierten Cluster ElastiCache für Redis OSS mit Ihren vorhandenen Valkey ElastiCache - oder Redis OSS-Daten auffüllen möchten.

Informationen zum Seeding eines neuen knotenbasierten Valkey- oder Redis OSS-Clusters aus einem Valkey- oder Redis OSS-Backup, das in Amazon erstellt wurde, finden Sie unter [ElastiCache Wiederherstellen aus einem Backup in einen neuen Cache](#)

Wenn Sie eine Valkey- oder Redis OSS .rdb-Datei zum Seeding eines neuen knotenbasierten Clusters verwenden, können Sie Folgendes tun:

- Führen Sie ein Upgrade von einem nicht partitionierten Cluster auf einen knotenbasierten Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) durch, auf dem Redis OSS Version 3.2.4 ausgeführt wird.
- Geben Sie eine Reihe von Shards (in der API und CLI als Knotengruppen bezeichnet) im neuen knotenbasierten Cluster an. Diese Anzahl kann sich von der Anzahl der Shards im knotenbasierten Cluster unterscheiden, der zur Erstellung der Backup-Datei verwendet wurde.
- Geben Sie einen anderen Knotentyp für den neuen knotenbasierten Cluster an, der größer oder kleiner ist als der Knotentyp, der in dem Cluster verwendet wurde, der das Backup erstellt hat. Wenn Sie auf einen kleineren Knotentyp skalieren, stellen Sie sicher, dass der neue Knotentyp über ausreichend Speicher für Ihre Daten und den OSS-Overhead von Valkey oder Redis verfügt. Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#).
- Verteilen Sie Ihre Schlüssel in den Steckplätzen des neuen Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) anders als in dem Cluster, der zur Erstellung der Backup-Datei verwendet wurde.

 Note

Sie können einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) nicht anhand einer .rdb-Datei starten, die aus einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) erstellt wurde.

 Important

- Sie müssen sicherstellen, dass Ihre Valkey- oder Redis OSS-Backup-Daten die Ressourcen des Knotens nicht überschreiten. Sie können beispielsweise keine .rdb-Datei mit 5 GB Valkey- oder Redis-OSS-Daten auf einen cache.m3.medium-Knoten hochladen, der über 2,9 GB Arbeitsspeicher verfügt.

Wenn die Sicherung zu groß ist, lautet der Status des resultierenden Clusters `restore-failed`. In diesem Fall müssen Sie den Cluster löschen und von Neuem beginnen.

Eine vollständige Liste der Knotentypen und Spezifikationen finden Sie unter [Redis OSS-Knotentyp-spezifische Parameter](#) und zu [ElastiCache Amazon-Produktfunktionen und -details](#).

- Sie können eine Valkey- oder Redis OSS .rdb-Datei nur mit der serverseitigen Amazon S3 S3-Verschlüsselung (SSE-S3) verschlüsseln. Weitere Informationen finden Sie unter [Schützen von Daten mithilfe serverseitiger Verschlüsselung](#).

Im Folgenden finden Sie Themen, die Sie durch die Migration Ihres Clusters von außerhalb ElastiCache für Valkey oder Redis OSS zu für Redis OSS führen. ElastiCache

Migration zu für Redis OSS ElastiCache

- [Schritt 1: Erstellen Sie ein Valkey- oder Redis OSS-Backup](#)
- [Schritt 2: Erstellen eines Amazon-S3-Buckets und -Ordners](#)
- [Schritt 3: Hochladen Ihrer Sicherung auf Amazon S3](#)
- [Schritt 4: Gewähren ElastiCache Sie Lesezugriff auf die .rdb-Datei](#)

Migration von externen Diensten zu ElastiCache für Redis OSS.

- [Schritt 1: Erstellen Sie ein Valkey- oder Redis OSS-Backup](#)

- [Schritt 2: Erstellen eines Amazon-S3-Buckets und -Ordners](#)
- [Schritt 3: Hochladen Ihrer Sicherung auf Amazon S3](#)
- [Schritt 4: Gewähren ElastiCache Sie Lesezugriff auf die .rdb-Datei](#)

Schritt 1: Erstellen Sie ein Valkey- oder Redis OSS-Backup

Um das Valkey- oder Redis OSS-Backup zu erstellen, um Ihre ElastiCache Redis OSS-Instanz zu starten

1. Connect zu Ihrer vorhandenen Valkey- oder Redis OSS-Instanz her.
2. Führen Sie einen der beiden BGSAVE SAVE Operationen aus, um ein Backup zu erstellen. Achten Sie auf den Speicherort der .rdb-Datei.

BGSAVE ist asynchron und blockiert während der Verarbeitung keine anderen Clients. Weitere Informationen finden Sie unter [BGSAVE](#) auf der Valkey-Website.

SAVE ist synchron und blockiert andere Vorgänge, bis sie Verarbeitung abgeschlossen ist. Weitere Informationen finden Sie unter [SAVE](#) auf der Valkey-Website.

Weitere Informationen zum Erstellen eines Backups finden Sie unter [Persistence](#) auf der Valkey-Website.

Schritt 2: Erstellen eines Amazon-S3-Buckets und -Ordners

Wenn Sie die Sicherungsdatei erstellt haben, müssen Sie sie zu einem Ordner innerhalb eines Amazon-S3-Buckets hochladen. Hierzu müssen bereits ein Amazon-S3-Bucket und ein Ordner innerhalb dieses Buckets vorhanden sein. Wenn Sie bereits einen Amazon-S3-Bucket und Ordner mit den entsprechenden Berechtigungen besitzen, können Sie mit [Schritt 3: Hochladen Ihrer Sicherung auf Amazon S3](#) fortfahren.

So erstellen Sie einen Amazon-S3-Bucket

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Führen Sie die Anweisungen zum Erstellen eines Amazon S3 Buckets unter [Erstellen eines Buckets](#) im Benutzerhandbuch zu Amazon Simple Storage Service aus.

Der Name Ihres Amazon-S3-Buckets muss DNS-konform sein. Andernfalls ElastiCache kann nicht auf Ihre Backup-Datei zugreifen. Die Regeln für die DNS-Konformität lauten:

- Bucket-Namen müssen mindestens 3 und dürfen höchstens 63 Zeichen umfassen.
- Die Namen müssen eine Folge aus einer oder mehreren Beschriftungen darstellen, die durch einen Punkt (.) getrennt sind, wobei jede Beschriftung:
 - mit einem Kleinbuchstaben oder einer Zahl beginnen.
 - mit einem Kleinbuchstaben oder einer Zahl beginnen.
 - Enthält nur Kleinbuchstaben, Zahlen und Bindestriche.
- Er darf nicht als IP-Adresse (z. B. 192.0.2.0) formatiert sein.

Sie müssen Ihren Amazon S3 S3-Bucket in derselben AWS Region wie Ihr neuer OSS-Cluster ElastiCache für Redis erstellen. Dieser Ansatz stellt sicher, dass beim ElastiCache Lesen Ihrer RDB-Datei aus Amazon S3 die höchste Datenübertragungsgeschwindigkeit erreicht wird.

 Note

Um Ihre Daten so sicher wie möglich zu halten, sollten Sie die Berechtigungen für Ihr Amazon-S3-Bucket so restriktiv wie möglich gestalten. Gleichzeitig müssen die Berechtigungen weiterhin zulassen, dass der Bucket und sein Inhalt für das Seeding Ihres neuen Valkey- oder Redis-OSS-Clusters verwendet werden können.

So fügen Sie einem Amazon S3 Bucket einen Ordner hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, in den die .rdb-Datei hochgeladen werden soll.
3. Wählen Sie Create folder.
4. Geben Sie einen Namen für den neuen Ordner ein.
5. Wählen Sie Speichern.

Notieren Sie sich sowohl den Bucketnamen als auch den Ordernamen.

Schritt 3: Hochladen Ihrer Sicherung auf Amazon S3

Laden Sie jetzt die .rdb-Datei hoch, die Sie in [Schritt 1: Erstellen Sie ein Valkey- oder Redis OSS-Backup](#) erstellt haben. Sie laden sie in den Amazon-S3-Bucket und -Ordner hoch, die sie in [Schritt 2: Erstellen eines Amazon-S3-Buckets und -Ordners](#) erstellt haben. Weitere Informationen zu dieser Aufgabe finden Sie unter [Objekte in einen Bucket einfügen](#). Wählen Sie zwischen den Schritten 2 und 3 den Namen des von Ihnen erstellten Ordners aus.

So laden Sie Ihre .rdb-Datei in einen Amazon-S3-Ordner hoch

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Amazon-S3-Bucket aus, den Sie in Schritt 2 erstellt haben.
3. Wählen Sie den Namen des Ordners aus, den Sie in Schritt 2 erstellt haben.
4. Klicken Sie auf Upload.
5. Klicken Sie auf Add files.
6. Navigieren Sie zu der Datei oder den Dateien, die Sie hochladen möchten, und wählen Sie dann die Datei oder die Dateien aus. Halten Sie zur Auswahl mehrerer Dateien die Strg-Taste während der Auswahl der Dateinamen gedrückt.
7. Klicken Sie auf Open.
8. Bestätigen Sie, dass die korrekte Datei bzw. die korrekten Dateien im Dialogfeld Upload aufgelistet werden. Wählen Sie dann Upload.

Notieren Sie den Pfad zu Ihrer .rdb-Datei. Wenn der Bucket-Name z. B. myBucket und der Pfad myFolder/redis.rdb lautet, geben Sie myBucket/myFolder/redis.rdb ein. Sie benötigen diesen Pfad zum Starten des neuen Clusters mit den Daten in dieser Sicherung.

Weitere Informationen finden Sie unter [Bucket-Einschränkungen und -Limits](#) im Entwicklerhandbuch zu Amazon Simple Storage Service.

Schritt 4: Gewähren ElastiCache Sie Lesezugriff auf die .rdb-Datei

Gewähren ElastiCache Sie jetzt Lesezugriff auf Ihre .rdb-Backup-Datei. Sie gewähren ElastiCache Zugriff auf Ihre Backup-Datei auf unterschiedliche Weise, je nachdem, ob sich Ihr Bucket in einer AWS Standardregion oder einer AWS Opt-in-Region befindet.

AWS Regionen, die vor dem 20. März 2019 eingeführt wurden, sind standardmäßig aktiviert. Sie können sofort mit der Arbeit in diesen AWS Regionen beginnen. Regionen, die nach dem 20. März 2019 eingeführt werden, wie Asien-Pazifik (Hongkong) und Naher Osten (Bahrain), sind standardmäßig deaktiviert. Sie müssen diese Regionen aktivieren oder sich dafür anmelden, bevor Sie sie verwenden können, wie unter [Verwalten von AWS-Regionen](#) im Allgemeine AWS-Referenz beschrieben.

Wählen Sie Ihren Ansatz je nach AWS Region:

- Verwenden Sie für eine Standardregion das Verfahren unter [Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei in einer Standardregion](#).
- Verwenden Sie für eine Opt-In-Region das Verfahren unter [Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei in einer Opt-in-Region](#).

Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei in einer Standardregion

AWS Regionen, die vor dem 20. März 2019 eingeführt wurden, sind standardmäßig aktiviert. Sie können sofort mit der Arbeit in diesen AWS Regionen beginnen. Regionen, die nach dem 20. März 2019 eingeführt werden, wie Asien-Pazifik (Hongkong) und Naher Osten (Bahrain), sind standardmäßig deaktiviert. Sie müssen diese Regionen aktivieren oder sich dafür anmelden, bevor Sie sie verwenden können, wie unter [Verwalten von AWS-Regionen](#) im Allgemeine AWS-Referenz beschrieben.

Um ElastiCache Lesezugriff auf die Backup-Datei in einer AWS Region zu gewähren, die standardmäßig aktiviert ist

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des S3-Buckets aus, in dem sich Ihre .rdb-Datei befindet.
3. Wählen Sie den Namen des Ordners aus, in dem sich Ihre .rdb-Datei befindet.
4. Wählen Sie den Namen Ihrer .rdb-Sicherungsdatei aus. Der Name der ausgewählten Datei erscheint oberhalb der Registerkarten oben auf der Seite.
5. Wählen Sie Berechtigungen.
6. Wenn aws-scs-s3-readonly oder einer der kanonischen Benutzer IDs in der folgenden Liste nicht als Benutzer aufgeführt ist, gehen Sie wie folgt vor:

- a. Wählen Sie unter Zugriff für andere AWS Konten die Option Zuschussempfänger hinzufügen aus.
- b. Fügen Sie in dem Feld die kanonische ID der AWS Region wie folgt hinzu:

- AWS GovCloud Region (USA West):

```
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
```

 Important

Das Backup muss sich in einem S3-Bucket befinden, damit Sie es auf einen Valkey- oder Redis-OSS-Cluster herunterladen können. AWS GovCloud (US) AWS GovCloud (US)

- AWS Standardmäßig aktivierte Regionen:

```
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
```

- c. Richten Sie die Berechtigungen für den Bucket durch Auswahl von Yes (Ja) für die folgenden Optionen ein:
 - List/write object (Objekt auflisten/schreiben)
 - Read/write object ACL permissions (Lese-/Schreibberechtigungen für Objekt-ACL)
- d. Wählen Sie Speichern.

7. Wählen Sie Overview (Übersicht) und dann Download (Herunterladen).

Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei in einer Opt-in-Region

AWS Regionen, die vor dem 20. März 2019 eingeführt wurden, sind standardmäßig aktiviert. Sie können sofort mit der Arbeit in diesen AWS Regionen beginnen. Regionen, die nach dem 20. März 2019 eingeführt werden, wie Asien-Pazifik (Hongkong) und Naher Osten (Bahrain), sind standardmäßig deaktiviert. Sie müssen diese Regionen aktivieren oder sich dafür anmelden, bevor Sie sie verwenden können, wie unter [Verwalten von AWS-Regionen](#) im Allgemeine AWS-Referenz beschrieben.

Gewähren Sie jetzt ElastiCache Lesezugriff auf Ihre .rdb-Backup-Datei.

Um ElastiCache Lesezugriff auf die Sicherungsdatei zu gewähren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des S3-Buckets aus, in dem sich Ihre .rdb-Datei befindet.
3. Wählen Sie den Namen des Ordners aus, in dem sich Ihre .rdb-Datei befindet.
4. Wählen Sie den Namen Ihrer .rdb-Sicherungsdatei aus. Der Name der ausgewählten Datei erscheint oberhalb der Registerkarten oben auf der Seite.
5. Wählen Sie die Registerkarte Berechtigungen.
6. Wählen Sie unter Permissions (Berechtigungen) Bucket policy (Bucket-Richtlinie) aus und klicken Sie dann auf Edit (Bearbeiten).
7. Aktualisieren Sie die Richtlinie, um die ElastiCache erforderlichen Berechtigungen zur Durchführung von Vorgängen zu gewähren:
 - Fügen Sie ["Service" : "*region-full-name*.elasticache-snapshot.amazonaws.com"] zu Principal hinzu.
 - Fügen Sie die folgenden Berechtigungen hinzu, die für das Exportieren eines Snapshots in den Amazon-S3-Bucket erforderlich sind:
 - "s3:GetObject"
 - "s3:ListBucket"
 - "s3:GetBucketAcl"

Nachfolgend finden Sie ein Beispiel dafür, wie die aktualisierte Richtlinie aussehen könnte.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399483",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
```

```

        "s3:GetBucketAcl"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/backup1.rdb",
        "arn:aws:s3:::amzn-s3-demo-bucket/backup2.rdb"
    ]
}

```

8. Wählen Sie **Änderungen speichern** aus.

Füllen Sie den ElastiCache Cluster mit den Daten der .rdb-Datei

Jetzt sind Sie bereit, einen ElastiCache Cluster zu erstellen und ihn mit den Daten aus der RDB-Datei zu versorgen. Folgen Sie zum Erstellen des Clusters den Anweisungen unter [Einen Cluster für Valkey oder Redis OSS erstellen](#) oder [Eine Valkey- oder Redis OSS-Replikationsgruppe von Grund auf neu erstellen](#). Achten Sie darauf, Valkey oder Redis OSS als Cluster-Engine zu wählen.

Die Methode, mit der Sie angeben, ElastiCache wo sich das Backup befindet, das Sie auf Amazon S3 hochgeladen haben, hängt von der Methode ab, mit der Sie den Cluster erstellt haben:

Füllen Sie den OSS-Cluster oder die Replikationsgruppe ElastiCache für Redis mit den Daten der .rdb-Datei

- Verwenden der Konsole ElastiCache

Wählen Sie bei der Auswahl der Cluster settings (Cluster-Einstellungen) die Option **Restore from backups** (Aus Backups wiederherstellen) als Methode zur Cluster-Erstellung aus, und wählen Sie dann im Abschnitt **Backup source** (Backup-Quelle) die Option **Other backups** (Andere Backups) als **Source** (Quelle) aus. Geben Sie im Feld **Seed-RDB-Datei-S3-Speicherort** den Amazon-S3-Pfad für die Datei(en) ein. Wenn Sie mehrere .rdb-Dateien besitzen, geben Sie den Pfad für jede Datei in eine durch Kommas getrennten Liste ein. Der Amazon-S3-Pfad sieht etwa aus wie *myBucket/myFolder/myBackupFilename.rdb*.

- Mit dem AWS CLI

Verwenden Sie bei Einsatz der Operation `create-cache-cluster` oder `create-replication-group` den Parameter `--snapshot-arns`, um einen vollqualifizierten ARN für jede .rdb-Datei anzugeben. Beispiel,

`arn:aws:s3:::myBucket/myFolder/myBackupFilename.rdb`. Der ARN muss auf die in Amazon S3 gespeicherten Sicherungsdateien aufgelöst werden.

- Mithilfe der ElastiCache API

Wenn Sie die `CreateReplicationGroup` ElastiCache API-Operation `CreateCacheCluster` oder verwenden, verwenden Sie den Parameter `SnapshotArns` um einen vollqualifizierten ARN für jede `.rdb`-Datei anzugeben. Beispiel, `arn:aws:s3:::myBucket/myFolder/myBackupFilename.rdb`. Der ARN muss auf die in Amazon S3 gespeicherten Sicherungsdateien aufgelöst werden.

Important

Beim Seeding eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) müssen Sie jede Knotengruppe (Shard) im neuen Cluster oder in der neuen Replikationsgruppe konfigurieren. Dazu verwenden Sie den Parameter `--node-group-configuration` (API: `NodeGroupConfiguration`). Weitere Informationen finden Sie hier:

- CLI: [create-replication-group](#) in der AWS CLI Referenz
- API: [CreateReplicationGroup](#) in der ElastiCache API-Referenz

Während der Erstellung Ihres Clusters werden die Daten in Ihrem Valkey- oder Redis OSS-Backup in den Cluster geschrieben. Sie können den Fortschritt überwachen, indem Sie sich die ElastiCache Ereignismeldungen ansehen. Rufen Sie dazu die ElastiCache Konsole auf und wählen Sie `Cache Events` aus. Sie können auch die AWS ElastiCache Befehlszeilenschnittstelle oder ElastiCache API verwenden, um Ereignismeldungen abzurufen. Weitere Informationen finden Sie unter [ElastiCache Ereignisse anzeigen](#).

Motorversionen und Aufrüstung in ElastiCache

In diesem Abschnitt werden die unterstützten Valkey-, Memcached- und Redis OSS-Engines sowie Informationen zum Upgrade behandelt. Beachten Sie, dass alle mit Redis OSS 7.2 verfügbaren Funktionen standardmäßig in Valkey 7.2 und höher verfügbar sind. Sie können auch ein Upgrade von einigen vorhandenen ElastiCache für Redis OSS-Engines auf eine Valkey-Engine durchführen.

Aufrüstung von Motorversionen, einschließlich motorübergreifender Upgrades

Valkey und Redis OSS

Mit Valkey und Redis OSS initiieren Sie Versionsupgrades für Ihren Cluster oder Ihre Replikationsgruppe, indem Sie ihn mithilfe der ElastiCache Konsole, der oder der AWS CLI ElastiCache API ändern und eine neuere Engine-Version angeben.

Sie können auch ein Cross-Upgrade von Redis OSS auf Valkey durchführen. Weitere Informationen zu Cross-Upgrades finden Sie unter: [Wie führe ich ein Upgrade von Redis OSS auf Valkey durch](#)

Themen

- [Wie führe ich ein Upgrade von Redis OSS auf Valkey durch](#)
- [Blockierte Valkey- oder Redis OSS-Engine-Upgrades lösen](#)

So ändern Sie Cluster und Replikationsgruppen

Caches	Replikationsgruppen
Unter Verwendung der ElastiCache AWS-Managementkonsole	Unter Verwendung der AWS-Managementkonsole
Verwenden Sie das mit AWS CLI ElastiCache	Verwenden des AWS CLI
Verwenden der ElastiCache API	Verwenden der API ElastiCache

Memcached

Um mit Memcached Versionsupgrades für Ihren Cluster zu starten, ändern Sie ihn und geben eine neuere Engine-Version an. Sie können dies tun, indem Sie die ElastiCache Konsole AWS CLI, die oder die ElastiCache API verwenden:

- Informationen zur Verwendung von AWS-Managementkonsole finden Sie unter —[Unter Verwendung der ElastiCache AWS-Managementkonsole](#).
- Informationen zur Verwendung von AWS CLI finden Sie unter[Verwenden Sie das mit AWS CLI ElastiCache](#).
- Informationen zur Verwendung der ElastiCache API finden Sie unter[Verwenden der ElastiCache API](#).

Wie führe ich ein Upgrade von Redis OSS auf Valkey durch

Valkey ist als Drop-In-Ersatz für Redis OSS 7 konzipiert. Sie können mithilfe der Konsole, API oder CLI ein Upgrade von Redis OSS auf Valkey durchführen, indem Sie die neue Engine und die Hauptversion der Engine angeben. Die IP-Adresse des Endpunkts und alle anderen Aspekte der Anwendung werden durch das Upgrade nicht geändert. Beim Upgrade von Redis OSS 5.0.6 und höher treten keine Ausfallzeiten auf.

Note

AWS Anforderungen an die CLI-Version für Upgrades von Redis OSS auf Valkey:

- Für AWS CLI v1: Erforderliche Mindestversion 1.35.2 (Aktuelle Version: 1.40.22)
- Für AWS CLI v2: Mindestens erforderliche Version 2.18.2 (Aktuelle Version: 2.27.22)

Note

- Bei einem Upgrade von früheren Redis OSS-Versionen als 5.0.6 kann es während der DNS-Propagierung zu einer Failover-Zeit von 30 bis 60 Sekunden kommen.
- Gehen Sie zunächst wie folgt vor, um einen vorhandenen Redis OSS-Single-Node-Cluster (Cluster-Modus deaktiviert) auf die Valkey-Engine zu aktualisieren: [Erstellen einer Replikationsgruppe unter Verwendung eines vorhandenen Clusters](#) Sobald der Redis OSS-Cluster (Cluster-Modus deaktiviert) mit einem Knoten zu einer Replikationsgruppe

hinzugefügt wurde, können Sie ein Engine-übergreifendes Upgrade auf Valkey durchführen.

Upgrade einer Replikationsgruppe von Redis OSS auf Valkey

Wenn Sie über eine bestehende Redis OSS-Replikationsgruppe verfügen, die die Standard-Cache-Parametergruppe verwendet, können Sie ein Upgrade auf Valkey durchführen, indem Sie die neue Engine und die Engine-Version mit der API angeben. `modify-replication-group`

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --engine valkey \  
  --engine-version 8.0
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --engine valkey ^  
  --engine-version 8.0
```

Wenn Sie eine benutzerdefinierte Cache-Parametergruppe auf die bestehende Redis OSS-Replikationsgruppe angewendet haben, die Sie aktualisieren möchten, müssen Sie in der Anfrage auch eine benutzerdefinierte Valkey-Cache-Parametergruppe übergeben. Die benutzerdefinierte Valkey-Eingabeparametergruppe muss dieselben statischen Redis OSS-Parameterwerte haben wie die vorhandene benutzerdefinierte Redis OSS-Parametergruppe.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --engine valkey \  
  --engine-version 8.0 \  
  --cache-parameter-group-name myParamGroup
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --engine valkey ^  
  --engine-version 8.0 ^  
  --cache-parameter-group-name myParamGroup
```

Aktualisieren eines serverlosen Redis OSS-Caches auf Valkey mit der CLI

Für Linux, macOS oder Unix:

```
aws elasticache modify-serverless-cache \  
  --serverless-cache-name myCluster \  
  --engine valkey \  
  --major-engine-version 8
```

Für Windows:

```
aws elasticache modify-serverless-cache ^  
  --serverless-cache-name myCluster ^  
  --engine valkey ^  
  --major-engine-version 8
```

Redis OSS mit der Konsole auf Valkey aktualisieren

Upgrade von Redis OSS 5 auf Valkey

1. Wählen Sie den Redis OSS-Cache für das Upgrade aus.
2. Ein Fenster „Upgrade to Valkey“ sollte angezeigt werden. Wählen Sie die Schaltfläche Auf Valkey aktualisieren.
3. Gehen Sie zu den Cache-Einstellungen und wählen Sie dann Engine-Version aus. Die neueste Version von Valkey wird empfohlen.
4. Wenn dieser Cache serverlos ist, müssen Sie die Parametergruppe aktualisieren. Gehen Sie in den Cache-Einstellungen zum Bereich Parametergruppen und wählen Sie eine entsprechende Parametergruppe aus, z. B. default.valkey8.
5. Wählen Sie Upgrade aus.

Dieser Cache wird nun im Valkey-Bereich der Konsole aufgeführt.

 Note

Ein direktes Upgrade von Redis OSS 4 oder niedriger auf Valkey kann eine längere Failover-Zeit von 30 bis 60 Sekunden während der DNS-Propagierung beinhalten.

Wie führe ich ein Downgrade von Valkey auf Redis OSS durch

Wenn Sie aus irgendeinem Grund Ihren aktualisierten Cluster zurücksetzen möchten, ElastiCache unterstützt Amazon das Rollback eines Valkey 7.2-Caches auf Redis OSS 7.1. Sie können ein Rollback mit denselben Konsolen-, API- oder CLI-Schritten wie bei einem Engine-Upgrade durchführen und Redis OSS 7.1 als Ziel-Engine-Version angeben. Rollbacks verwenden dieselben Prozesse wie ein Upgrade. Die IP-Adresse des Endpunkts und alle anderen Aspekte der Anwendung werden durch das Rollback nicht geändert, und es treten keine Ausfallzeiten auf.

Darüber hinaus können Sie einen aus Ihrem Valkey 7.2-Cache erstellten Snapshot als Redis OSS 7.1-Cache wiederherstellen. Wenn Sie aus einem Snapshot wiederherstellen, können Sie Redis OSS 7.1 als Ziel-Engine-Version angeben. Wenn Sie diese Option verwenden, wird aus dem Snapshot ein neuer Cache erstellt. Die Wiederherstellung aus einem Snapshot hat keine Auswirkung auf den Valkey-Cache, aus dem der Snapshot erstellt wurde.

Bei der Durchführung eines Rollbacks gelten die folgenden Anforderungen und Einschränkungen:

- ElastiCache unterstützt nur das Rollback von Valkey 7.2 auf Redis OSS 7.1. Dies gilt auch dann, wenn Sie von einer früheren Version als Redis OSS 7.1 auf Valkey 7.2 aktualisiert haben.
- Jede Benutzergruppe und jeder Benutzer, die der Replikationsgruppe oder dem serverlosen Cache zugeordnet sind, für den ein Rollback ausgeführt wird, müssen mit dem Engine-Typ konfiguriert werden. REDIS

Blockierte Valkey- oder Redis OSS-Engine-Upgrades lösen

Wie in der folgenden Tabelle dargestellt, ist Ihr Valkey- oder Redis OSS-Engine-Upgrade-Vorgang blockiert, wenn ein Scale-Up-Vorgang aussteht.

Ausstehende Vorgänge	Blockierte Vorgänge
Aufwärtsskalierung	Unmittelbares Engine-Upgrade

Ausstehende Vorgänge	Blockierte Vorgänge
Engine-Upgrade	Unmittelbares Aufwärtsskalieren
Aufwärtsskalierung und Engine-Upgrade	Unmittelbares Aufwärtsskalieren
	Unmittelbares Engine-Upgrade

Um ein blockiertes Valkey- oder Redis OSS-Engine-Upgrade zu beheben

- Führen Sie eine der folgenden Aktionen aus:
 - Planen Sie Ihren Redis OSS- oder Valkey-Engine-Upgrade-Vorgang für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Sofort anwenden deaktivieren.

Mit der CLI verwenden Sie `--no-apply-immediately`. Mit der API verwenden Sie `ApplyImmediately=false`.

- Warten Sie bis zum nächsten Wartungsfenster (oder danach), um Ihr Redis OSS-Engine-Upgrade durchzuführen.
- Fügen Sie dieser Cluster-Änderung den Redis OSS-Scale-Up-Vorgang hinzu, indem Sie das Kontrollkästchen Sofort anwenden aktivieren.

Mit der CLI verwenden Sie `--apply-immediately`. Mit der API verwenden Sie `ApplyImmediately=true`.

Dieser Ansatz bricht das Engine-Upgrade während des nächsten Wartungsfensters effektiv ab, indem es direkt ausgeführt wird.

ElastiCache Erweiterter Support

Mit ElastiCache Extended Support können Sie Ihren Cache gegen Aufpreis auch nach Ablauf des Standard-Supportdatums auf einer Hauptversion der Engine weiter ausführen. Wenn Sie nach Ablauf des Standard-Supportdatums kein Upgrade durchführen, wird Ihnen eine Gebühr berechnet.

Extended Support bietet die folgenden Updates und technischen Support:

- Sicherheitsupdates für kritische und hohe Sicherheitsaktualisierungen CVEs für Ihren Cache und Ihre Cache-Engine

- Bugfixes und Patches für kritische Probleme
- Die Möglichkeit, im Rahmen des standardmäßigen ElastiCache Service Level Agreements Supportanfragen zu eröffnen und Hilfe bei der Fehlerbehebung zu erhalten

Dieses kostenpflichtige Angebot gibt Ihnen mehr Zeit für ein Upgrade auf eine unterstützte Hauptversion der Engine.

Beispielsweise ElastiCache endet der Standardsupport für Redis OSS 4.0.10 am 31. Januar 2026. Wenn Sie bis zu diesem Datum nicht bereit sind, manuell auf Valkey oder auf Redis OSS 6 oder höher zu aktualisieren, ElastiCache werden Ihre Caches automatisch für Extended Support registriert und Sie können Redis OSS 4.0.10 weiterhin ausführen. Ab dem ersten Tag des Monats nach Ablauf des Standard-Supports, dem 1. Februar 2026, ElastiCache wird Ihnen automatisch der erweiterte Support in Rechnung gestellt.

Extended Support ist bis zu 3 Jahre nach Ablauf des Standard-Supportdatums für eine Hauptversion der Engine verfügbar. Für die OSS-Versionen 4 und 5 von ElastiCache für Redis ist das der 31. Januar 2029. Nach diesem Datum werden alle Caches, auf denen die Redis OSS-Versionen 4 und 5 noch ausgeführt werden, automatisch auf die neueste Version von Valkey aktualisiert.

Sobald der Supportzeitraum einer Engine endet, werden Caches, auf denen die alte Version weiterhin ausgeführt wird, automatisch auf Extended Support umgestellt. Sie werden vor dem Startdatum der Preise für den erweiterten Support benachrichtigt, sodass Sie stattdessen ein Upgrade Ihrer Instance durchführen können. Sie können sich auch jederzeit ausdrücklich abmelden, indem Sie ein Upgrade auf unterstützte Versionen durchführen.

Weitere Informationen zum Ende der Standard-Supportdaten und zum Ende der Extended Support-Daten finden Sie unter [ElastiCache Zeitplan für das Ende der Lebensdauer von Versionen für Redis OSS](#) für Valkey, Memcached oder Redis OSS.

Themen

- [ElastiCache Gebühren für erweiterten Support](#)
- [Versionen mit ElastiCache erweitertem Support](#)
- [ElastiCache und Kundenverantwortung mit ElastiCache Extended Support](#)

ElastiCache Gebühren für erweiterten Support

Ab dem Tag nach dem Ende des Standardsupports fallen Gebühren für alle Engines an, die für den ElastiCache erweiterten Support registriert sind. Informationen zum ElastiCache Ende des Standard-Supports finden Sie unter [Versionen mit ElastiCache erweitertem Support](#)

Die zusätzliche Gebühr für den ElastiCache erweiterten Support endet automatisch, wenn Sie eine der folgenden Maßnahmen ergreifen:

- Führen Sie ein Upgrade auf eine Engine-Version durch, für die der Standardsupport gilt.
- Löschen Sie den Cache, auf dem eine Hauptversion nach ElastiCache Ablauf des Standard-Supportdatums ausgeführt wird.

Die Gebühren werden wieder aufgenommen, wenn Ihre Ziel-Engine-Version in future in den erweiterten Support aufgenommen wird.

Nehmen wir zum Beispiel an, dass ElastiCache Version 4 für Redis OSS am 1. Februar 2026 in den erweiterten Support aufgenommen wird und Sie Ihre Caches von v4 am 1. Januar 2027 auf v6 aktualisieren. Bei ElastiCache Version 4 für Redis OSS werden Ihnen nur 11 Monate Extended Support in Rechnung gestellt. Wenn Sie ElastiCache Version 6 für Redis OSS nach dem Ende des Standard-Supports am 31. Januar 2027 weiter ausführen, fallen für diese Caches ab dem 1. Februar 2027 erneut Gebühren für den erweiterten Support an.

Sie können vermeiden, dass ElastiCacheExtended Support in Rechnung gestellt wird, indem Sie ElastiCache verhindern, dass nach ElastiCache Ablauf des Standard-Supportdatums ein Cache erstellt oder wiederhergestellt wird.

Weitere Informationen finden Sie unter [ElastiCache Amazon-Preise](#).

Versionen mit ElastiCache erweitertem Support

Die Versionen 4 und 5 von Redis Open Source Software (OSS) erreichten 2020 bzw. 2022 das Ende der Nutzungsdauer ihrer Community. Dies bedeutet, dass keine weiteren Updates, Bugfixes oder Sicherheitspatches von der Community veröffentlicht werden. Der Standardsupport für die ElastiCache Redis OSS-Versionen 4 und 5 ElastiCache endet am 31. Januar 2026. Wenn Sie weiterhin nicht unterstützte Versionen von Redis OSS verwenden, könnten Ihre Daten anfällig für [bekannte Sicherheitslücken und Sicherheitslücken](#) () werden. CVEs

Ab dem 1. Februar 2026 werden ElastiCache Caches, die noch auf den Redis OSS-Versionen 4 und 5 laufen, automatisch für Extended Support registriert, um kontinuierliche Verfügbarkeit und

Sicherheit zu gewährleisten. Extended Support bietet zwar Flexibilität, wir empfehlen jedoch, das Ende des Standard-Supports als Planungsmeilenstein für Ihre Produktionsworkloads zu betrachten. Wir empfehlen Ihnen dringend, Ihre Redis OSS v4- und v5-Caches vor dem Ende des ElastiCache Standardsupports auf Valkey oder Redis OSS v6 oder höher zu aktualisieren.

In der folgenden Tabelle sind das Datum für das ElastiCache Ende des Standard-Supports und das Datum des erweiterten Supports von Amazon zusammengefasst.

Erweiterter Support und Zeitplan für das Ende der Nutzungsdauer

Hauptversion der Engine	Ende des Standard-Supports	Beginn des erweiterten Support Y1 Premium	Beginn des erweiterten Support Y2 Premium	Beginn des erweiterten Support Y3 Premium	Ende des erweiterten Support und Versions-EOL
Redis OSS v4	31.1.2026	1.2.2026	1.2.2027	1.2.2028	31.1.2029
Redis OSS v5	31.1.2026	1.2.2026	1.2.2027	1.2.2028	31.1.2029
Redis OSS v6	31.1.2027	1.2.2027	1.2.2028	1.2.2029	31.1.2030

Erweiterter Support wird nur für die neueste unterstützte Patch-Version jeder Hauptversion von Redis OSS angeboten. Wenn der erweiterte Support am 1. Februar 2026 beginnt und Ihre Redis OSS v4- und v5-Cluster noch nicht auf den neuesten Patch-Versionen installiert sind, werden sie automatisch auf v4.0.10 für Redis OSS v4 und v5.0.6 für Redis OSS v5 aktualisiert, bevor sie für Extended Support registriert werden. Dadurch wird sichergestellt, dass Sie Sicherheitsupdates und Bugfixes über den erweiterten Support erhalten. Sie müssen keine Maßnahmen ergreifen, um im Rahmen der Umstellung auf den erweiterten Support ein Upgrade auf diese neuesten Patch-Versionen durchzuführen.

ElastiCache und Kundenverantwortung mit ElastiCache Extended Support

Im Folgenden sind die Aufgaben von Amazon ElastiCache und Ihre Verantwortlichkeiten mit ElastiCache Extended Support aufgeführt.

ElastiCache Verantwortlichkeiten von Amazon

Nach ElastiCache Ablauf des Standard-Supportdatums ElastiCache stellt Amazon Patches, Bugfixes und Upgrades für Engines bereit, die für den ElastiCache erweiterten Support registriert sind. Dies gilt für bis zu 3 Jahre oder bis Sie die Engines im Extended Support nicht mehr verwenden, je nachdem, was zuerst eintritt.

Ihre Aufgaben

Sie sind dafür verantwortlich, die Patches, Bugfixes und Upgrades zu installieren, die für Caches im ElastiCache Extended Support bereitgestellt werden. Amazon ElastiCache behält sich das Recht vor, solche Patches, Bugfixes und Upgrades jederzeit zu ändern, zu ersetzen oder zurückzuziehen. Wenn ein Patch erforderlich ist, um Sicherheits- oder kritische Stabilitätsprobleme zu beheben, ElastiCache behält sich Amazon das Recht vor, Ihre Caches mit dem Patch zu aktualisieren oder zu verlangen, dass Sie den Patch installieren.

Sie sind auch dafür verantwortlich, Ihre Engine vor ElastiCache Ablauf des Extended Support-Datums auf eine neuere Engine-Version zu aktualisieren. Das ElastiCache Ende des erweiterten Support liegt in der Regel 3 Jahre nach dem ElastiCache Ende des Standard-Supportdatums.

Wenn Sie Ihren Motor nicht aufrüsten, versucht Amazon ElastiCache nach ElastiCache Ablauf des Extended Support-Datums, Ihren Motor auf eine neuere Engine-Version aufzurüsten, die im Rahmen des ElastiCache Standardsupports unterstützt wird. Wenn das Upgrade fehlschlägt, ElastiCache behält sich Amazon das Recht vor, den Cache zu löschen, in dem die Engine nach ElastiCache Ablauf des Standard-Supportdatums läuft. Vorher bewahrt Amazon ElastiCache jedoch Ihre Daten aus dieser Engine auf.

Versionsverwaltung für ElastiCache

Sie können festlegen, wie Sie Ihre ElastiCache Caches und knotenbasierten Cluster aktualisieren möchten, die für die Valkey-, Memcached- und Redis OSS-Engines aktualisiert wurden.

ElastiCache Versionsverwaltung für Serverless Cache

Verwalten Sie, ob und wann der ElastiCache Serverless Cache aktualisiert wird, und führen Sie Versionsupgrades zu Ihren eigenen Bedingungen und Zeitplänen durch.

ElastiCache Serverless wendet automatisch die neueste Minor- und Patch-Softwareversion auf Ihren Cache an, ohne dass dies Auswirkungen oder Ausfallzeiten auf Ihre Anwendung hat. Von Ihrer Seite aus ist keine Aktion erforderlich.

Wenn eine neue Hauptversion verfügbar ist, sendet Ihnen ElastiCache Serverless eine Benachrichtigung in der Konsole und ein Ereignis in EventBridge. Sie können Ihren Cache auf die neueste Hauptversion aktualisieren, indem Sie ihn mithilfe der Konsole, CLI oder API ändern und die neueste Engine-Version auswählen. Ähnlich wie bei kleineren Upgrades und Patch-Upgrades werden Hauptversions-Upgrades ohne Ausfallzeiten Ihrer Anwendung durchgeführt.

Versionsverwaltung für knotenbasierte Cluster ElastiCache

Wenn Sie mit knotenbasierten ElastiCache Clustern arbeiten, können Sie steuern, wann die Software, die Ihren Cluster unterstützt, auf neue Versionen aktualisiert wird, die von unterstützt werden. ElastiCache Sie können steuern, wann Ihr Cache auf die neuesten verfügbaren Haupt-, Neben- und Patch-Versionen aktualisiert werden soll. Sie starten Engine-Versions-Upgrades für Ihren Cluster oder Ihre Replikationsgruppe, indem Sie ihn bzw. sie ändern und eine neue Engine-Version angeben.

Sie können steuern, ob und wann die protokollkonforme Software, die Ihren Cluster unterstützt, auf neue Versionen aktualisiert wird, die von unterstützt werden. ElastiCache Mit diesem Maß an Kontrolle können Sie die Kompatibilität mit bestimmten Versionen aufrechterhalten, neue Versionen mit Ihrer Anwendung testen, bevor Sie sie für die Produktion bereitstellen, und Versions-Upgrades nach Ihren eigenen Vorgaben und Zeitplänen durchführen lassen.

Da Versions-Upgrades ggf. mit Kompatibilitätsrisiken verbunden sind, können sie nicht automatisch eingestellt werden. Sie müssen von Ihnen selbst installiert werden.

Valkey- und Redis OSS-Cluster

Note

- Wenn ein Valkey- oder Redis OSS-Cluster in einer oder mehreren Regionen repliziert wird, wird die Engine-Version für sekundäre Regionen und dann für die primäre Region aktualisiert.
- ElastiCache für Redis werden OSS-Versionen mit einer semantischen Version identifiziert, die eine Haupt- und eine Nebekomponente umfasst. In Redis OSS 6.2 ist die Hauptversion beispielsweise 6 und die Nebenversion 2. Beim Betrieb knotenbasierter Cluster wird ElastiCache für Redis OSS auch die Patch-Komponente verfügbar gemacht, z. B. Redis OSS 6.2.1, und die Patch-Version ist 1.

Hauptversionen sind für API-inkompatible Änderungen und Nebenversionen für neue Funktionen vorgesehen, die abwärtskompatibel hinzugefügt wurden. Patch-Versionen sind für abwärtskompatible Bugfixes und nicht funktionale Änderungen vorgesehen.

Mit Valkey und Redis OSS initiieren Sie Engine-Versions-Upgrades für Ihren Cluster oder Ihre Replikationsgruppe, indem Sie ihn ändern und eine neue Engine-Version angeben. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

Memcached

Bei Memcached müssen Sie für ein Upgrade auf eine neuere Version Ihren Cluster ändern und die neue Engine-Version angeben, die Sie verwenden möchten. Das Upgrade auf eine neuere Memcached-Version ist ein destruktiver Prozess – Sie verlieren Ihre Daten und beginnen mit einem kalten Cache. Weitere Informationen finden Sie unter [Einen ElastiCache Cluster ändern](#).

Beim Upgraden einer älteren Memcached-Version auf Memcached Version 1.4.33 oder höher sind folgende Anforderungen zu beachten. `CreateCacheCluster` und `ModifyCacheCluster` schlagen unter den folgenden Bedingungen fehl:

- Wenn `slab_chunk_max > max_item_size`.
- Wenn `max_item_size modulo slab_chunk_max != 0`.
- Wenn `max_item_size > ((max_cache_memory - memcached_connections_overhead) / 4)`.

Der Wert `(max_cache_memory - memcached_connections_overhead)` ist der für Daten nutzbare Speicher des Knotens. Weitere Informationen finden Sie unter [Overhead von Memcached-Verbindungen](#).

Unterstützte Engines und Versionen

ElastiCache Serverlose Caches unterstützen ElastiCache Version 7.2 für Valkey und höher, ElastiCache Version 1.6 für Memcached und höher und ElastiCache 7.0 für Redis OSS und höher.

Knotenbasierte ElastiCache Cluster unterstützen ElastiCache Version 7.2 für Valkey und höher, ElastiCache Version 1.4.5 für Memcached und höher und 4.0.10 für Redis OSS und höher.

ElastiCache

Knotenbasierte ElastiCache Cluster unterstützen die folgenden Valkey-Versionen:

- [Unterstützte Valkey-Versionen](#)
- [ElastiCache Version 8.2 für Valkey](#)
- [ElastiCache Version 8.1 für Valkey](#)
- [ElastiCache Version 8.0 für Valkey](#)
- [ElastiCache Version 7.2.6 für Valkey](#)

Unterstützte Valkey-Versionen

Nachfolgend finden Sie die unterstützten Valkey-Versionen. Beachten Sie, dass Valkey standardmäßig die meisten Funktionen unterstützt, die in ElastiCache Version 7.2 für Redis OSS verfügbar sind.

- Sie können Ihre ElastiCache Cluster auch mit Versionen vor 5.0.6 aktualisieren. Der Prozess ist identisch, kann jedoch längere Failover-Zeit während der DNS-Ausbreitung (30 Sek. - 1 Min.) verursachen.
- Ab Redis OSS 7 wird ElastiCache das Umschalten zwischen Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert) unterstützt.
- Der Upgrade-Prozess ElastiCache für die Amazon for Redis OSS-Engine ist darauf ausgelegt, Ihre vorhandenen Daten bestmöglich beizubehalten, und erfordert eine erfolgreiche Redis OSS-Replikation.
- Beim Upgrade der Engine ElastiCache werden bestehende Client-Verbindungen beendet. [Um Ausfallzeiten bei Engine-Upgrades zu minimieren, empfehlen wir Ihnen, bewährte Methoden für Redis OSS-Clients mit Fehlerwiederholungen und exponentiellem Backoff sowie bewährte Methoden zur Minimierung von Ausfallzeiten während der Wartung zu implementieren.](#)
- Sie können beim Upgrade Ihrer Engine kein direktes Upgrade von Valkey oder Redis OSS (Clustermodus deaktiviert) auf Valkey oder Redis OSS (Clustermodus aktiviert) durchführen. Das folgende Verfahren zeigt Ihnen, wie Sie von Valkey oder Redis OSS (Clustermodus deaktiviert) auf Valkey oder Redis OSS (Clustermodus aktiviert) aktualisieren.

Um ein Upgrade von einer Valkey- oder Redis OSS-Engine-Version (Cluster-Modus deaktiviert) auf eine Valkey- oder Redis OSS-Engine-Version (Clustermodus aktiviert) durchzuführen

1. Erstellen Sie eine Sicherungskopie Ihres Valkey- oder Redis OSS-Clusters oder Ihrer Replikationsgruppe (Clustermodus deaktiviert). Weitere Informationen finden Sie unter [Erstellen manueller Backups](#).

2. Verwenden Sie das Backup, um einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mit einem Shard (Knotengruppe) zu erstellen und zu starten. Geben Sie die neue Engine-Version an und aktivieren Sie den Cluster-Modus, wenn Sie den Cluster oder die Replikationsgruppe erstellen. Weitere Informationen finden Sie unter [Tutorial: Seeding eines neuen knotenbasierten Clusters mit einem extern erstellten Backup](#).
 3. Löschen Sie den alten Valkey- oder Redis OSS-Cluster oder die alte Replikationsgruppe (Clustermodus deaktiviert). Für weitere Informationen siehe [Löschen eines Clusters in ElastiCache](#) oder [Löschen einer Replikationsgruppe](#).
 4. Skalieren Sie den neuen Valkey- oder Redis OSS-Cluster oder die neue Replikationsgruppe (Clustermodus aktiviert) auf die Anzahl der Shards (Knotengruppen), die Sie benötigen. Weitere Informationen finden Sie unter [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#).
- Beim Upgrade von Hauptversionen der Engine, beispielsweise von 5.0.6 auf 6.0, müssen Sie auch eine neue Parametergruppe auswählen, die mit der neuen Engine-Version kompatibel ist.
 - Für einzelne Redis OSS-Cluster und Cluster mit deaktiviertem Multi-AZ empfehlen wir, Redis OSS ausreichend Speicher zur Verfügung zu stellen, wie unter beschrieben. [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#)
In diesen Fällen steht der primäre Knoten während des Upgrade-Prozesses für Serviceanfragen nicht zur Verfügung.
 - Für Redis OSS-Cluster mit aktiviertem Multi-AZ empfehlen wir außerdem, Engine-Upgrades in Zeiten mit geringem eingehendem Schreibverkehr zu planen. Bei einem Upgrade auf Redis OSS 5.0.6 oder höher steht der primäre Cluster während des Upgrade-Vorgangs weiterhin für Serviceanfragen zur Verfügung.

Cluster und Replikationsgruppen mit mehreren Shards werden wie folgt verarbeitet und gepatcht:

- Alle Shards werden parallel verarbeitet. Es wird jeweils nur eine Upgrade-Operation für einen Shard gleichzeitig durchgeführt.
- In jedem Shard werden alle Replicas verarbeitet, bevor der Primärknoten verarbeitet wird. Wenn es in einem Shard weniger Replicas gibt, kann der Primärknoten in diesem Shard verarbeitet werden, bevor die Verarbeitung der Replicas in anderen Shards abgeschlossen wird.
- Die Primärknoten für alle Shards werden seriell verarbeitet. Es erfolgt jeweils nur ein Upgrade für einen Primärknoten gleichzeitig.
- Wenn die Verschlüsselung in Ihrem aktuellen Cluster oder Ihrer Replikationsgruppe aktiviert ist, können Sie nicht auf eine Engine-Version aktualisieren, die keine Verschlüsselung unterstützt, z. B. von 3.2.6 auf 3.2.10.

Überlegungen zu Memcached

Beachten Sie beim Upgrade eines knotenbasierten Memcached-Clusters Folgendes.

- Versionsverwaltung der Engine ist so entwickelt, dass Sie so viel Kontrolle wie möglich darüber haben, wie Patchen erfolgt. Behält sich jedoch ElastiCache das Recht vor, Ihren Cluster in Ihrem Namen zu patchen, sollte der unwahrscheinliche Fall eintreten, dass das System oder die Cache-Software eine kritische Sicherheitslücke aufweist.
- Da die Memcached-Engine keine Persistenz unterstützt, stellen Versions-Upgrades der Memcached-Engine immer einen Störfall dar, bei dem alle Cache-Daten im Cluster gelöscht werden.

ElastiCache Version 8.2 für Valkey

Hier sind einige der neuen Funktionen, die in Valkey 8.2 eingeführt wurden (im Vergleich zu ElastiCache Valkey 8.1):

- Native Unterstützung für [Vector Search](#), mit der Sie Milliarden von hochdimensionalen Vektoreinbettungen im Speicher speichern, indexieren, suchen und aktualisieren können, mit Latenzen von nur Mikrosekunden.

[Weitere Informationen zu Valkey finden Sie unter Valkey.](#)

[Weitere Informationen zur Version Valkey 8.2, in der die Vektorsuche eingeführt wird, finden Sie unter Valkey Search.](#)

ElastiCache Version 8.1 für Valkey

Hier sind einige der neuen Funktionen, die in Valkey 8.1 eingeführt wurden (im Vergleich zu ElastiCache Valkey 8.0):

- Eine [neue Hash-Tabellenimplementierung](#), die den Speicheraufwand reduziert, um den Speicherverbrauch für gängige key/value Muster um bis zu 20% zu senken.
- Native Unterstützung für [Bloom-Filter](#), einen neuen Datentyp, mit dem Sie Suchvorgänge mit bis zu 98% weniger Speicher durchführen können als mit dem Set-Datentyp.
- Neuer Befehl [COMMANDLOG](#), der langsame Ausführungen, umfangreiche Anfragen und umfangreiche Antworten aufzeichnet.
- Neue Unterstützung für bedingte Updates des SET-Befehls mithilfe des IFEQ-Arguments.

- Leistungsverbesserungen, darunter eine um bis zu 45% geringere Latenz für den Befehl ZRANK, eine bis zu 12-mal schnellere Leistung für PFMERGE und PFCOUNT und ein um bis zu 514% höherer Durchsatz für BITCOUNT.

[Weitere Informationen zu Valkey finden Sie unter Valkey](#)

Weitere Informationen zur Version Valkey 8.1 finden Sie in den Versionshinweisen zu [Valkey 8.1](#)

ElastiCache Version 8.0 für Valkey

Hier sind einige der neuen Funktionen, die in Valkey 8.0 eingeführt wurden (im Vergleich zu ElastiCache Valkey 7.2.6):

- Verbesserungen der Speichereffizienz, sodass Benutzer bis zu 20% mehr Daten pro Knoten speichern können, ohne dass Änderungen an der Anwendung vorgenommen werden müssen.
- Neu eingeführte Metrikinfrastruktur pro Steckplatz für knotenbasierte Cluster, die detaillierte Einblicke in die Leistung und den Ressourcenverbrauch einzelner Steckplätze bietet.
- ElastiCache Serverless for Valkey 8.0 kann die Anzahl der unterstützten Anfragen pro Sekunde (RPS) alle 2 bis 3 Minuten verdoppeln, sodass 5 Millionen RPS pro Cache in weniger als 13 Minuten von Null erreicht werden, und das bei gleichbleibender p50-Leselatenz von unter einer Millisekunde.

[Weitere Informationen zu Valkey finden Sie unter Valkey](#)

Weitere Informationen zur Valkey 8-Version finden Sie in den Versionshinweisen zu [Valkey 8](#)

ElastiCache Version 7.2.6 für Valkey

Am 10. Oktober 2024 ElastiCache wurde Version 7.2.6 für Valkey veröffentlicht. Hier sind einige der neuen Funktionen, die in 7.2 eingeführt wurden (im Vergleich zu ElastiCache Version 7.1 für Redis OSS):

- Leistungs- und Speicheroptimierungen für verschiedene Datentypen: Speicheroptimierung für Tasten vom Typ List und Set, Geschwindigkeitsoptimierung für Befehle mit sortierten Sets, Leistungsoptimierung für Befehle mit mehreren Tasten im Clustermodus, pub/sub Leistungsverbesserungen, Leistungsoptimierung für SCAN-, SSCAN-, HSCAN-, ZSCAN-Befehle und zahlreiche andere kleinere Optimierungen.
- Neue WITHSCORE-Option für die Befehle ZRANK und ZREVRANK

- CLIENT NO-TOUCH ermöglicht es Clients, Befehle auszuführen, ohne die Tasten zu beeinträchtigen. LRU/LFU
- Neuer Befehl CLUSTER MYSHARDID, der die Shard-ID des Knotens zurückgibt, um Knoten im Clustermodus auf der Grundlage der Replikation logisch zu gruppieren.

[Weitere Informationen zu Valkey finden Sie unter Valkey](#)

Weitere Informationen zur ElastiCache Version 7.2 für Valkey finden Sie in den [Versionshinweisen zu Redis OSS 7.2.4](#) (ElastiCache Version 7.2 für Valkey enthält alle Änderungen von Version 7.1 für Redis OSS bis ElastiCache ElastiCache Version 7.2.4 für Redis OSS). Versionshinweise zu Valkey 7.2 bei [Valkey](#) am. GitHub

ElastiCache Version 8.2 für Valkey

Hier sind einige der neuen Funktionen, die in Valkey 8.2 eingeführt wurden (im Vergleich zu ElastiCache Valkey 8.1):

- Native Unterstützung für [Vector Search](#), mit der Sie Milliarden von hochdimensionalen Vektoreinbettungen im Speicher speichern, indexieren, suchen und aktualisieren können, mit Latenzen von nur Mikrosekunden.

[Weitere Informationen zu Valkey finden Sie unter Valkey.](#)

[Weitere Informationen zur Version Valkey 8.2, in der die Vektorsuche eingeführt wird, finden Sie unter Valkey Search.](#)

ElastiCache Version 8.1 für Valkey

Hier sind einige der neuen Funktionen, die in Valkey 8.1 eingeführt wurden (im Vergleich zu ElastiCache Valkey 8.0):

- Eine [neue Hash-Tabellenimplementierung](#), die den Speicheraufwand reduziert, um den Speicherverbrauch für gängige key/value Muster um bis zu 20% zu senken.
- Native Unterstützung für [Bloom-Filter](#), einen neuen Datentyp, mit dem Sie Suchvorgänge mit bis zu 98% weniger Speicher durchführen können als mit dem Set-Datentyp.
- Neuer Befehl [COMMANDLOG](#), der langsame Ausführungen, umfangreiche Anfragen und umfangreiche Antworten aufzeichnet.

- Neue Unterstützung für bedingte Updates des SET-Befehls mithilfe des IFEQ-Arguments.
- Leistungsverbesserungen, darunter eine um bis zu 45% geringere Latenz für den Befehl ZRANK, eine bis zu 12-mal schnellere Leistung für PFMERGE und PFCOUNT und ein um bis zu 514% höherer Durchsatz für BITCOUNT.

[Weitere Informationen zu Valkey finden Sie unter Valkey](#)

Weitere Informationen zur Version Valkey 8.1 finden Sie in den Versionshinweisen zu [Valkey](#) 8.1

ElastiCache Version 8.0 für Valkey

Hier sind einige der neuen Funktionen, die in Valkey 8.0 eingeführt wurden (im Vergleich zu ElastiCache Valkey 7.2.6):

- Verbesserungen der Speichereffizienz, sodass Benutzer bis zu 20% mehr Daten pro Knoten speichern können, ohne dass Änderungen an der Anwendung erforderlich sind.
- Neu eingeführte Metrikinfrastruktur pro Steckplatz für knotenbasierte Cluster, die detaillierte Einblicke in die Leistung und den Ressourcenverbrauch einzelner Steckplätze bietet.
- ElastiCache Serverless for Valkey 8.0 kann die Anzahl der unterstützten Anfragen pro Sekunde (RPS) alle 2 bis 3 Minuten verdoppeln, sodass 5 Millionen RPS pro Cache in weniger als 13 Minuten erreicht werden, und das bei gleichbleibender p50-Leselatenz von unter einer Millisekunde.

[Weitere Informationen zu Valkey finden Sie unter Valkey](#)

Weitere Informationen zur Valkey 8-Version finden Sie in den Versionshinweisen zu [Valkey](#) 8

ElastiCache Version 7.2.6 für Valkey

Am 10. Oktober 2024 ElastiCache wurde Version 7.2.6 für Valkey veröffentlicht. Hier sind einige der neuen Funktionen, die in 7.2 eingeführt wurden (im Vergleich zu ElastiCache Version 7.1 für Redis OSS):

- Leistungs- und Speicheroptimierungen für verschiedene Datentypen: Speicheroptimierung für Tasten vom Typ List und Set, Geschwindigkeitsoptimierung für Befehle mit sortierten Sets, Leistungsoptimierung für Befehle mit mehreren Tasten im Clustermodus, pub/sub Leistungsverbesserungen, Leistungsoptimierung für SCAN-, SSCAN-, HSCAN-, ZSCAN-Befehle und zahlreiche andere kleinere Optimierungen.

- Neue WITHSCORE-Option für die Befehle ZRANK und ZREVRANK
- CLIENT NO-TOUCH ermöglicht es Clients, Befehle auszuführen, ohne die Tasten zu beeinträchtigen. LRU/LFU
- Neuer Befehl CLUSTER MYSHARDID, der die Shard-ID des Knotens zurückgibt, um Knoten im Clustermodus auf der Grundlage der Replikation logisch zu gruppieren.

[Weitere Informationen zu Valkey finden Sie unter Valkey](#)

Weitere Informationen zur ElastiCache Version 7.2 für Valkey finden Sie in den [Versionshinweisen zu Redis OSS 7.2.4](#) (ElastiCache Version 7.2 für Valkey enthält alle Änderungen von Version 7.1 für Redis OSS bis ElastiCache ElastiCache Version 7.2.4 für Redis OSS). Versionshinweise zu Valkey 7.2 bei [Valkey](#) am. GitHub

Unterstützte Redis OSS-Engine-Versionen

ElastiCache Serverlose Caches und knotenbasierte Cluster unterstützen alle Redis OSS-Versionen 7.1 und früher.

- [ElastiCache Version 7.1 für Redis OSS \(erweitert\)](#)

Knotenbasierte ElastiCache Cluster unterstützen die folgenden Redis OSS-Versionen:

- [ElastiCache Version 7.1 für Redis OSS \(erweitert\)](#)
- [ElastiCache Version 7.0 für Redis OSS \(erweitert\)](#)
- [ElastiCache Version 6.2 für Redis OSS \(erweitert\)](#)
- [ElastiCache Version 6.0 für Redis OSS \(erweitert\)](#)
- [ElastiCache Version 5.0.6 für Redis OSS \(erweitert\)](#)
- [ElastiCache Version 5.0.5 für Redis OSS \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [ElastiCache Version 5.0.4 für Redis OSS \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [ElastiCache Version 5.0.3 für Redis OSS \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [ElastiCache Version 5.0.0 für Redis OSS \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [ElastiCache Version 4.0.10 für Redis OSS \(erweitert\)](#)
- [Frühere Versionen für das Ende der Lebensdauer \(EOL\) \(3.x\)](#)
- [Frühere Versionen für das Ende der Lebensdauer \(EOL\) \(2.x\)](#)

ElastiCache Version 7.1 für Redis OSS (erweitert)

Diese Version enthält Leistungsverbesserungen, die es Workloads ermöglichen, einen höheren Durchsatz und geringere Betriebslatenzen zu erzielen. ElastiCache Version 7.1 für Redis OSS bietet [zwei](#) Hauptverbesserungen:

Wir haben die erweiterte I/O Thread-Funktionalität erweitert, um auch die Logik der Präsentationsebene zu handhaben. Mit Presentation Layer meinen wir die erweiterten I/O-Threads, die jetzt nicht nur Client-Eingaben lesen, sondern die Eingabe auch im Redis OSS-Binärbefehlsformat analysieren. Diese wird dann zur Ausführung an den Haupt-Thread weitergeleitet, was zu einer Leistungssteigerung führt. Verbessertes Redis OSS-Speicherzugriffsmuster. Die Ausführungsschritte vieler Datenstrukturoperationen sind miteinander verwoben, um Speicherzugriff und reduzierte Speicherzugriffslatenz parallel zu gewährleisten. Kunden, die ElastiCache auf Graviton3 R7g.4xlarge oder höher basieren, können über 1 Million Anfragen pro Sekunde und Knoten erreichen. Mit den Leistungsverbesserungen von Redis OSS v7.1 können Kunden im Vergleich zu ElastiCache Redis OSS v7.0 einen um bis zu 100% höheren Durchsatz und eine um 50% geringere P99-Latenz erzielen. ElastiCache Diese Verbesserungen sind bei Knotengrößen mit mindestens 8 physischen Kernen (2xlarge auf Graviton und 4xlarge auf x86) unabhängig vom CPU-Typ aktiviert und erfordern keine Client-Änderungen.

Note

ElastiCache v7.1 ist mit Redis OSS v7.0 kompatibel.

ElastiCache Version 7.0 für Redis OSS (erweitert)

ElastiCache für Redis OSS 7.0 bietet eine Reihe von Verbesserungen und Unterstützung für neue Funktionen:

- [Funktionen](#): ElastiCache für Redis OSS 7 bietet Unterstützung für Redis OSS-Funktionen und bietet eine verwaltete Oberfläche, die es Entwicklern ermöglicht, [LUA-Skripts](#) mit auf dem ElastiCache Cluster gespeicherter Anwendungslogik auszuführen, ohne dass Clients die Skripts bei jeder Verbindung erneut an den Server senden müssen.
- [ACL-Verbesserungen](#): Valkey und Redis OSS 7 bieten Unterstützung für die nächste Version von Access Control Lists (). ACLs Clients können jetzt mehrere Berechtigungssätze für bestimmte Schlüssel oder Schlüsselräume in Valkey und Redis OSS angeben.

- [Sharded Pub/Sub](#): ElastiCache Für Valkey und Redis bietet OSS 7 Unterstützung für Pub/Sub functionality in a sharded way when running ElastiCache in Cluster Mode Enabled (CME). Pub/Sub Run-Funktionen, mit denen Publisher Nachrichten an eine beliebige Anzahl von Abonnenten auf einem Kanal versenden können. Kanäle sind an einen Shard im ElastiCache Cluster gebunden, sodass Kanalinformationen nicht mehr über mehrere Shards verteilt werden müssen, was zu einer verbesserten Skalierbarkeit führt.
- Verbessertes I/O Multiplexing: ElastiCache Für Valkey und Redis führt OSS 7 ein erweitertes I/O-Multiplexing ein, das einen höheren Durchsatz und eine geringere Latenz für Workloads mit hohem Durchsatz und vielen gleichzeitigen Client-Verbindungen zu einem Cluster bietet. ElastiCache Wenn Sie beispielsweise einen Cluster von r6g.xlarge-Knoten verwenden und 5200 Clients gleichzeitig ausführen, können Sie im Vergleich zu ElastiCache Version 6 für Redis OSS einen um bis zu 72% höheren Durchsatz (Lese- und Schreibvorgänge pro Sekunde) und eine um bis zu 71% verringerte P99-Latenz erzielen.

[Weitere Informationen zu Valkey finden Sie unter Valkey](#). Weitere Informationen zur Version Redis OSS 7.0 finden Sie in den [Versionshinweisen zu Redis OSS 7.0 unter Redis OSS](#) auf GitHub

ElastiCache Version 6.2 für Redis OSS (erweitert)

ElastiCache für Redis OSS 6.2 beinhaltet Leistungsverbesserungen für TLS-fähige Cluster, die x86-Knotentypen mit 8 V CPUs oder mehr oder Graviton2-Knotentypen mit 4 V oder mehr verwenden. CPUs Diese Verbesserungen verbessern den Durchsatz und reduzieren die Zeit für den Verbindungsaufbau der Clients, indem die Verschlüsselung auf andere V verlagert wird. CPUs Mit Redis OSS 6.2 können Sie auch den Zugriff auf Pub/Sub-Kanäle mithilfe von ACL-Regeln (Access Control List) verwalten.

Mit dieser Version führen wir auch Unterstützung für Daten-Tiering auf Clusterknoten ein, die lokal angeschlossene SSDs enthalten. NVMe Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

Version 6.2.6 der Redis OSS Engine bietet auch Unterstützung für das native JSON-Format (JavaScript Object Notation), eine einfache, schemalose Methode zur Kodierung komplexer Datensätze in Redis OSS-Clustern. Mit der JSON-Unterstützung können Sie die Leistung und Redis OSS für Anwendungen nutzen, die über JSON arbeiten. APIs Weitere Informationen finden Sie unter [Erste Schritte mit JSON](#). Ebenfalls enthalten sind JSON-bezogene Metriken `JsonBasedCmds` und `JsonBasedCmdsLatency`, die CloudWatch zur Überwachung der Verwendung dieses Datentyps integriert sind. Weitere Informationen finden Sie unter [Metriken für Valkey und Redis OSS](#).

Sie geben die Engine-Version mit 6.2 an. ElastiCache ruft automatisch die bevorzugte Patch-Version von Redis OSS 6.2 auf, die verfügbar ist. Wenn Sie beispielsweise einen Cluster erstellen/ändern, setzen Sie den `--engine-version` Parameter auf 6.2. Der Cluster wird mit der jeweils aktuell verfügbaren bevorzugten Patch-Version von Redis OSS 6.2 gestartet. creation/modification Die Angabe der Engine-Version 6.x in der API führt zur neuesten Nebenversion von Redis OSS 6.

Für bestehende 6.0-Cluster können Sie sich für das nächste Autominor-Versions-Upgrade entscheiden, indem Sie den `AutoMinorVersionUpgrade` Parameter `yes` im `CreateCacheCluster`, `ModifyCacheCluster`, `CreateReplicationGroup` oder `ModifyReplicationGroup` APIs auf setzen. ElastiCache aktualisiert die Nebenversion Ihrer vorhandenen 6.0-Cluster mithilfe von Self-Service-Updates auf 6.2. Weitere Informationen finden Sie unter [Self-Service-Updates in Amazon ElastiCache](#).

Beim Aufrufen der `DescribeCacheEngineVersions` API wird der `EngineVersion` Parameterwert auf 6.2 gesetzt und die tatsächliche Engine-Version mit der Patch-Version wird im `CacheEngineVersionDescription` Feld zurückgegeben.

Weitere Informationen zur Version Redis OSS 6.2 finden Sie in den [Versionshinweisen zu Redis OSS 6.2](#) unter Redis OSS auf GitHub

ElastiCache Version 6.0 für Redis OSS (erweitert)

Amazon ElastiCache stellt die nächste Version der ElastiCache Redis OSS-Engine vor, die die [Authentifizierung von Benutzern mit rollenbasierter Zugriffskontrolle](#), clientseitiges Caching und erhebliche betriebliche Verbesserungen umfasst.

Ab Redis OSS 6.0 ElastiCache wird es eine einzige Version für jede Redis OSS-Nebenversion geben, anstatt mehrere Patch-Versionen anzubieten. ElastiCache verwaltet automatisch die Patch-Version Ihrer laufenden Cluster und sorgt so für eine verbesserte Leistung und erhöhte Sicherheit.

Sie können sich auch für das nächste auto Nebenversions-Upgrade anmelden, indem Sie den `AutoMinorVersionUpgrade` Parameter auf `yes` setzen. Das Upgrade der Nebenversion ElastiCache wird über Self-Service-Updates verwaltet. Weitere Informationen finden Sie unter [Service-Updates in ElastiCache](#).

Sie geben die Engine-Version an, indem Sie 6.0 ElastiCache ruft automatisch die bevorzugte Patch-Version von Redis OSS 6.0 auf, die verfügbar ist. Wenn Sie beispielsweise create/modify einen Cluster erstellen, setzen Sie den `--engine-version` Parameter auf 6.0. Der Cluster wird zum Zeitpunkt der Erstellung/Änderung mit der aktuell verfügbaren bevorzugten Patch-Version von Redis

OSS 6.0 gestartet. Jede Anfrage mit einem bestimmten Patch-Versionswert wird abgelehnt, eine Ausnahme wird ausgelöst und der Vorgang schlägt fehl.

Beim Aufrufen der DescribeCacheEngineVersions API wird der EngineVersion Parameterwert auf 6.0 gesetzt und die tatsächliche Engine-Version mit der Patch-Version wird im Feld zurückgegeben. CacheEngineVersionDescription

Weitere Informationen zur Version Redis OSS 6.0 finden Sie in den [Versionshinweisen zu Redis OSS 6.0](#) unter Redis OSS auf GitHub

ElastiCache Version 5.0.6 für Redis OSS (erweitert)

Amazon ElastiCache stellt die nächste Version von ElastiCache für die Redis OSS-Engine vor, die Fehlerkorrekturen und die folgenden kumulativen Updates enthält:

- Garantierte Engine-Stabilität in speziellen Bedingungen.
- Verbesserte Hyperloglog-Fehlerbehandlung.
- Erweiterte Handshake-Befehle für eine zuverlässige Replikation.
- Konsistente Nachverfolgung der Nachrichtenzustellung über den XCLAIM-Befehl.
- Verbesserte LFU -Feldverwaltung in Objekten.
- Verbesserte Transaktionsverwaltung bei Verwendung von ZPOP.
- Möglichkeit, Befehle umzubenennen: Ein Parameter namens `rename-commands`, der es Ihnen ermöglicht, potenziell gefährliche oder teure Redis OSS-Befehle umzubenennen, die zu versehentlichem Datenverlust führen könnten, wie z. B. `FLUSHALL` oder `FLUSHDB`. Dies ähnelt der Konfiguration mit `Rename-Commands` im Open-Source-Redis OSS. ElastiCache hat jedoch die Benutzererfahrung durch die Bereitstellung eines vollständig verwalteten Workflows verbessert. Die Befehlsnamensänderungen werden sofort übernommen und automatisch in allen Knoten im Cluster vorgenommen, die die Befehlsliste enthalten. Es ist kein Eingriff Ihrerseits erforderlich, wie etwa das Neustarten der Knoten.

Die folgenden Beispiele zeigen, wie Sie vorhandene Parametergruppen ändern. Sie enthalten den `rename-commands`-Parameter, eine durch Leerzeichen getrennte Liste von Befehlen, die Sie umbenennen möchten:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-  
name custom_param_group  
--parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall  
restrictedflushall'" --region region
```

In diesem Beispiel wird der `rename-commands`-Parameter zum Umbenennen des `flushall`-Befehls in `restrictedflushall` verwendet.

Zum Umbenennen mehrerer Befehle verwenden Sie Folgendes:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall flushdb restrictedflushdb'" --region region
```

Zum Zurücksetzen einer Änderung führen Sie den Befehl erneut aus und schließen sämtliche umbenannten Werte folgendermaßen von der `ParameterValue`-Liste aus, die Sie beibehalten möchten:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall'" --region region
```

In diesem Fall wird der `flushall`-Befehl in `restrictedflushall` umbenannt und sämtliche anderen umbenannten Befehle werden auf ihre ursprünglichen Befehlsnamen zurückgesetzt.

Note

Beim Umbenennen von Befehlen unterliegen Sie den folgenden Einschränkungen:

- Alle umbenannten Befehle müssen alphanumerisch sein.
- Die maximale Länge neuer Befehlsnamen beträgt 20 alphanumerische Zeichen.
- Beim Umbenennen von Befehlen muss sichergestellt werden, dass Sie die Parametergruppe aktualisieren, die mit Ihrem Cluster verknüpft ist.
- Um die Nutzung eines Befehls gänzlich zu verhindern, verwenden Sie das Schlüsselwort `blocked` folgendermaßen:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall blocked'" --region region
```

Weitere Informationen zu den Parameteränderungen und eine Liste der Befehle, die umbenannt werden können, finden Sie unter [Änderungen an den Parametern von Redis OSS 5.0.3](#).

- Redis OSS Streams: Dabei wird eine Protokolldatenstruktur modelliert, die es Produzenten ermöglicht, neue Elemente in Echtzeit anzuhängen. Nachrichten können auch blockierend oder blockierungsfrei abgerufen werden. Streams ermöglichen auch Konsumentengruppen als Gruppe von Clients, die zusammen unterschiedliche Teile desselben Nachrichten-Streams abrufen, ähnlich wie bei [Apache Kafka](#). [Weitere Informationen finden Sie unter Streams](#).
- Unterstützung für eine Reihe von Stream-Befehlen wie etwa XADD, XRANGE und XREAD. Weitere Informationen finden Sie unter [Streams-Befehle](#).
- Eine Anzahl neuer und umbenannter Parameter. Weitere Informationen finden Sie unter [Änderungen der Parameter von Redis OSS 5.0.0](#).
- Eine neue Redis-OSS-Metrik, `StreamBasedCmds`.
- Etwas schnellere Snapshot-Zeit für Redis OSS-Knoten.

Important

ElastiCache hat zwei kritische Bugfixes aus der [Open-Source-Version 5.0.1 von Redis OSS](#) zurückportiert. Sie sind im Folgenden aufgeführt:

- RESTORE-Antwort bei fehlender Übereinstimmung, wenn bestimmte Schlüssel bereits abgelaufen sind.
- Der Befehl XCLAIM kann einen falschen Eintrag zurückgeben oder die Synchronisierung des Protokolls verhindern.

Beide Bugfixes sind in der Redis OSS-Unterstützung ElastiCache für Redis OSS Engine Version 5.0.0 enthalten und werden in future Versionsupdates verwendet.

Weitere Informationen finden Sie in den [Versionshinweisen zu Redis OSS 5.0.6](#) unter Redis OSS unter. GitHub

ElastiCache Version 5.0.5 für Redis OSS (veraltet, verwenden Sie Version 5.0.6)

Amazon ElastiCache stellt die nächste Version von ElastiCache für die Redis OSS-Engine vor;. Es beinhaltet Online-Konfigurationsänderungen für ElastiCache Auto-Failover-Cluster während aller geplanten Operationen. Sie können jetzt Ihren Cluster skalieren, die Redis OSS-Engine-Version aktualisieren und Patches und Wartungsupdates anwenden, während der Cluster online bleibt und weiterhin eingehende Anfragen bearbeitet. Das beinhaltet auch Fehlerbehebungen.

Weitere Informationen finden Sie in den [Versionshinweisen zu Redis OSS 5.0.5](#) unter Redis OSS unter. GitHub

ElastiCache Version 5.0.4 für Redis OSS (veraltet, verwenden Sie Version 5.0.6)

Amazon ElastiCache stellt die nächste Version der Redis OSS-Engine vor, die von ElastiCache unterstützt wird. Die Version enthält die folgenden Erweiterungen:

- Garantierte Engine-Stabilität in speziellen Bedingungen.
- Verbesserte Hyperloglog-Fehlerbehandlung.
- Erweiterte Handshake-Befehle für eine zuverlässige Replikation.
- Konsistente Nachverfolgung der Nachrichtenzustellung über den XCLAIM-Befehl.
- Verbesserte LFU -Feldverwaltung in Objekten.
- Verbesserte Transaktionsverwaltung bei Verwendung von ZPOP.

Weitere Informationen finden Sie in den [Versionshinweisen zu Redis OSS 5.0.4](#) unter Redis OSS am. GitHub

ElastiCache Version 5.0.3 für Redis OSS (veraltet, verwenden Sie Version 5.0.6)

Amazon ElastiCache stellt die nächste Version von ElastiCache für die Redis OSS-Engine vor, die Fehlerkorrekturen enthält.

ElastiCache Version 5.0.0 für Redis OSS (veraltet, verwenden Sie Version 5.0.6)

Amazon ElastiCache stellt die nächste Hauptversion von ElastiCache für die Redis OSS-Engine vor. ElastiCache Version 5.0.0 für Redis OSS bietet Unterstützung für die folgenden Verbesserungen:

- Redis OSS Streams: Dies modelliert eine Protokolldatenstruktur, die es Produzenten ermöglicht, neue Elemente in Echtzeit anzuhängen. Nachrichten können auch blockierend oder

blockierungsfrei abgerufen werden. Streams ermöglichen auch Konsumentengruppen als Gruppe von Clients, die zusammen unterschiedliche Teile desselben Nachrichten-Streams abrufen, ähnlich wie bei [Apache Kafka](#). [Weitere Informationen finden Sie unter Streams](#).

- Unterstützung für eine Reihe von Stream-Befehlen wie etwa XADD, XRANGE und XREAD. Weitere Informationen finden Sie unter [Streams-Befehle](#).
- Eine Anzahl neuer und umbenannter Parameter. Weitere Informationen finden Sie unter [Änderungen der Parameter von Redis OSS 5.0.0](#).
- Eine neue Redis-OSS-Metrik, StreamBasedCmds.
- Etwas schnellere Snapshot-Zeit für Redis OSS-Knoten.

ElastiCache Version 4.0.10 für Redis OSS (erweitert)

Amazon ElastiCache stellt die nächste Hauptversion von ElastiCache für die Redis OSS-Engine vor. ElastiCache Version 4.0.10 für Redis OSS bietet Unterstützung für die folgenden Verbesserungen:

- Sowohl Online-Cluster-Größenänderung als auch Verschlüsselung in einer einzigen Version. ElastiCache Weitere Informationen finden Sie hier:
 - [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#)
 - [Online-Resharding für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
 - [Datensicherheit bei Amazon ElastiCache](#)
- Eine Anzahl neuer Parameter. Weitere Informationen finden Sie unter [Änderungen an den Parametern von Redis OSS 4.0.10](#).
- Unterstützung für eine Reihe von Speicherbefehlen wie etwa MEMORY. Weitere Informationen finden Sie unter [Befehle](#) (Suche auf MEMO).
- Unterstützung für Speicherdefragmentierung, während das System online ist, wodurch eine effizientere Speichernutzung ermöglicht wird und mehr Speicher für Ihre Daten zur Verfügung steht.
- Support für asynchrone Leerungen und Löschungen. ElastiCache für Redis unterstützt OSS Befehle wie UNLINK FLUSHDB und FLUSHALL zur Ausführung in einem anderen Thread als dem Haupt-Thread. Dies hilft, Leistung und Reaktionszeiten für Ihre Anwendungen zu verbessern, indem Speicher asynchron freigegeben wird.
- Eine neue Redis OSS-Metrik, ActiveDefragHits Weitere Informationen finden Sie unter [Metriken für Redis OSS](#).

Redis OSS-Benutzer (Clustermodus deaktiviert), die ElastiCache Version 3.2.10 für Redis OSS ausführen, können die Konsole verwenden, um ihre Cluster per Online-Upgrade zu aktualisieren.

Vergleich der ElastiCache Clustergrößenänderung und der Verschlüsselungsunterstützung

Feature	3.2.6	3.2.10	4.0.10 und höher
Online-Größenanpassung eines Clusters *	Nein	Ja	Ja
Verschlüsselung während der Übertragung **	Ja	Nein	Ja
Verschlüsselung im Ruhezustand **	Ja	Nein	Ja
* Hinzufügen, Entfernen und Ausgleichen von Shards.			

** Erforderlich für FedRAMP-, HIPAA- und PCI DSS-konforme Anwendungen. Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

Frühere Versionen für das Ende der Lebensdauer (EOL) (3.x)

ElastiCache Version 3.2.10 für Redis OSS (erweitert)

Amazon ElastiCache stellt die nächste Hauptversion von ElastiCache für die Redis OSS-Engine vor. ElastiCache Version 3.2.10 für Redis OSS (verbessert) führt die Online-Größenänderung von Clustern ein, um Shards zum Cluster hinzuzufügen oder daraus zu entfernen, während der Cluster weiterhin eingehende Anfragen bearbeitet. I/O ElastiCache für Redis OSS 3.2.10 stehen Benutzern alle Funktionen früherer Redis OSS-Versionen zur Verfügung, mit Ausnahme der Möglichkeit, ihre Daten zu verschlüsseln. Diese Möglichkeit ist derzeit nur in Version 3.2.6 verfügbar.

Vergleich der ElastiCache Versionen 3.2.6 und 3.2.10 für Redis OSS

Feature	3.2.6	3.2.10
Online-Größenanpassung eines Clusters *	Nein	Ja
Verschlüsselung während der Übertragung **	Ja	Nein

Feature	3.2.6	3.2.10
Verschlüsselung im Ruhezustand **	Ja	Nein

* Hinzufügen, Entfernen und Ausgleichen von Shards.

** Erforderlich für FedRAMP-, HIPAA- und PCI DSS-konforme Anwendungen. Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

Weitere Informationen finden Sie hier:

- [Online-Resharding für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
- [Online-Größenanpassung von Clustern](#)

ElastiCache Version 3.2.6 für Redis OSS (erweitert)

Amazon ElastiCache stellt die nächste Hauptversion von ElastiCache für die Redis OSS-Engine vor. ElastiCache Version 3.2.6 für Redis OSS-Benutzer haben Zugriff auf alle Funktionen früherer Redis OSS-Versionen sowie auf die Option, ihre Daten zu verschlüsseln. Weitere Informationen finden Sie hier:

- [ElastiCache Verschlüsselung bei der Übertragung \(TLS\)](#)
- [Verschlüsselung im Ruhezustand in ElastiCache](#)
- [Konformitätsvalidierung für Amazon ElastiCache](#)

ElastiCache Version 3.2.4 für Redis OSS (erweitert)

ElastiCache Amazon-Version 3.2.4 führt die nächste Hauptversion von ElastiCache für die Redis OSS-Engine ein. ElastiCache 3.2.4-Benutzern stehen alle Funktionen früherer Redis OSS-Versionen sowie die Option zur Ausführung im Cluster-Modus oder Nicht-Cluster-Modus zur Verfügung. Die folgende Tabelle enthält eine Zusammenfassung.

Vergleich von Redis OSS 3.2.4 im Nicht-Cluster-Modus und im Cluster-Modus

Feature	Nicht-Cluster-Modus	Cluster-Modus
Datenpartitionierung	Nein	Ja

Feature	Nicht-Cluster-Modus	Cluster-Modus
Räumliche Indizierung	Ja	Ja
Knotentyp ändern	Ja	Ja*
Replica-Skalierung	Ja	Ja*
Horizontale Skalierung	Nein	Ja*
Datenbank-Support	Mehrere	Einzel
Parametergruppe	default.redis3.2 **	default.redis3.2.cluster.on **

* Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#)

** Oder davon abgeleitet.

Hinweise:

- Partitionierung – die Möglichkeit, Ihre Daten auf 2 bis 500 Knotengruppen (Shards) mit Replikationsunterstützung für jede Knotengruppe aufzuteilen.
- Geospatiale Indizierung — Redis OSS 3.2.4 bietet Unterstützung für die Geodatenindizierung über sechs GEO-Befehle. Weitere Informationen finden Sie in der Redis OSS GEO*-Befehlsdokumentation [Befehle: GEO auf der Seite Valkey-Befehle \(gefiltert nach GEO\)](#).

Informationen zu zusätzlichen Funktionen von Redis OSS 3 finden Sie in den Versionshinweisen zu [Redis OSS 3.2 und den Versionshinweisen](#) zu [Redis](#) OSS 3.0.

Derzeit ElastiCache verwaltetes Valkey oder Redis OSS (Clustermodus aktiviert) unterstützt die folgenden Redis OSS 3.2-Funktionen nicht:

- Replikatmigration
- Cluster-Ausgleich
- Lua Debugger

ElastiCache deaktiviert die folgenden Redis OSS 3.2-Verwaltungsbefehle:

- `cluster meet`
- `cluster replicate`
- `cluster flushslots`
- `cluster addslots`
- `cluster delslots`
- `cluster setslot`
- `cluster saveconfig`
- `cluster forget`
- `cluster failover`
- `cluster bumpepoch`
- `cluster set-config-epoch`
- `cluster reset`

Informationen zu den Redis OSS 3.2.4-Parametern finden Sie unter [Änderungen der Parameter in Redis OSS 3.2.4](#)

Frühere Versionen für das Ende der Lebensdauer (EOL) (2.x)

ElastiCache Version 2.8.24 für Redis OSS (erweitert)

Zu den seit Version 2.8.23 hinzugefügten Verbesserungen an Redis OSS gehören Bugfixes und die Protokollierung fehlerhafter Speicherzugriffsadressen. Weitere Informationen finden Sie in den Versionshinweisen zu [Redis OSS 2.8](#).

ElastiCache Version 2.8.23 für Redis OSS (erweitert)

Die seit Version 2.8.22 hinzugefügten Redis OSS-Verbesserungen beinhalten Fehlerkorrekturen. Weitere Informationen finden Sie in den Versionshinweisen zu [Redis OSS 2.8](#). Diese Version umfasst außerdem Support für den neuen Parameter `close-on-slave-write` der in aktiviertem Zustand die Verbindung zu Clients trennt, die versuchen, in ein schreibgeschütztes Replikat zu schreiben.

Weitere Informationen zu den Parametern von Redis OSS 2.8.23 finden Sie [Redis OSS 2.8.23 \(erweitert\) hat Parameter hinzugefügt](#) im Benutzerhandbuch. ElastiCache

ElastiCache Version 2.8.22 für Redis OSS (erweitert)

Zu den Verbesserungen von Redis OSS, die seit Version 2.8.21 hinzugefügt wurden, gehören:

- Support für unvergabelte Sicherungen und Synchronisationen, sodass Sie weniger Speicher für die Sicherungsverwaltung und mehr für Ihre Anwendung zuweisen können. Weitere Informationen finden Sie unter [So werden Synchronisation und Backup implementiert](#). Der unvergabelte Prozess kann sich sowohl auf die Latenz als auch den Durchsatz auswirken. Bei einem hohen Schreibdurchsatz kann ein Replikat, das erneut synchronisiert wird, für den gesamten Synchronisierungszeitraum nicht erreichbar sein.
- Bei einem Failover werden Replikationsgruppen nun schneller wiederhergestellt, da Replikate möglichst eine partielle Synchronisierung mit dem primären Knoten anstelle von vollständigen Synchronisierungen durchführen. Darüber hinaus verwenden sowohl der primäre Knoten als auch die Replicas während der Synchronisierungen keinen Datenträger mehr, was sich positiv auf die Geschwindigkeit auswirkt.
- Support für zwei neue CloudWatch Metriken.
 - `ReplicationBytes` – Die Anzahl der Bytes, die der primäre Cluster einer Replikationsgruppe an die Lesereplikate sendet.
 - `SaveInProgress` – Ein Binärwert, der angibt, ob ein Speicherprozess im Hintergrund ausgeführt wird oder nicht.

Weitere Informationen finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#).

- Eine Reihe entscheidender Fehlerbehebungen im PSYNC-Replikationsverhalten. Weitere Informationen finden Sie in den [Versionshinweisen zu Redis OSS 2.8](#).
- Um die verbesserte Replikationsleistung in Multi-AZ-Replikationsgruppen aufrechtzuerhalten und die Cluster-Stabilität zu erhöhen, werden ElastiCache Nicht-Replikate nicht mehr unterstützt.
- Zur Verbesserung der Datenkonsistenz zwischen dem primären Cluster und Replikaten in einer Replikationsgruppe entfernen die Replikate keine Schlüssel unabhängig vom primären Cluster mehr.
- Redis OSS-Konfigurationsvariablen `appendonly` und `appendfsync` werden in Redis OSS Version 2.8.22 und höher nicht unterstützt.
- In Situationen mit geringem Speicher können Clients mit einem großen Ausgabepuffer von einem Replikat-Cluster getrennt werden. In diesem Fall muss der Client die Verbindung erneut herstellen. Solche Situationen treten häufig bei PUBSUB-Clients auf.

ElastiCache Version 2.8.21 für Redis OSS

Die seit Version 2.8.19 hinzugefügten Redis OSS-Verbesserungen beinhalten eine Reihe von Fehlerkorrekturen. Weitere Informationen finden Sie in den Versionshinweisen zu [Redis OSS 2.8](#).

ElastiCache Version 2.8.19 für Redis OSS

Zu den Verbesserungen von Redis OSS, die seit Version 2.8.6 hinzugefügt wurden, gehören:

- Support für HyperLogLog. Weitere Informationen finden Sie unter [Neue Datenstruktur von Redis OSS: HyperLogLog](#).
- Der Datentyp „Sorted Set“ verfügt mit den neuen Befehlen ZRANGEBYLEX, ZLEXCOUNT und ZREMRANGEBYLEX nun über Support für lexikografische Bereichsabfragen.
- Damit ein primärer Knoten keine veralteten Daten an Replica-Knoten sendet, schlägt die Master-Synchronisierung fehl, wenn ein untergeordneter Speicherprozess im Hintergrund (bgsave) abgebrochen wird.
- Support für die HyperLogLogBasedCommands CloudWatchMetrik. Weitere Informationen finden Sie unter [Metriken für Valkey und Redis OSS](#).

ElastiCache Version 2.8.6 für Redis OSS

Zu den Verbesserungen von Redis OSS, die seit Version 2.6.13 hinzugefügt wurden, gehören:

- Verbesserte Resilienz und Fehlertoleranz für Read Replicas.
- Support für partielle Synchronisierung.
- Support für eine benutzerdefinierte Mindestanzahl von Read Replicas, die jederzeit verfügbar sein müssen.
- Volle Unterstützung für Pub/Sub – Benachrichtigen von Clients über Ereignisse auf dem Server.
- Automatische Erkennung eines Ausfalls des primären Knotens und eines Failovers Ihres primären auf einen sekundären Knoten.

ElastiCache Version 2.6.13 für Redis OSS

ElastiCache Version 2.6.13 für Redis OSS war die erste Version, die Redis OSS unterstützte.

ElastiCache Multi-AZ wird in ElastiCache Version 2.6.13 für Redis OSS nicht unterstützt.

ElastiCache Zeitplan für das Ende der Lebensdauer von Versionen für Redis OSS

In diesem Abschnitt werden die Termine für das Ende der Lebensdauer (EOL) für ältere Hauptversionen so festgelegt, wie sie angekündigt werden. Auf diese Weise können Sie Versions- und Upgrade-Entscheidungen für die Zukunft treffen.

 Note

ElastiCache Versionen von 5.0.0 bis 5.0.5 für Redis OSS sind veraltet. Verwenden Sie Version 5.0.6 oder höher.

Die folgende Tabelle zeigt den Zeitplan für den [erweiterten Support](#) ElastiCache für Redis OSS-Engines.

Erweiterter Support und Zeitplan für das Ende des Lebenszyklus

Hauptversion der Engine	Ende des Standard-Supports	Beginn des erweiterten Support Y1 Premium	Beginn des erweiterten Support Y2 Premium	Beginn des erweiterten Support Y3 Premium	Ende des erweiterten Support und Versions-EOL
Redis OSS v4	31.1.2026	1.2.2026	1.2.2027	1.2.2028	31.1.2029
Redis OSS v5	31.1.2026	1.2.2026	1.2.2027	1.2.2028	31.1.2029
Redis OSS v6	31.1.2027	1.2.2027	1.2.2028	1.2.2029	31.1.2030

Die folgende Tabelle fasst jede Version und ihr angekündigtes EOL-Datum sowie die empfohlene Upgrade-Zielversion zusammen.

Vergangenes EOL

Quelle für Unterversionen	Empfohlenes Upgrade-Ziel	EOL-Datum
V 3.2.4, 3.2.6 und 3.2.10 3	Version 6.2 oder höher  Note Für die Regionen US-ISO-EAST US- ISO-WEST -1, -1 und US-ISOB-EAST -1 empfehlen wir 5.0.6 oder höher.	31. Juli 2023
V 2.8.24, 2.8.23, 2.8.22, 2.8.21, 2 2.8.19, 2.8.12, 2.8.6, 2.6.13	Version 6.2 oder höher  Note Für die Regionen -1, -1 und -1 empfehlen wir 5.0.6 oder höher. US-ISO-EAST US- ISO-WEST US-ISOB- EAST	13. Januar 2023

Wird ElastiCache für Memcached-Versionen unterstützt

ElastiCache unterstützt die folgenden Memcached-Versionen und Upgrades auf neuere Versionen. Achten Sie beim Upgrade auf eine neuere Version sorgfältig auf die Bedingungen, die Ihr Upgrade fehlschlagen lassen, wenn sie nicht erfüllt werden.

ElastiCache für Memcached-Versionen

- [ElastiCache Version 1.6.22 für Memcached](#)

- [ElastiCache Version 1.6.17 für Memcached](#)
- [ElastiCache Version 1.6.12 für Memcached](#)
- [ElastiCache Version 1.6.6 für Memcached](#)
- [ElastiCache Version 1.5.16 für Memcached](#)
- [ElastiCache Version 1.5.10 für Memcached](#)
- [ElastiCache Version 1.4.34 für Memcached](#)
- [ElastiCache Version 1.4.33 für Memcached](#)
- [ElastiCache Version 1.4.24 für Memcached](#)
- [ElastiCache Version 1.4.14 für Memcached](#)
- [ElastiCache Version 1.4.5 für Memcached](#)

ElastiCache Version 1.6.22 für Memcached

ElastiCache für Memcached Version 1.6.22 für Memcached fügt Unterstützung für Memcached Version 1.6.22 hinzu. Es sind keine neuen Funktionen enthalten, aber Fehlerbehebungen und kumulative Updates von [Memcached 1.6.18](#).

[Weitere Informationen finden Sie unter 1622 bei Memcached am. ReleaseNotes](#) GitHub

ElastiCache Version 1.6.17 für Memcached

ElastiCache für Memcached Version 1.6.17 für Memcached fügt Unterstützung für Memcached Engine Version 1.6.17 hinzu. Es sind keine neuen Funktionen enthalten, aber Fehlerbehebungen und kumulative Updates von [Memcached 1.6.17](#).

[Weitere Informationen finden Sie unter 1617 bei Memcached am. ReleaseNotes](#) GitHub

ElastiCache Version 1.6.12 für Memcached

ElastiCache Version 1.6.12 für Memcached für Memcached fügt Unterstützung für Memcached Engine 1.6.12 und Verschlüsselung bei der Übertragung hinzu. Es enthält Fehlerkorrekturen und kumulative Aktualisierungen von [Memcached 1.5.16](#).

Weitere Informationen finden Sie [ReleaseNotesunter 1612](#) bei GitHub Memcached am.

ElastiCache Version 1.6.6 für Memcached

ElastiCache für Memcached Version 1.6.6 für Memcached fügt Unterstützung für Memcached Version 1.6.6 hinzu. [Sie enthält keine neuen Funktionen, jedoch Bugfixes und kumulative Updates von Memcached 1.5.16.](#) ElastiCache [für Memcached beinhaltet keine Unterstützung für Extstore.](#)

Weitere Informationen finden Sie in Memcached unter [ReleaseNotes166](#). GitHub

ElastiCache Version 1.5.16 für Memcached

ElastiCache Version 1.5.16 für Memcached fügt Unterstützung für Memcached Version 1.5.16 hinzu. Es sind keine neuen Funktionen enthalten, dafür aber Fehlerbehebungen und kumulative Updates von [Memcached 1.5.14](#) und [Memcached 1.5.15](#).

[Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.5.16 auf Memcached am.](#) GitHub

ElastiCache Version 1.5.10 für Memcached

ElastiCache Version 1.5.10 für Memcached unterstützt die folgenden Memcached-Funktionen:

- Automatisierter Slab-Ausgleich
- Schnelleres Suchen von Hash-Tabellen durch murmur3-Algorithmus
- Segmentierter LRU-Algorithmus
- LRU-Crawler, um im Hintergrund Speicher zurückzugewinnen
- `--enable-seccomp`: eine Kompilierungszeitoption

Es werden auch die Parameter `no_modern` und `inline_ascii_resp` eingeführt. Weitere Informationen finden Sie unter [Parameteränderungen für Memcached 1.5.10](#).

Zu den Verbesserungen von Memcached, die seit ElastiCache Version 1.4.34 für Memcached hinzugefügt wurden, gehören:

- Kumulative Korrekturen wie ASCII-Multigets, CVE-2017-9951 und Limit-Crawler for metadumper
- Verbesserte Verbindungsverwaltung durch das Schließen von Verbindungen beim Erreichen der maximalen Verbindungsanzahl
- Verbesserte Verwaltung der Elementgröße bei Elementen, deren Größe 1 MB überschreitet
- Bessere Leistung und Optimierung des Speichermehraufwands durch Reduzierung der Speicheranforderungen pro Element um wenige Bytes

Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.5.10 auf [Memcached](#) am. GitHub

ElastiCache Version 1.4.34 für Memcached

ElastiCache Version 1.4.34 für Memcached fügt Version 1.4.33 keine neuen Funktionen hinzu. Version 1.4.34 dient der Fehlerbehebung und ist größer als für eine solche Version üblich.

Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.4.34 auf [Memcached](#) am. GitHub

ElastiCache Version 1.4.33 für Memcached

Zu den Verbesserungen, die seit Version 1.4.24 hinzugefügt wurden, gehören:

- Die Fähigkeit, alle Metadaten für eine bestimmte Slab-Klasse, eine Liste der Slab-Klassen oder alle Slab-Klassen zu sichern. Weitere Informationen finden Sie unter [Memcached 1.4.31 Release Notes](#).
- Verbesserter Support für große Elemente über den Standardwert von 1 MB. Weitere Informationen finden Sie unter [Memcached 1.4.29 Release Notes](#).
- Die Fähigkeit, anzugeben, wie lange sich ein Client im Leerlauf befinden kann, bevor er zum Schließen aufgefordert wird.

Die Fähigkeit, den für Memcached verfügbaren Speicherplatz dynamisch zu erhöhen, ohne den Cluster neu starten zu müssen. Weitere Informationen finden Sie unter [Memcached 1.4.27 Release Notes](#).

- Die Protokollierung von `fetchers`, `mutations` und `evictions` wird jetzt unterstützt. Weitere Informationen finden Sie unter [Memcached 1.4.26 Release Notes](#).
- Freier Speicher kann für einen globalen Pool wieder freigegeben und neuen Slab-Klassen zugewiesen werden. Weitere Informationen finden Sie unter [Memcached 1.4.25 Release Notes](#).
- Mehrere Fehlerbehebungen
- Einige neue Befehle und Parameter Eine Liste finden Sie hier: [In Memcached 1.4.33 hinzugefügte Parameter](#).

ElastiCache Version 1.4.24 für Memcached

Zu den Verbesserungen, die seit Version 1.4.14 hinzugefügt wurden, gehören:

- Die LRU-Verwaltung (Least Recently Used) mit einem Hintergrundprozess wurde verbessert.
- Die Option, jenkins oder murmur3 als Hash-Algorithmus zu verwenden, wurde hinzugefügt.
- Einige neue Befehle und Parameter Eine Liste finden Sie hier: [In Memcached 1.4.24 hinzugefügte Parameter](#).
- Mehrere Fehlerbehebungen

ElastiCache Version 1.4.14 für Memcached

Zu den Verbesserungen, die seit Version 1.4.5 hinzugefügt wurden, gehören:

- Verbesselter Slab-Ausgleich.
- Performance und Skalierbarkeit wurden verbessert.
- Der Befehl touch wurde eingeführt, um das Ablaufdatum eines vorhandenen Elements zu aktualisieren, ohne es abzurufen.
- Automatische Erkennung – die Möglichkeit für Client-Programme, automatisch alle Cache-Knoten in einem Cluster zu ermitteln und Verbindungen zu all diesen Knoten zu initiieren und aufrechtzuerhalten.

ElastiCache Version 1.4.5 für Memcached

ElastiCache Version 1.4.5 für Memcached war die erste Engine und Version, die von Amazon ElastiCache für Memcached unterstützt wurden.

Wichtige Unterschiede zwischen dem Verhalten und der Kompatibilität der Engine-Versionen mit Valkey

Valkey 7.2.6 weist ähnliche Kompatibilitätsunterschiede zu früheren Versionen von Redis OSS 7.2.4 auf. Die neueste unterstützte Version von Valkey finden Sie unter [Unterstützte Engines und Versionen](#)

Weitere Informationen zur Version Valkey 7.2 finden Sie in den [Versionshinweisen zu Redis OSS 7.2.4](#) (Valkey 7.2 enthält alle Änderungen von Redis OSS bis Version 7.2.4) und in den Versionshinweisen zu Valkey 7.2 unter [Valkey](#) am. GitHub

Hier sind die potenziell schwerwiegenden Verhaltensänderungen zwischen Valkey 7.2 und Redis OSS 7.1 (oder 7.0):

- Das Freeze Time-Sampling erfolgt während der Befehlsausführung und in Skripten.
- Ein blockierter Stream-Befehl, der freigegeben wird, wenn der Schlüssel nicht mehr existiert, enthält einen anderen Fehlercode (-NOGROUP oder -WRONGTYPE statt -UNBLOCKED).
- Die clientseitige Nachverfolgung von Skripten verfolgt nun die Schlüssel, die vom Skript gelesen werden, und nicht die Schlüssel, die vom Aufrufer von EVAL/FCALL deklariert wurden.

Wesentliche Unterschiede zwischen dem Verhalten und der Kompatibilität der Engine-Versionen mit Redis OSS

Important

Auf der folgenden Seite sind alle Inkompatibilitätsunterschiede zwischen den Versionen aufgeführt und alle Aspekte angegeben, die Sie beim Upgrade auf neuere Versionen beachten sollten. In dieser Liste sind alle Probleme bezüglich Versionsinkompatibilitäten aufgeführt, die beim Upgrade auftreten können.

Sie können direkt von Ihrer aktuellen Redis OSS-Version auf die neueste verfügbare Redis OSS-Version aktualisieren, ohne dass sequentielle Upgrades erforderlich sind. Sie können beispielsweise direkt von Redis OSS Version 3.0 auf Version 7.0 aktualisieren.

Redis OSS-Versionen werden mit einer semantischen Version identifiziert, die aus einer Haupt-, Neben- und einer Patch-Komponente besteht. In Redis OSS 4.0.10 ist die Hauptversion

beispielsweise 4, die Nebenversion 0 und die Patch-Version 10. Diese Werte werden im Allgemeinen basierend auf den folgenden Konventionen schrittweise erhöht:

- Hauptversionen sind für API-inkompatible Änderungen vorgesehen
- Nebenversionen sind für neue Funktionen vorgesehen, die abwärtskompatibel hinzugefügt wurden
- Patch-Versionen sind für abwärtskompatible Bugfixes und nicht funktionale Änderungen vorgesehen

Wir empfehlen, immer die neueste Patch-Version innerhalb einer bestimmten Haupt-/Nebenversion zu verwenden, um die neuesten Leistungs- und Stabilitätsverbesserungen zu erzielen. Ab ElastiCache Version 6.0 für Redis OSS ElastiCache wird eine einzige Version für jede Redis OSS-Nebenversion angeboten, anstatt mehrere Patch-Versionen anzubieten. ElastiCache verwaltet automatisch die Patch-Version Ihrer laufenden Cluster und sorgt so für eine verbesserte Leistung und erhöhte Sicherheit.

Wir empfehlen außerdem, regelmäßig auf die neueste Hauptversion zu aktualisieren, da die meisten wichtigen Verbesserungen nicht auf ältere Versionen zurückportiert werden. Da die Verfügbarkeit auf eine neue AWS Region ElastiCache ausgedehnt wird, unterstützt OSS ElastiCache für Redis die beiden jeweils neuesten Major.Minor-Versionen für die neue Region. Wenn beispielsweise eine neue AWS Region eingeführt wird und die neuesten ElastiCache Major.Minor-Versionen für Redis OSS 7.0 und 6.2 sind, ElastiCache werden die Redis OSS-Versionen 7.0 und 6.2 in der neuen Region unterstützt. AWS. Sobald neuere Major.Minor-Versionen von ElastiCache für Redis OSS veröffentlicht werden, ElastiCache wird weiterhin Unterstützung für die neu veröffentlichten Versionen hinzugefügt. Weitere Informationen zur Auswahl von Regionen für finden Sie unter [Regionen ElastiCache und Verfügbarkeitszonen auswählen](#).

Wenn Sie ein Upgrade durchführen, das Haupt- oder Nebenversionen umfasst, beachten Sie bitte die folgende Liste, die Verhaltens- und rückwärtsinkompatible Änderungen enthält, die im Laufe der Zeit mit Redis OSS veröffentlicht wurden.

Verhalten und abwärtsinkompatible Änderungen von Redis OSS 7.0

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 7.0](#).

- `SCRIPT LOAD` und `SCRIPT FLUSH` werden nicht mehr auf Replikate weitergeleitet. Wenn Sie eine gewisse Haltbarkeit für Skripts benötigen, empfehlen wir Ihnen, die Verwendung von [Redis OSS-Funktionen](#) in Betracht zu ziehen.
- Pubsub-Kanäle sind jetzt für neue ACL-Benutzer standardmäßig gesperrt.

- Der STRALGO-Befehl wurde durch den LCS-Befehl ersetzt.
- Das Format für ACL GETUSER wurde geändert, sodass alle Felder das standardmäßige Zugriffszeichenfolgemuster anzeigen. Wenn Sie Automatisierung mit ACL GETUSER verwendet haben, sollten Sie überprüfen, ob beide Formate verarbeitet werden.
- Die ACL-Kategorien für SELECT, WAIT, ROLE, LASTSAVE, READONLY, READWRITE und ASKING haben sich geändert.
- Der INFO-Befehl zeigt jetzt Befehlsstatistiken pro Unterbefehl und anstatt bei den Container-Befehlen in der obersten Ebene an.
- Die Rückgabewerte der Befehle LPOP, RPOP, ZPOPMIN und ZPOPMAX haben sich in bestimmten Randfällen geändert. Wenn Sie diese Befehle verwenden, sollten Sie die Versionshinweise überprüfen und bewerten, ob Sie betroffen sind.
- Die Befehle SORT und SORT_RO erfordern jetzt Zugriff auf den gesamten Schlüsselraum, um die Argumente GET sowie BY verwenden zu können.

Verhalten und abwärtsinkompatible Änderungen in Redis OSS 6.2

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 6.2](#).

- Die ACL-Flags der Befehle TIME, ECHO, ROLE und LASTSAVE wurden geändert. Dies kann dazu führen, dass Befehle, die zuvor gestattet waren, abgelehnt werden und umgekehrt.

Note

Keiner dieser Befehle verändert Daten oder gewährt Zugriff darauf.

- Beim Upgrade von Redis OSS 6.0 wird die Reihenfolge der key/value Paare geändert, die von einer Map-Antwort an ein Lua-Skript zurückgegeben werden. Wenn Ihre Skripte eine Map verwenden `redis.setresp()` oder zurückgeben (neu in Redis OSS 6.0), sollten Sie die Auswirkungen berücksichtigen, die sich daraus ergeben können, dass das Skript bei Upgrades nicht mehr funktioniert.

Verhalten von Redis OSS 6.0 und abwärtsinkompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 6.0](#).

- Die maximale Anzahl zugelassener Datenbanken wurde von 1,2 Millionen auf 10.000 verringert. Der Standardwert ist 16, und wir raten davon ab, Werte zu verwenden, die viel größer sind, da wir Probleme mit der Leistung und dem Arbeitsspeicher festgestellt haben.
- Stellen Sie den `AutoMinorVersionUpgrade` Parameter auf `yes` ein und das Upgrade der Nebenversion ElastiCache wird über Self-Service-Updates verwaltet. Dies wird über Standardkanäle für die Kundenbenachrichtigung über eine Self-Service-Update-Kampagne abgewickelt. Weitere Informationen finden Sie unter [Self-Service-Updates](#) unter ElastiCache

Verhalten und abwärtsinkompatible Änderungen in Redis OSS 5.0

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 5.0](#).

- Skripte werden durch Effekte repliziert, statt das Skript auf dem Replikat erneut auszuführen. Dies verbessert im Allgemeinen die Leistung, kann jedoch die Menge der zwischen primären Replikaten und Replikaten replizierten Daten erhöhen. Es gibt eine Option, um zum vorherigen Verhalten zurückzukehren, die nur in ElastiCache Version 5.0 für Redis OSS verfügbar ist.
- Wenn Sie ein Upgrade von Redis OSS 4.0 durchführen, geben einige Befehle in LUA-Skripten Argumente in einer anderen Reihenfolge zurück als in früheren Versionen. In Redis OSS 4.0 ordnete Redis OSS einige Antworten lexographisch an, um die Antworten deterministisch zu machen. Diese Reihenfolge wird nicht angewendet, wenn Skripten durch Effekte repliziert werden.
- In Redis OSS 5.0.3 und höher wird bei Redis OSS bei ElastiCache Instance-Typen mit mehr als 4 ein Teil der I/O-Arbeit auf Hintergrundkerne ausgelagert. VCPUs Dies kann die Leistungsmerkmale von Redis OSS und die Werte einiger Metriken ändern. Weitere Informationen finden Sie unter [Welche Metriken sollte ich überwachen?](#), um zu verstehen, ob Sie ändern müssen, welche Metriken Sie sich ansehen.

Verhalten von Redis OSS 4.0 und abwärtsinkompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 4.0](#).

- Slow Log protokolliert jetzt zwei zusätzliche Argumente, den Namen und die Adresse des Clients. Diese Änderung sollte abwärtskompatibel sein, sofern Sie sich nicht explizit darauf verlassen, dass jeder Slow-Log-Eintrag 3 Werte enthält.
- Der Befehl `CLUSTER NODES` gibt jetzt ein etwas anderes Format zurück, das nicht abwärtskompatibel ist. Wir empfehlen, dass Clients diesen Befehl nicht verwenden, um mehr über

die in einem Cluster vorhandenen Knoten zu erfahren, sondern stattdessen CLUSTER SLOTS verwenden.

Vergangenes EOL

Verhalten von Redis OSS 3.2 und abwärtsinkompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 3.2](#).

- Für diese Version sind keine Kompatibilitätsänderungen erforderlich.

Weitere Informationen finden Sie unter [ElastiCache Zeitplan für das Ende der Lebensdauer von Versionen für Redis OSS](#).

Verhalten von Redis OSS 2.8 und abwärtsinkompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 2.8](#).

- Ab Redis OSS 2.8.22 wird Redis OSS AOF für Redis OSS nicht mehr unterstützt. ElastiCache Wir empfehlen die Verwendung von MemoryDB, wenn Daten dauerhaft gespeichert werden müssen.
- Ab Redis OSS 2.8.22 unterstützt Redis OSS das Anhängen von Replikaten an ElastiCache darin gehostete Primärdateien nicht mehr. ElastiCache Während des Upgrades werden externe Replikate getrennt und sie können keine erneute Verbindung herstellen. Wir empfehlen die Verwendung von clientseitigem Caching, das in Redis OSS 6.0 als Alternative zu externen Replikaten verfügbar ist.
- Die Befehle TTL und PTTL geben jetzt -2 zurück, wenn der Schlüssel nicht existiert, und -1, wenn er existiert, aber kein zugehöriges Ablaufdatum hat. Redis OSS 2.6 und frühere Versionen gaben in beiden Fällen -1 zurück.
- SORT mit ALPHA sortiert jetzt nach lokalem Standardgebietsschema, wenn keine STORE-Option verwendet wird.

Weitere Informationen finden Sie unter [ElastiCache Zeitplan für das Ende der Lebensdauer von Versionen für Redis OSS](#).

Überlegungen zum Upgrade bei der Arbeit mit knotenbasierten Clustern

Note

Die folgenden Überlegungen gelten nur für das Upgrade knotenbasierter Cluster. Sie gelten nicht für Serverless. ElastiCache

Überlegungen zu Valkey und Redis OSS

Beachten Sie beim Upgrade eines knotenbasierten Valkey- oder Redis OSS-Clusters Folgendes.

- Versionsverwaltung der Engine ist so entwickelt, dass Sie so viel Kontrolle wie möglich darüber haben, wie Patchen erfolgt. ElastiCache behält sich jedoch das Recht vor, Ihren Cluster in Ihrem Namen zu patchen, falls im unwahrscheinlichen Fall eine kritische Sicherheitslücke im System oder in der Cache-Software auftritt.
- Ab ElastiCache Version 7.2 für Valkey und ElastiCache Version 6.0 für Redis OSS ElastiCache wird für jede Nebenversion eine einzige Version angeboten, anstatt mehrere Patch-Versionen anzubieten.
- Ab Redis OSS Engine Version 5.0.6 können Sie Ihre Cluster-Version mit minimaler Ausfallzeit aktualisieren. Der Cluster kann während des gesamten Upgrades gelesen und in der Regel auch beschrieben werden, ausgenommen während der Failover-Operation, der nur einige Sekunden dauert.
- Sie können Ihre ElastiCache Cluster auch mit Versionen vor 5.0.6 aktualisieren. Der Prozess ist identisch, kann jedoch längere Failover-Zeit während der DNS-Ausbreitung (30 Sek. - 1 Min.) verursachen.
- Ab Redis OSS 7 wird ElastiCache das Umschalten zwischen Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert) unterstützt.
- Der Upgrade-Prozess ElastiCache für die Amazon for Redis OSS-Engine ist darauf ausgelegt, Ihre vorhandenen Daten bestmöglich beizubehalten, und erfordert eine erfolgreiche Redis OSS-Replikation.
- Beim Upgrade der Engine ElastiCache werden bestehende Client-Verbindungen beendet. [Um Ausfallzeiten bei Engine-Upgrades zu minimieren, empfehlen wir Ihnen, bewährte Methoden für Redis OSS-Clients mit Fehlerwiederholungen und exponentiellem Backoff sowie bewährte Methoden zur Minimierung von Ausfallzeiten während der Wartung zu implementieren.](#)

- Sie können beim Upgrade Ihrer Engine kein direktes Upgrade von Valkey oder Redis OSS (Clustermodus deaktiviert) auf Valkey oder Redis OSS (Clustermodus aktiviert) durchführen. Das folgende Verfahren zeigt Ihnen, wie Sie von Valkey oder Redis OSS (Clustermodus deaktiviert) auf Valkey oder Redis OSS (Clustermodus aktiviert) aktualisieren.

Um ein Upgrade von einer Valkey- oder Redis OSS-Engine-Version (Cluster-Modus deaktiviert) auf eine Valkey- oder Redis OSS-Engine-Version (Clustermodus aktiviert) durchzuführen

1. Erstellen Sie eine Sicherungskopie Ihres Valkey- oder Redis OSS-Clusters oder Ihrer Replikationsgruppe (Clustermodus deaktiviert). Weitere Informationen finden Sie unter [Erstellen manueller Backups](#).
 2. Verwenden Sie das Backup, um einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mit einem Shard (Knotengruppe) zu erstellen und zu starten. Geben Sie die neue Engine-Version an und aktivieren Sie den Cluster-Modus, wenn Sie den Cluster oder die Replikationsgruppe erstellen. Weitere Informationen finden Sie unter [Tutorial: Seeding eines neuen knotenbasierten Clusters mit einem extern erstellten Backup](#).
 3. Löschen Sie den alten Valkey- oder Redis OSS-Cluster oder die alte Replikationsgruppe (Clustermodus deaktiviert). Für weitere Informationen siehe [Löschen eines Clusters in ElastiCache](#) oder [Löschen einer Replikationsgruppe](#).
 4. Skalieren Sie den neuen Valkey- oder Redis OSS-Cluster oder die neue Replikationsgruppe (Clustermodus aktiviert) auf die Anzahl der Shards (Knotengruppen), die Sie benötigen. Weitere Informationen finden Sie unter [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#).
- Beim Upgrade von Hauptversionen der Engine, beispielsweise von 5.0.6 auf 6.0, müssen Sie auch eine neue Parametergruppe auswählen, die mit der neuen Engine-Version kompatibel ist.
 - Für einzelne Redis OSS-Cluster und Cluster mit deaktiviertem Multi-AZ empfehlen wir, Redis OSS ausreichend Speicher zur Verfügung zu stellen, wie unter beschrieben. [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#)
In diesen Fällen steht der primäre Knoten während des Upgrade-Prozesses für Serviceanfragen nicht zur Verfügung.
 - Für Redis OSS-Cluster mit aktiviertem Multi-AZ empfehlen wir außerdem, Engine-Upgrades in Zeiten mit geringem eingehendem Schreibverkehr zu planen. Bei einem Upgrade auf Redis OSS 5.0.6 oder höher steht der primäre Cluster während des Upgrade-Vorgangs weiterhin für Serviceanfragen zur Verfügung.

Cluster und Replikationsgruppen mit mehreren Shards werden wie folgt verarbeitet und gepatcht:

- Alle Shards werden parallel verarbeitet. Es wird jeweils nur eine Upgrade-Operation für einen Shard gleichzeitig durchgeführt.
- In jedem Shard werden alle Replicas verarbeitet, bevor der Primärknoten verarbeitet wird. Wenn es in einem Shard weniger Replicas gibt, kann der Primärknoten in diesem Shard verarbeitet werden, bevor die Verarbeitung der Replicas in anderen Shards abgeschlossen wird.
- Die Primärknoten für alle Shards werden seriell verarbeitet. Es erfolgt jeweils nur ein Upgrade für einen Primärknoten gleichzeitig.
- Wenn die Verschlüsselung in Ihrem aktuellen Cluster oder Ihrer Replikationsgruppe aktiviert ist, können Sie nicht auf eine Engine-Version aktualisieren, die keine Verschlüsselung unterstützt, z. B. von 3.2.6 auf 3.2.10.

Überlegungen zu Memcached

Beachten Sie beim Upgrade eines knotenbasierten Memcached-Clusters Folgendes.

- Versionsverwaltung der Engine ist so entwickelt, dass Sie so viel Kontrolle wie möglich darüber haben, wie Patchen erfolgt. Behält sich jedoch ElastiCache das Recht vor, Ihren Cluster in Ihrem Namen zu patchen, sollte der unwahrscheinliche Fall eintreten, dass das System oder die Cache-Software eine kritische Sicherheitslücke aufweist.
- Da die Memcached-Engine keine Persistenz unterstützt, stellen Versions-Upgrades der Memcached-Engine immer einen Störfall dar, bei dem alle Cache-Daten im Cluster gelöscht werden.

ElastiCache Best Practices und Caching-Strategien

Im Folgenden finden Sie empfohlene Best Practices für Amazon ElastiCache. Durch die Einhaltung dieser Methoden lassen sich die Performance und Zuverlässigkeit des Caches verbessern.

Themen

- [Allgemeine Best Practices](#)
- [Bewährte Methoden für die Verwendung von Read Replicas](#)
- [Unterstützte und eingeschränkte Valkey-, Memcached- und Redis OSS-Befehle](#)
- [Konfiguration und Grenzwerte für Valkey und Redis OSS](#)
- [IPv6 Kundenbeispiele für Valkey, Memcached und Redis OSS](#)

- [Bewährte Methoden für Kunden \(Valkey und Redis OSS\)](#)
- [Bewährte Methoden für Kunden \(Memcached\)](#)
- [TLS-fähige ElastiCache Dual-Stack-Cluster](#)
- [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#)
- [Bewährte Methoden bei der Arbeit mit nodenbasierten Valkey- und Redis OSS-Clustern](#)
- [Caching-Strategien für Memcached](#)

Allgemeine Best Practices

Im Folgenden finden Sie Informationen zu bewährten Methoden für die Verwendung der OSS-Schnittstellen Valkey, Memcached und Redis. ElastiCache

- Verwenden Sie Konfigurationen mit aktiviertem Clustermodus — Bei aktiviertem Clustermodus kann der Cache horizontal skaliert werden, um mehr Speicherplatz und Durchsatz zu erzielen als bei einer Konfiguration mit deaktiviertem Clustermodus. ElastiCache Serverless ist nur in einer Konfiguration mit aktiviertem Clustermodus verfügbar.
- Langlebige Verbindungen verwenden – Das Erstellen einer neuen Verbindung ist teuer und beansprucht Zeit und CPU-Ressourcen aus dem Cache. Verwenden Sie Verbindungen nach Möglichkeit wieder (z. B. mit Verbindungspooling), um diese Kosten für viele Befehle zu amortisieren.
- Aus Replikaten lesen — Wenn Sie ElastiCache serverlose Systeme verwenden oder Read Replicas (knotenbasierte Cluster) bereitgestellt haben, leiten Sie Lesevorgänge direkt an Replikate weiter, um eine bessere Skalierbarkeit und geringere Latenz zu erreichen. and/or Lesevorgänge aus Replikaten sind mit dem Primärknoten letztendlich konsistent.

Vermeiden Sie es in einem knotenbasierten Cluster, Leseanforderungen an eine einzelne Read Replica weiterzuleiten, da Lesevorgänge möglicherweise vorübergehend nicht verfügbar sind, wenn der Knoten ausfällt. Konfigurieren Sie Ihren Client entweder so, dass Leseanfragen an mindestens zwei Read Replicas weitergeleitet werden oder Lesevorgänge an ein einzelnes Replikat und den Primärknoten weitergeleitet werden.

Im ElastiCache serverlosen Modus werden Lesevorgänge, wenn vom Replikat-Port (6380) gelesen wird, nach Möglichkeit in die lokale Availability Zone des Clients geleitet, wodurch die Latenz beim Abrufen reduziert wird. Bei Ausfällen wird automatisch auf die anderen Knoten zurückgegriffen.

- Vermeiden Sie teure Befehle — Vermeiden Sie es, I/O rechenintensive Operationen wie die Befehle `KEYS` und `SMEMBERS` auszuführen. Wir empfehlen diesen Ansatz, da diese Operationen

die Last auf dem Cluster erhöhen und Einfluss auf die Performance des Clusters haben.

Verwenden Sie stattdessen die Befehle SCAN und SSCAN.

- Befolgen Sie die bewährten Methoden von Lua – Vermeiden Sie lange laufende Lua-Skripte und deklarieren Sie Schlüssel, die in Lua-Skripten verwendet werden, immer im Voraus. Wir empfehlen diesen Ansatz, um festzustellen, dass im Lua-Skript keine slotübergreifenden Befehle verwendet werden. Vergewissern Sie sich, dass die in Lua-Skripten verwendeten Schlüssel zum gleichen Slot gehören.
- Sharded Pub/Sub verwenden — Wenn Sie Valkey oder Redis OSS verwenden, um pub/sub Workloads mit hohem Durchsatz zu unterstützen, empfehlen wir die Verwendung von [Sharded Pub/Sub](#) (verfügbar mit Valkey und mit Redis OSS 7 oder höher). Herkömmliche Cluster, die pub/sub im Clustermodus aktiviert sind, senden Nachrichten an alle Knoten im Cluster, was zu hohen Werten führen kann. `EngineCPUUtilization` Beachten Sie, dass es sich bei ElastiCache serverlosen Befehlen um herkömmliche Befehle handelt. pub/sub commands internally use sharded pub/sub

Bewährte Methoden für die Verwendung von Read Replicas

Viele Anwendungen, wie Sitzungsspeicher, Bestenlisten und Empfehlungsmodule, erfordern eine hohe Verfügbarkeit und verarbeiten deutlich mehr Lese- als Schreibvorgänge. Diese Anwendungen können häufig leicht veraltete Daten tolerieren (letztendliche Konsistenz), was bedeutet, dass es akzeptabel ist, wenn verschiedene Benutzer vorübergehend leicht unterschiedliche Versionen derselben Daten sehen. Beispiel:

- Im Cache gespeicherte Abfrageergebnisse können häufig leicht veraltete Daten tolerieren, insbesondere bei zwischengespeicherten Mustern, bei denen die Quelle der Wahrheit extern ist.
- In einer Gaming-Bestenliste wirkt sich eine Verzögerung von einigen Sekunden bei aktualisierten Ergebnissen häufig nicht wesentlich auf die Benutzererfahrung aus.
- Bei Sitzungsspeichern wirken sich geringfügige Verzögerungen bei der Übertragung von Sitzungsdaten zwischen Replikaten selten auf die Anwendungsfunktionalität aus.
- Empfehlungs-Engines verwenden in der Regel historische Datenanalysen, sodass Konsistenz in Echtzeit weniger wichtig ist.

Letztendliche Konsistenz bedeutet, dass alle Replikatknoten irgendwann dieselben Daten zurückgeben, sobald der Replikationsprozess abgeschlossen ist, normalerweise innerhalb von

Millisekunden. Für solche Anwendungsfälle ist die Implementierung von Read Replicas eine effektive Strategie, um die Latenz beim Lesen aus Ihrer Instance zu reduzieren. ElastiCache

Die Verwendung von Read Replicas in Amazon ElastiCache kann erhebliche Leistungsvorteile bieten durch:

Verbesserte Leseskalierbarkeit

- Verteilt Lesevorgänge auf mehrere Replikatknoten
- Lädt den Lesetraffic vom primären Knoten ab
- Reduziert die Leselatenz, indem Anfragen von geografisch näher gelegenen Replikaten bearbeitet werden

Optimierte Leistung des Primärknotens

- Weist Ressourcen des primären Knotens für Schreiboperationen zu
- Reduziert den Verbindungsaufwand auf dem Primärknoten
- Verbessert die Schreibleistung und sorgt für bessere Antwortzeiten in Zeiten mit hohem Datenverkehr

Verwenden von Read from Replica in Serverless ElastiCache

ElastiCache Serverless bietet zwei verschiedene Endpunkte für unterschiedliche Konsistenzanforderungen. Die beiden Endpunkte verwenden denselben DNS-Namen, aber unterschiedliche Ports. Um den read-from-replica Port verwenden zu können, müssen Sie den Zugriff auf beide Ports von Ihrer Client-Anwendung aus autorisieren, indem Sie [die Sicherheitsgruppen und Netzwerkzugriffskontrolllisten Ihrer VPC konfigurieren](#).

Primärer Endpunkt (Port 6379)

- Wird für Operationen verwendet, die sofortige Konsistenz erfordern
- Garantiert das Lesen der meisten up-to-date Daten
- Am besten für kritische Transaktionen und Schreibvorgänge
- Erforderlich für Schreiboperationen
- Beispiel: `test-12345.serverless.use1.cache.amazonaws.com:6379`

Latenzoptimierter Endpunkt (Port 6380)

- Optimiert für Lesevorgänge, die eine eventuelle Konsistenz tolerieren
- Wenn möglich, leitet ElastiCache Serverless Leseanfragen automatisch an den Replikatknoten in der lokalen Availability Zone des Clients weiter. Diese Optimierung sorgt für eine geringere Latenz, da die zusätzliche Netzwerklatenz vermieden wird, die beim Abrufen von Daten von einem Knoten in einer anderen Availability Zone entsteht.
- ElastiCache Serverless wählt automatisch verfügbare Knoten in anderen Zonen aus, wenn ein lokaler Knoten nicht verfügbar ist
- Beispiel: `test-12345.serverless.use1.cache.amazonaws.com:6380`
- Clients wie Glide und Lettuce erkennen Lesevorgänge automatisch und leiten sie an den latenzoptimierten Endpunkt weiter, wenn Sie die Konfiguration „Read from Replica“ angeben. Wenn Ihr Client die Routing-Konfiguration nicht unterstützt (z. B. Valkey-Java und ältere Jedis-Versionen), müssen Sie den richtigen Port und die richtige Client-Konfiguration definieren, um aus Replikaten zu lesen.

Verbindung herstellen, um Repliken in Serverless zu lesen — Valkey und Glide ElastiCache

Der folgende Codeausschnitt zeigt, wie Sie Read from Replica für ElastiCache Serverless in der Valkey-Glide-Bibliothek konfigurieren können. Sie müssen keinen Port für das Lesen von Replikaten angeben, aber Sie müssen die Routing-Konfiguration konfigurieren. `ReadFrom.PREFER_REPLICA`

```
package glide.examples;

import glide.api.GlideClusterClient;
import glide.api.logging.Logger;
import glide.api.models.configuration.GlideClusterClientConfiguration;
import glide.api.models.configuration.NodeAddress;
import glide.api.models.exceptions.ClosingException;
import glide.api.models.exceptions.ConnectionException;
import glide.api.models.exceptions.TimeoutException;
import glide.api.models.configuration.ReadFrom;

import java.util.concurrent.CompletableFuture;
import java.util.concurrent.ExecutionException;

public class ClusterExample {

    public static void main(String[] args) {
        // Set logger configuration
```

```
Logger.setLoggerConfig(Logger.Level.INFO);

GlideClusterClient client = null;

try {
    System.out.println("Connecting to Valkey Glide...");

    // Configure the Glide Client
    GlideClusterClientConfiguration config =
GlideClusterClientConfiguration.builder()
        .address(NodeAddress.builder()
            .host("your-endpoint")
            .port(6379)
            .build())
        .useTLS(true)
        .readFrom(ReadFrom.PREFER_REPLICA)
        .build();

    // Create the GlideClusterClient
    client = GlideClusterClient.createClient(config).get();
    System.out.println("Connected successfully.");

    // Perform SET operation
    CompletableFuture<String> setResponse = client.set("key", "value");
    System.out.println("Set key 'key' to 'value': " + setResponse.get());

    // Perform GET operation
    CompletableFuture<String> getResponse = client.get("key");
    System.out.println("Get response for 'key': " + getResponse.get());

    // Perform PING operation
    CompletableFuture<String> pingResponse = client.ping();
    System.out.println("PING response: " + pingResponse.get());

} catch (ClosingException | ConnectionException | TimeoutException |
ExecutionException e) {
    System.err.println("An exception occurred: ");
    e.printStackTrace();
} catch (InterruptedException e) {
    Thread.currentThread().interrupt();
} finally {
    // Close the client connection
    if (client != null) {
        try {
```



```
        client.close();
        System.out.println("Client connection closed.");
    } catch (ClosingException | ExecutionException e) {
        System.err.println("Error closing client: " + e.getMessage());
    }
}
}
```

Unterstützte und eingeschränkte Valkey-, Memcached- und Redis OSS-Befehle

Unterstützte Valkey- und Redis OSS-Befehle

Unterstützte Valkey- und Redis OSS-Befehle

Die folgenden Valkey- und Redis OSS-Befehle werden von serverlosen Caches unterstützt.

Zusätzlich zu diesen Befehlen werden außerdem diese [Unterstützte Valkey- und Redis OSS-Befehle](#) unterstützt.

Informationen zu Bloom-Filter-Befehlen finden Sie unter [Bloom-Filter-Befehle](#)

Bitmap-Befehle

- BITCOUNT

Zählt die Anzahl der gesetzten Bits (Populationszählung) in einer Zeichenfolge.

[Weitere Informationen](#)

- BITFIELD

Führt willkürliche Bitfeld-Ganzzahl-Operationen für Zeichenfolgen aus.

[Weitere Informationen](#)

- BITFIELD_RO

Führt willkürliche, schreibgeschützte Bitfeld-Ganzzahl-Operationen für Zeichenfolgen aus.

[Weitere Informationen](#)

- BITOP

Führt bitweise Operationen an mehreren Zeichenfolgen aus und speichert das Ergebnis.

[Weitere Informationen](#)

- BITPOS

Findet das erste gesetzte (1) oder nicht gesetzte Bit (0) in einer Zeichenfolge.

[Weitere Informationen](#)

- GETBIT

Gibt einen Bitwert nach Offset zurück.

[Weitere Informationen](#)

- SETBIT

Setzt oder löscht das Bit am Offset des Zeichenfolgenwerts. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

Befehle zur Clusterverwaltung

- CLUSTER COUNTKEYSINSLOT

Gibt die Anzahl der Schlüssel in einem Hash-Slot zurück.

[Weitere Informationen](#)

- CLUSTER GETKEYSINSLOT

Gibt die Schlüsselnamen in einem Hash-Slot zurück.

[Weitere Informationen](#)

- CLUSTER INFO

Gibt Informationen zum Status eines Knotens zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

[Weitere Informationen](#)

- CLUSTER KEYSLOT

Gibt den Hash-Slot für einen Schlüssel zurück.

[Weitere Informationen](#)

- **CLUSTER MYID**

Gibt die ID eines Knotens zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

[Weitere Informationen](#)

- **CLUSTER NODES**

Gibt die Clusterkonfiguration für einen Knoten zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

[Weitere Informationen](#)

- **CLUSTER REPLICAS**

Listet die Replikatknoten eines Hauptknotens auf. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

[Weitere Informationen](#)

- **CLUSTER SHARDS**

Gibt die Zuordnung von Cluster-Slots zu Shards zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

[Weitere Informationen](#)

- **CLUSTER SLOTS**

Gibt die Zuordnung von Cluster-Slots zu Knoten zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

[Weitere Informationen](#)

- **CLUSTER SLOT-STATS**

Ermöglicht die Nachverfolgung von Metriken pro Steckplatz für Schlüsselanzahl, CPU-Auslastung, eingehende Netzwerk-Bytes und ausgehende Netzwerk-Bytes.

[Weitere Informationen](#)

- READONLY

Ermöglicht schreibgeschützte Abfragen für eine Verbindung zu einem Valkey- oder Redis OSS-Cluster-Replikatknoten.

[Weitere Informationen](#)

- READWRITE

Aktiviert Lese-/Schreibabfragen für eine Verbindung zu einem Valkey- oder Redis OSS-Cluster-Replikatknoten.

[Weitere Informationen](#)

- SCRIPT SHOW

Gibt den ursprünglichen Quellcode eines Skripts im Skript-Cache zurück.

[Weitere Informationen](#)

Befehle zur Verbindungsverwaltung

- AUTH

Authentifiziert die Verbindung.

[Weitere Informationen](#)

- CLIENT GETNAME

Gibt den Namen der Verbindung zurück.

[Weitere Informationen](#)

- CLIENT REPLY

Weist den Server an, ob er auf Befehle antworten soll.

[Weitere Informationen](#)

- CLIENT SETNAME

Legt den Verbindungsnamen fest.

[Weitere Informationen](#)

- ECHO

Gibt die angegebene Zeichenfolge zurück.

[Weitere Informationen](#)

- HELLO

Handshakes mit dem Valkey- oder Redis OSS-Server.

[Weitere Informationen](#)

- PING

Gibt die Lebhaftigkeitsantwort des Servers zurück.

[Weitere Informationen](#)

- QUIT

Schließt die Verbindung.

[Weitere Informationen](#)

- RESET

Setzt die Verbindung zurück.

[Weitere Informationen](#)

- SELECT

Ändert die ausgewählte Datenbank.

[Weitere Informationen](#)

Generische Befehle

- COPY

Kopiert den Wert eines Schlüssels in einen neuen Schlüssel.

[Weitere Informationen](#)

- DEL

Löscht einen oder mehrere Schlüssel.

[Weitere Informationen](#)

- DUMP

Gibt eine serialisierte Darstellung des in einem Schlüssel gespeicherten Werts zurück.

[Weitere Informationen](#)

- EXISTS

Ermittelt, ob ein oder mehrere Schlüssel existieren.

[Weitere Informationen](#)

- EXPIRE

Legt die Ablaufzeit eines Schlüssels in Sekunden fest.

[Weitere Informationen](#)

- EXPIREAT

Legt die Ablaufzeit eines Schlüssels auf einen Unix-Zeitstempel fest.

[Weitere Informationen](#)

- EXPIRETIME

Gibt die Ablaufzeit eines Schlüssels als Unix-Zeitstempel zurück.

[Weitere Informationen](#)

- PERSIST

Entfernt die Ablaufzeit eines Schlüssels.

[Weitere Informationen](#)

- PEXPIRE

Legt die Ablaufzeit eines Schlüssels in Millisekunden fest.

[Weitere Informationen](#)

- PEXPIREAT

Legt die Ablaufzeit eines Schlüssels auf einen Unix-Millisekunden-Zeitstempel fest.

[Weitere Informationen](#)

- PEXPIRETIME

Gibt die Ablaufzeit eines Schlüssels als Unix-Millisekunden-Zeitstempel zurück.

[Weitere Informationen](#)

- PTTL

Gibt die Ablaufzeit eines Schlüssels in Millisekunden zurück.

[Weitere Informationen](#)

- RANDOMKEY

Gibt einen zufälligen Schlüsselnamen aus der Datenbank zurück.

[Weitere Informationen](#)

- RENAME

Benennt einen Schlüssel um und überschreibt das Ziel.

[Weitere Informationen](#)

- RENAMENX

Benennt einen Schlüssel nur um, wenn der Name des Zielschlüssels nicht vorhanden ist.

[Weitere Informationen](#)

- RESTORE

Erstellt einen Schlüssel aus der serialisierten Darstellung eines Werts.

[Weitere Informationen](#)

- SCAN

Iteriert über die Schlüsselnamen in der Datenbank.

[Weitere Informationen](#)

- SORT

Sortiert die Elemente in einer Liste, einem Set oder einem sortierten Set und speichert optional das Ergebnis.

[Weitere Informationen](#)

- SORT_R0

Gibt die sortierten Elemente einer Liste, eines Sets oder eines sortierten Sets zurück.

[Weitere Informationen](#)

- TOUCH

Gibt die Anzahl der vorhandenen Schlüssel aus den angegebenen Schlüsseln zurück, nachdem der Zeitpunkt des letzten Zugriffs aktualisiert wurde.

[Weitere Informationen](#)

- TTL

Gibt die Ablaufzeit eines Schlüssels in Sekunden zurück.

[Weitere Informationen](#)

- TYPE

Ermittelt den Typ des Wertes, der in einem Schlüssel gespeichert ist.

[Weitere Informationen](#)

- UNLINK

Löscht asynchron einen oder mehrere Schlüssel.

[Weitere Informationen](#)

Geodatenbefehle

- GEOADD

Fügt einem Geodatenindex ein oder mehrere Mitglieder hinzu. Der Schlüssel wird erstellt, wenn er nicht existiert.

[Weitere Informationen](#)

- GEODIST

Gibt den Abstand zwischen zwei Mitgliedern eines Geodatenindex zurück.

[Weitere Informationen](#)

- GEOHASH

Gibt Elemente aus einem Geodatenindex als Geohash-Zeichenfolgen zurück.

[Weitere Informationen](#)

- GEOPOS

Gibt den Längen- und Breitengrad von Elementen aus einem Geodatenindex zurück.

[Weitere Informationen](#)

- GEORADIUS

Fragt einen Geodatenindex nach Elementen ab, die sich innerhalb einer Entfernung von einer Koordinate befinden, und speichert optional das Ergebnis.

[Weitere Informationen](#)

- GEORADIUS_RO

Gibt Elemente aus einem Geodatenindex zurück, die sich innerhalb einer Entfernung von einer Koordinate befinden.

[Weitere Informationen](#)

- GEORADIUSBYMEMBER

Fragt einen Geodatenindex nach Elementen ab, die sich innerhalb einer Entfernung von einem Element befinden, und speichert optional das Ergebnis.

[Weitere Informationen](#)

- GEORADIUSBYMEMBER_RO

Gibt Elemente aus einem Geodatenindex zurück, die sich innerhalb einer Entfernung von einem Element befinden.

[Weitere Informationen](#)

- GEOSEARCH

Fragt einen Geodatenindex nach Elementen ab, die sich in einem Bereich eines Felds oder Kreises befinden.

[Weitere Informationen](#)

- GEOSEARCHSTORE

Fragt einen Geodatenindex nach Elementen ab, die sich in einem Bereich eines Felds oder Kreises befinden, und speichert optional das Ergebnis.

[Weitere Informationen](#)

Hash-Befehle

- HDEL

Löscht ein oder mehrere Felder und deren Werte aus einem Hash. Löscht den Hash, wenn keine Felder mehr vorhanden sind.

[Weitere Informationen](#)

- HEXISTS

Ermittelt, ob ein Feld in einem Hash existiert.

[Weitere Informationen](#)

- HGET

Gibt den Wert eines Felds in einem Hash zurück.

[Weitere Informationen](#)

- HGETALL

Gibt alle Felder und Werte in einem Hash zurück.

[Weitere Informationen](#)

- HINCRBY

Inkrementiert den Ganzzahlwert eines Felds in einem Hash um eine Zahl. Verwendet 0 als Anfangswert, wenn das Feld nicht existiert.

[Weitere Informationen](#)

- HINCRBYFLOAT

Inkrementiert den Gleitkommawert eines Felds um eine Zahl. Verwendet 0 als Anfangswert, wenn das Feld nicht existiert.

[Weitere Informationen](#)

- HKEYS

Gibt alle Felder in einem Hash zurück.

[Weitere Informationen](#)

- HLEN

Gibt die Anzahl der Felder in einem Hash zurück.

[Weitere Informationen](#)

- HMGET

Gibt die Werte aller Felder in einem Hash zurück.

[Weitere Informationen](#)

- HMSET

Legt die Werte mehrerer Felder fest.

[Weitere Informationen](#)

- HRANDFIELD

Gibt ein oder mehrere zufällige Felder aus einem Hash zurück.

[Weitere Informationen](#)

- HSCAN

Iteriert über Felder und Werte eines Hashs.

[Weitere Informationen](#)

- HSET

Erstellt oder ändert den Wert eines Felds in einem Hash.

[Weitere Informationen](#)

- HSETNX

Legt den Wert eines Felds in einem Hash nur fest, wenn das Feld nicht existiert.

[Weitere Informationen](#)

- HSTRLEN

Gibt die Länge des Werts eines Felds zurück.

[Weitere Informationen](#)

- HVALS

Gibt alle Werte in einem Hash zurück.

[Weitere Informationen](#)

HyperLogLog Befehle

- PFADD

Fügt einem Schlüssel Elemente hinzu. HyperLogLog Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- PFCOUNT

Gibt die ungefähre Kardinalität der Menge (n) zurück, die von dem/den HyperLogLog Schlüssel (n) beobachtet wurde.

[Weitere Informationen](#)

- PFMERGE

Führt einen oder mehrere HyperLogLog Werte zu einem einzigen Schlüssel zusammen.

[Weitere Informationen](#)

Listenbefehle

- BLMOVE

Löscht ein Element aus einer Liste, verschiebt es in eine andere Liste und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element verschoben wurde.

[Weitere Informationen](#)

- BLMPOP

Löscht das erste Element aus einer von mehreren Listen. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- BLPOP

Entfernt das erste Element in einer Liste und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- BRPOP

Entfernt das letzte Element in einer Liste und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- BRPOPLPUSH

Löscht ein Element aus einer Liste, verschiebt es in eine andere Liste und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- LINDEX

Gibt ein Element anhand seines Index aus einer Liste zurück.

[Weitere Informationen](#)

- LINSERT

Fügt ein Element vor oder nach einem anderen Element in eine Liste ein.

[Weitere Informationen](#)

- LLEN

Gibt die Länge einer Liste zurück.

[Weitere Informationen](#)

- LMOVE

Gibt ein Element zurück, nachdem es aus einer Liste entfernt und in eine andere verschoben wurde. Löscht die Liste, wenn das letzte Element verschoben wurde.

[Weitere Informationen](#)

- LMPOP

Gibt mehrere Elemente aus einer Liste zurück, nachdem sie entfernt wurden. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- LPOP

Gibt die ersten Elemente in einer Liste zurück, nachdem sie entfernt wurde. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- LPOS

Gibt den Index übereinstimmender Elemente in einer Liste zurück.

[Weitere Informationen](#)

- LPUSH

Stellt einer Liste ein oder mehrere Elemente voran. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- LPUSHX

Stellt einer Liste nur dann ein oder mehrere Elemente voran, wenn die Liste existiert.

[Weitere Informationen](#)

- LRANGE

Gibt einen Bereich von Elementen aus einer Liste zurück.

[Weitere Informationen](#)

- LREM

Entfernt Elemente aus einer Liste. Löscht die Liste, wenn das letzte Element entfernt wurde.

[Weitere Informationen](#)

- LSET

Legt den Wert eines Elements in einer Liste anhand seines Index fest.

[Weitere Informationen](#)

- LTRIM

Entfernt Elemente am Anfang und Ende einer Liste. Löscht die Liste, wenn alle Elemente gekürzt wurden.

[Weitere Informationen](#)

- RPOP

Gibt die letzten Elemente einer Liste zurück und entfernt sie. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- RPOPLPUSH

Gibt das letzte Element einer Liste zurück, nachdem es entfernt und in eine andere Liste verschoben wurde. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- RPUSH

Fügt ein oder mehrere Elemente an eine Liste an. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- RPU SHX

Fügt ein Element nur dann an eine Liste an, wenn die Liste existiert.

[Weitere Informationen](#)

Pub/Sub-Befehle

Note

PUBSUB-Befehle verwenden intern Sharded PUBSUB, sodass Kanalnamen gemischt werden.

- PUBLISH

Sendet eine Nachricht an einen Kanal.

[Weitere Informationen](#)

- PUBSUB CHANNELS

Gibt die aktiven Kanäle zurück.

[Weitere Informationen](#)

- PUBSUB NUMSUB

Gibt die Anzahl der Subscriber von Kanälen zurück.

[Weitere Informationen](#)

- PUBSUB SHARDCHANNELS

Gibt die aktiven Shard-Kanäle zurück.

[Weitere Informationen](#)

- PUBSUB SHARDNUMSUB

Gibt die Anzahl der Subscriber von Shard-Kanälen zurück.

[Weitere Informationen](#)

- SPUBLISH

Sendet eine Nachricht an einen Shard-Kanal.

[Weitere Informationen](#)

- SSUBSCRIBE

Lauscht nach Nachrichten, die auf Shard-Kanälen veröffentlicht wurden.

[Weitere Informationen](#)

- SUBSCRIBE

Lauscht nach Nachrichten, die auf Kanälen veröffentlicht wurden.

[Weitere Informationen](#)

- SUNSUBSCRIBE

Beendet das Lauschen nach Nachrichten, die an Shard-Kanäle gesendet wurden.

[Weitere Informationen](#)

- UNSUBSCRIBE

Beendet das Lauschen nach Nachrichten, die an Kanäle gesendet wurden.

[Weitere Informationen](#)

Befehle zur Skripterstellung

- EVAL

Führt ein serverseitiges Lua-Skript aus.

[Weitere Informationen](#)

- EVAL_R0

Führt ein schreibgeschütztes serverseitiges Lua-Skript aus.

[Weitere Informationen](#)

- EVALSHA

Führt ein serverseitiges Lua-Skript per Digest aus. SHA1

[Weitere Informationen](#)

- EVALSHA_RO

Führt ein schreibgeschütztes serverseitiges Lua-Skript per Digest aus. SHA1

[Weitere Informationen](#)

- SCRIPT EXISTS

Ermittelt, ob serverseitige Lua-Skripte im Skriptcache vorhanden sind.

[Weitere Informationen](#)

- SCRIPT FLUSH

Derzeit wird ein No-Op-Skript-Cache vom Service verwaltet.

[Weitere Informationen](#)

- SCRIPT LOAD

Lädt ein serverseitiges Lua-Skript in den Skript-Cache.

[Weitere Informationen](#)

Befehle zur Serververwaltung

Note

Bei der Verwendung knotenbasierter ElastiCache Cluster für Valkey und Redis OSS müssen vom Client Flush-Befehle an jeden Primärserver gesendet werden, um alle Schlüssel zu leeren. ElastiCache Serverless für Valkey und Redis OSS funktioniert unterschiedlich, da es die zugrunde liegende Cluster-Topologie wegabstrahiert. Das Ergebnis ist, dass in ElastiCache Serverless FLUSHDB und mit FLUSHALL Befehlen immer alle Schlüssel im Cluster geleert werden. Aus diesem Grund können Flush-Befehle nicht in eine serverlose Transaktion aufgenommen werden.

- ACL CAT

Listet die ACL-Kategorien oder die Befehle innerhalb einer Kategorie auf.

[Weitere Informationen](#)

- ACL GENPASS

Generiert ein pseudozufälliges, sicheres Passwort, das zur Identifizierung von ACL-Benutzern verwendet werden kann.

[Weitere Informationen](#)

- ACL GETUSER

Listet die ACL-Regeln eines Benutzers auf.

[Weitere Informationen](#)

- ACL LIST

Gibt die effektiven Regeln im ACL-Dateiformat aus.

[Weitere Informationen](#)

- ACL USERS

Listet alle ACL-Benutzer auf.

[Weitere Informationen](#)

- ACL WHOAMI

Gibt den authentifizierten Benutzernamen der aktuellen Verbindung zurück.

[Weitere Informationen](#)

- DBSIZE

Gibt die Anzahl der Schlüssel in der aktuell ausgewählten Datenbank zurück. Es kann nicht garantiert werden, dass diese Operation in allen Slots atomar abläuft.

[Weitere Informationen](#)

- COMMAND

Gibt detaillierte Informationen zu allen Befehlen zurück.

[Weitere Informationen](#)

- `COMMAND COUNT`

Gibt eine Anzahl von Befehlen zurück.

[Weitere Informationen](#)

- `COMMAND DOCS`

Gibt durch Dokumente belegte Informationen zu einem, mehreren oder allen Befehlen zurück.

[Weitere Informationen](#)

- `COMMAND GETKEYS`

Extrahiert die Schlüsselnamen aus einem willkürlichen Befehl.

[Weitere Informationen](#)

- `COMMAND GETKEYSANDFLAGS`

Extrahiert die Schlüsselnamen und Zugriffs-Flags für einen willkürlichen Befehl.

[Weitere Informationen](#)

- `COMMAND INFO`

Gibt Informationen zu einem, mehreren oder allen Befehlen zurück.

[Weitere Informationen](#)

- `COMMAND LIST`

Gibt eine Liste von Befehlsnamen zurück.

[Weitere Informationen](#)

- `COMMANDLOG`

Ein Container für Befehlsprotokollbefehle.

[Weitere Informationen](#)

- `COMMANDLOG GET`

Gibt die Einträge des angegebenen Befehlsprotokolls zurück.

[Weitere Informationen](#)

- `COMMANDLOG HELP`

Zeigt hilfreichen Text zu den verschiedenen Unterbefehlen an.

[Weitere Informationen](#)

- `COMMANDLOG LEN`

Gibt die Anzahl der Einträge im angegebenen Typ von Befehlsprotokoll zurück.

[Weitere Informationen](#)

- `COMMANDLOG RESET`

Löscht alle Einträge aus dem angegebenen Typ von Befehlsprotokoll.

[Weitere Informationen](#)

- `FLUSHALL`

Entfernt alle Schlüssel aus allen Datenbanken. Es kann nicht garantiert werden, dass diese Operation in allen Slots atomar abläuft.

[Weitere Informationen](#)

- `FLUSHDB`

Entfernt alle Schlüssel aus der aktuellen Datenbank. Es kann nicht garantiert werden, dass diese Operation in allen Slots atomar abläuft.

[Weitere Informationen](#)

- `INFO`

Gibt Informationen und Statistiken über den Server zurück.

[Weitere Informationen](#)

- `LOLWUT`

Zeigt Computergrafiken und die Valkey- oder Redis-OSS-Version an.

[Weitere Informationen](#)

- `ROLE`

Gibt die Replikationsrolle zurück.

[Weitere Informationen](#)

- TIME

Gibt die Serverzeit zurück.

[Weitere Informationen](#)

Set-Befehle

- SADD

Fügt einem Set ein oder mehrere Mitglieder hinzu. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- SCARD

Gibt die Anzahl der Mitglieder in einem Set zurück.

[Weitere Informationen](#)

- SDIFF

Gibt die Differenz zwischen mehreren Sets zurück.

[Weitere Informationen](#)

- SDIFFSTORE

Speichert die Differenz zwischen mehreren Sets in einem Schlüssel.

[Weitere Informationen](#)

- SINTER

Gibt den Schnittpunkt mehrerer Sets zurück.

[Weitere Informationen](#)

- SINTERCARD

Gibt die Anzahl der Mitglieder des Schnittpunkts mehrerer Sets zurück.

[Weitere Informationen](#)

- SINTERSTORE

Speichert den Schnittpunkt mehrerer Sets in einem Schlüssel.

[Weitere Informationen](#)

- SISMEMBER

Ermittelt, ob ein Mitglied zu einem Set gehört.

[Weitere Informationen](#)

- SMEMBERS

Gibt alle Mitglieder eines Sets zurück.

[Weitere Informationen](#)

- SMISMEMBER

Ermittelt, ob mehrere Mitglieder zu einem Set gehören.

[Weitere Informationen](#)

- SMOVE

Verschiebt ein Mitglied von einem Set in ein anderes.

[Weitere Informationen](#)

- SPOP

Gibt ein oder mehrere zufällige Mitglieder aus einer Gruppe zurück, nachdem sie entfernt wurden.
Löscht das Set, wenn das letzte Mitglied gelöscht wurde.

[Weitere Informationen](#)

- SRANDMEMBER

Ruft ein oder mehrere zufällige Mitglieder aus einem Set ab.

[Weitere Informationen](#)

- SREM

Entfernt ein oder mehrere Mitglieder aus einem Set. Löscht das Set, wenn das letzte Mitglied entfernt wurde.

[Weitere Informationen](#)

- SSCAN

Iteriert über Mitglieder eines Sets.

[Weitere Informationen](#)

- SUNION

Gibt die Vereinigung mehrerer Sets zurück.

[Weitere Informationen](#)

- SUNIONSTORE

Speichert die Vereinigung mehrerer Sets in einem Schlüssel.

[Weitere Informationen](#)

Befehle zu Sorted Sets

- BZMPOP

Entfernt ein Mitglied aus einem oder mehreren sortierten Sets und gibt es nach Score zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht das sortierte Set, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- BZPOPMAX

Entfernt das Element mit dem höchsten Score aus einem oder mehreren sortierten Sets und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht das sortierte Set, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- BZPOPMIN

Entfernt das Element mit dem niedrigsten Score aus einem oder mehreren sortierten Sets und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht das sortierte Set, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- ZADD

Fügt einem sortierten Set ein oder mehrere Mitglieder hinzu oder aktualisiert deren Scores. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- ZCARD

Gibt die Anzahl der Mitglieder in einem sortierten Set zurück.

[Weitere Informationen](#)

- ZCOUNT

Gibt die Anzahl der Mitglieder in einem sortierten Set zurück, deren Scores innerhalb eines bestimmten Bereichs liegen.

[Weitere Informationen](#)

- ZDIFF

Gibt den Unterschied zwischen mehreren sortierten Sets zurück.

[Weitere Informationen](#)

- ZDIFFSTORE

Speichert den Unterschied mehrerer sortierter Sets in einem Schlüssel.

[Weitere Informationen](#)

- ZINCRBY

Inkrementiert den Score eines Mitglieds in einem sortierten Set.

[Weitere Informationen](#)

- ZINTER

Gibt den Schnittpunkt mehrerer sortierter Sets zurück.

[Weitere Informationen](#)

- ZINTERCARD

Gibt die Anzahl der Mitglieder des Schnittpunkts mehrerer sortierter Sets zurück.

[Weitere Informationen](#)

- ZINTERSTORE

Speichert den Schnittpunkt mehrerer sortierter Sets in einem Schlüssel.

[Weitere Informationen](#)

- ZLEXCOUNT

Gibt die Anzahl der Elemente in einem sortierten Set innerhalb eines lexikografischen Bereichs zurück.

[Weitere Informationen](#)

- ZMPOP

Gibt die Elemente mit dem höchsten oder niedrigsten Score aus einem oder mehreren sortierten Sets zurück, nachdem sie entfernt wurden. Löscht das sortierte Set, wenn das letzte Mitglied gelöscht wurde.

[Weitere Informationen](#)

- ZMSCORE

Gibt den Score eines oder mehrerer Mitglieder eines sortierten Sets zurück.

[Weitere Informationen](#)

- ZPOPMAX

Gibt die Mitglieder mit dem höchsten Score aus einem sortierten Set zurück, nachdem sie entfernt wurden. Löscht das sortierte Set, wenn das letzte Mitglied gelöscht wurde.

[Weitere Informationen](#)

- ZPOPMIN

Gibt die Mitglieder mit dem niedrigsten Score aus einem sortierten Set zurück, nachdem sie entfernt wurden. Löscht das sortierte Set, wenn das letzte Mitglied gelöscht wurde.

[Weitere Informationen](#)

- ZRANDMEMBER

Gibt ein oder mehrere zufällige Mitglieder aus einem sortierten Set zurück.

[Weitere Informationen](#)

- ZRANGE

Gibt Elemente in einem sortierten Set innerhalb eines Indexbereichs zurück.

[Weitere Informationen](#)

- ZRANGEBYLEX

Gibt Elemente in einem sortierten Set innerhalb eines lexikografischen Bereichs zurück.

[Weitere Informationen](#)

- ZRANGEBYSCORE

Gibt Elemente in einem sortierten Set innerhalb eines Score-Bereichs zurück.

[Weitere Informationen](#)

- ZRANGESTORE

Speichert einen Bereich von Elementen aus einem sortierten Set in einem Schlüssel.

[Weitere Informationen](#)

- ZRANK

Gibt den Index eines Elements in einem sortierten Set zurück, sortiert nach aufsteigenden Scores.

[Weitere Informationen](#)

- ZREM

Entfernt ein oder mehrere Elemente aus einem sortierten Set. Löscht das sortierte Set, wenn alle Elemente entfernt wurden.

[Weitere Informationen](#)

- ZREMRANGEBYLEX

Entfernt Elemente in einem sortierten Set innerhalb eines lexikografischen Bereichs. Löscht das sortierte Set, wenn alle Elemente entfernt wurden.

[Weitere Informationen](#)

- ZREMRANGEBYRANK

Entfernt Elemente in einem sortierten Sete innerhalb eines Indexbereichs. Löscht das sortierte Set, wenn alle Elemente entfernt wurden.

[Weitere Informationen](#)

- ZREMRANGEBYSCORE

Entfernt Elemente aus einem sortierten Set innerhalb eines Score-Bereichs. Löscht das sortierte Set, wenn alle Elemente entfernt wurden.

[Weitere Informationen](#)

- ZREVRANGE

Gibt Elemente in einem sortierten Set innerhalb eines Indexbereichs in umgekehrter Reihenfolge zurück.

[Weitere Informationen](#)

- ZREVRANGEBYLEX

Gibt Elemente in einem sortierten Set innerhalb eines lexikografischen Bereichs in umgekehrter Reihenfolge zurück.

[Weitere Informationen](#)

- ZREVRANGEBYSCORE

Gibt Elemente in einem sortierten Set innerhalb eines Score-Bereichs in umgekehrter Reihenfolge zurück.

[Weitere Informationen](#)

- ZREVRANK

Gibt den Index eines Elements in einem sortierten Set zurück, sortiert nach absteigenden Scores.

[Weitere Informationen](#)

- ZSCAN

Iteriert über Elemente und Scores eines sortierten Sets.

[Weitere Informationen](#)

- ZSCORE

Gibt den Score eines Elements in einem sortierten Set zurück.

[Weitere Informationen](#)

- ZUNION

Gibt die Vereinigung mehrerer sortierter Sets zurück.

[Weitere Informationen](#)

- ZUNIONSTORE

Speichert die Vereinigung mehrerer sortierter Sets in einem Schlüssel.

[Weitere Informationen](#)

Stream-Befehle

- XACK

Gibt die Anzahl der Nachrichten zurück, die vom Mitglied der Verbrauchergruppe eines Streams erfolgreich bestätigt wurden.

[Weitere Informationen](#)

- XADD

Hängt eine neue Nachricht an einen Stream an. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- XAUTOCLAIM

Ändert oder erwirbt den Besitz von Nachrichten in einer Verbrauchergruppe, als ob die Nachrichten als Mitglied der Verbrauchergruppe zugestellt würden.

[Weitere Informationen](#)

- XCLAIM

Ändert oder erwirbt den Besitz einer Nachricht in einer Verbrauchergruppe, als ob die Nachricht als Mitglied der Verbrauchergruppe zugestellt würde.

[Weitere Informationen](#)

- XDEL

Gibt die Anzahl der Nachrichten zurück, nachdem sie aus einem Stream entfernt wurden.

[Weitere Informationen](#)

- XGROUP CREATE

Erstellt eine Verbrauchergruppe.

[Weitere Informationen](#)

- XGROUP CREATECONSUMER

Erstellt einen Verbraucher in einer Verbrauchergruppe.

[Weitere Informationen](#)

- XGROUP DELCONSUMER

Löscht einen Verbraucher aus einer Verbrauchergruppe.

[Weitere Informationen](#)

- XGROUP DESTROY

Löscht eine Verbrauchergruppe.

[Weitere Informationen](#)

- XGROUP SETID

Legt die zuletzt zugestellte ID einer Verbrauchergruppe fest.

[Weitere Informationen](#)

- XINFO CONSUMERS

Gibt eine Liste der Verbraucher in einer Verbrauchergruppe zurück.

[Weitere Informationen](#)

- XINFO GROUPS

Gibt eine Liste der Verbrauchergruppen eines Streams zurück.

[Weitere Informationen](#)

- XINFO STREAM

Gibt Informationen zu einem Stream zurück.

[Weitere Informationen](#)

- XLEN

Gibt die Anzahl der Nachrichten in einem Stream zurück.

[Weitere Informationen](#)

- XPENDING

Gibt die Informationen und Einträge aus der Liste der ausstehenden Einträge einer Stream-Verbrauchergruppe zurück.

[Weitere Informationen](#)

- XRANGE

Gibt die Nachrichten aus einem Stream innerhalb eines Bereichs von zurück. IDs

[Weitere Informationen](#)

- XREAD

Gibt Nachrichten aus mehreren Streams zurück, deren IDs Anzahl größer als die angeforderten ist. Blockiert, bis eine Nachricht anderweitig verfügbar ist.

[Weitere Informationen](#)

- XREADGROUP

Gibt neue oder historische Nachrichten aus einem Stream für einen Verbraucher in einer Gruppe zurück. Blockiert, bis eine Nachricht anderweitig verfügbar ist.

[Weitere Informationen](#)

- XREVRANGE

Gibt die Nachrichten aus einem Stream innerhalb eines Bereichs von IDs in umgekehrter Reihenfolge zurück.

[Weitere Informationen](#)

- XTRIM

Löscht Nachrichten am Anfang eines Streams.

[Weitere Informationen](#)

Zeichenfolgenbefehle

- APPEND

Hängt eine Zeichenfolge an den Wert eines Schlüssels an. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- DECR

Dekrementiert den Ganzzahlwert eines Schlüssels um eins. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

[Weitere Informationen](#)

- DECRBY

Dekrementiert eine Zahl vom Ganzzahlwert eines Schlüssels. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

[Weitere Informationen](#)

- GET

Gibt den Zeichenfolgenwert eines Schlüssels zurück.

[Weitere Informationen](#)

- GETDEL

Gibt den Zeichenfolgenwert eines Schlüssels zurück, nachdem der Schlüssel gelöscht wurde.

[Weitere Informationen](#)

- GETEX

Gibt den Zeichenfolgenwert eines Schlüssels zurück, nachdem dessen Ablaufzeit festgelegt wurde.

[Weitere Informationen](#)

- GETRANGE

Gibt eine Teilzeichenfolge der Zeichenfolge zurück, die in einem Schlüssel gespeichert ist.

[Weitere Informationen](#)

- GETSET

Gibt den vorherigen Zeichenfolgenwert eines Schlüssels zurück, nachdem dieser auf einen neuen Wert festgelegt wurde.

[Weitere Informationen](#)

- INCR

Inkrementiert den Ganzzahlwert eines Schlüssels um eins. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

[Weitere Informationen](#)

- INCRBY

Inkrementiert den Ganzzahlwert eines Schlüssels um eine Zahl. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

[Weitere Informationen](#)

- INCRBYFLOAT

Inkrementiert den Gleitkommawert eines Schlüssels um eine Zahl. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

[Weitere Informationen](#)

- LCS

Findet die längste gemeinsame Teilzeichenfolge.

[Weitere Informationen](#)

- MGET

Gibt atomar die Zeichenfolgenwerte eines oder mehrerer Schlüssel zurück.

[Weitere Informationen](#)

- MSET

Erstellt oder ändert atomar die Zeichenfolgenwerte eines oder mehrerer Schlüssel.

[Weitere Informationen](#)

- MSETNX

Ändert die Zeichenfolgenwerte eines oder mehrerer Schlüssel nur dann atomar, wenn alle Schlüssel nicht existieren.

[Weitere Informationen](#)

- PSETEX

Legt sowohl den Zeichenfolgenwert als auch die Ablaufzeit eines Schlüssels in Millisekunden fest. Der Schlüssel wird erstellt, wenn er nicht existiert.

[Weitere Informationen](#)

- SET

Legt den Zeichenfolgenwert eines Schlüssels fest, wobei sein Typ ignoriert wird. Der Schlüssel wird erstellt, wenn er nicht existiert.

[Weitere Informationen](#)

- SETEX

Legt den Zeichenfolgenwert und die Ablaufzeit eines Schlüssels fest. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- SETNX

Legt den Zeichenfolgenwert eines Schlüssels nur fest, wenn der Schlüssel nicht existiert.

[Weitere Informationen](#)

- SETRANGE

Überschreibt einen Teil eines Zeichenfolgenwerts durch einen anderen um einen Offset. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- STRLEN

Gibt die Länge eines Zeichenfolgenwerts zurück.

[Weitere Informationen](#)

- SUBSTR

Gibt eine Teilzeichenfolge aus einem Zeichenfolgenwert zurück.

[Weitere Informationen](#)

Transaktionsbefehle

- DISCARD

Verwirft eine Transaktion.

[Weitere Informationen](#)

- EXEC

Führt alle Befehle in einer Transaktion aus.

[Weitere Informationen](#)

- MULTI

Startet eine Transaktion.

[Weitere Informationen](#)

Eingeschränkte Valkey- und Redis OSS-Befehle

ElastiCache schränkt den Zugriff auf bestimmte Cache-Engine-spezifische Befehle ein, die erweiterte Rechte erfordern, um ein Managed Service-Erlebnis zu bieten. Für Caches, auf denen Redis OSS ausgeführt wird, sind die folgenden Befehle nicht verfügbar:

- `acl setuser`
- `acl load`
- `acl save`
- `acl deluser`
- `bgrewriteaof`
- `bgsave`
- `cluster addslot`
- `cluster addslotsrange`
- `cluster bumpepoch`
- `cluster delslot`
- `cluster delslotsrange`
- `cluster failover`
- `cluster flushslots`
- `cluster forget`
- `cluster links`
- `cluster meet`
- `cluster setslot`
- `config`
- `debug`
- `migrate`
- `psync`
- `replicaof`
- `save`

- `slaveof`
- `shutdown`
- `sync`

Darüber hinaus sind die folgenden Befehle für Serverless-Caches nicht verfügbar:

- `acl log`
- `client caching`
- `client getredir`
- `client id`
- `client info`
- `client kill`
- `client list`
- `client no-evict`
- `client pause`
- `client tracking`
- `client trackinginfo`
- `client unblock`
- `client unpause`
- `cluster count-failure-reports`
- `commandlog`
- `commandlog get`
- `commandlog help`
- `commandlog len`
- `commandlog reset`
- `fcall`
- `fcall_ro`
- `function`
- `function delete`
- `function dump`
- `function flush`

- `function help`
- `function kill`
- `function list`
- `function load`
- `function restore`
- `function stats`
- `keys`
- `lastsave`
- `latency`
- `latency doctor`
- `latency graph`
- `latency help`
- `latency histogram`
- `latency history`
- `latency latest`
- `latency reset`
- `memory`
- `memory doctor`
- `memory help`
- `memory malloc-stats`
- `memory purge`
- `memory stats`
- `memory usage`
- `monitor`
- `move`
- `object`
- `object encoding`
- `object freq`
- `object help`
- `object idleTime`

- `object refcount`
- `pfdebug`
- `pfselftest`
- `psubscribe`
- `pubsub numpat`
- `punsubscribe`
- `script kill`
- `slowlog`
- `slowlog get`
- `slowlog help`
- `slowlog len`
- `slowlog reset`
- `swapdb`
- `wait`

Unterstützte Memcached-Befehle

ElastiCache Serverless for Memcached unterstützt alle [Memcached-Befehle](#) in Open Source Memcached 1.6 mit Ausnahme der folgenden:

- Client-Verbindungen erfordern TLS, weshalb das UDP-Protokoll nicht unterstützt wird.
- Das Binärprotokoll wird nicht unterstützt, da es in Memcached 1.6 offiziell [veraltet](#) ist.
- GET/GETS-Befehle sind auf 16 KB begrenzt, um einen möglichen DoS-Angriff auf den Server durch Abrufen einer großen Anzahl von Schlüsseln zu vermeiden.
- Ein verzögerter `flush_all`-Befehl wird mit `CLIENT_ERROR` zurückgewiesen.
- Befehle, die die Engine konfigurieren oder interne Informationen zum Engine-Status oder zu Protokollen preisgeben, werden nicht unterstützt. Dazu gehören:
 - Für STATS-Befehle werden nur `stats` und `stats reset` unterstützt. Für andere Varianten wird `ERROR` zurückgegeben.
 - `lru / lru_crawler` – Änderung der LRU- und LRU-Crawler-Einstellungen
 - `watch` – beobachtet Memcached-Serverprotokolle
 - `verbosity` – konfiguriert die Protokollstufe des Servers

- me- Der Befehl meta debug (me) wird nicht unterstützt

Konfiguration und Grenzwerte für Valkey und Redis OSS

Die Valkey- und Redis OSS-Engines bieten jeweils eine Reihe von Konfigurationsparametern, von denen einige ElastiCache für Redis OSS modifizierbar sind und andere nicht modifizierbar sind, um eine stabile Leistung und Zuverlässigkeit zu gewährleisten.

Serverless-Caches

Für serverlose Caches werden keine Parametergruppen verwendet und die gesamte Valkey- oder Redis-OSS-Konfiguration kann nicht geändert werden. Die folgenden Valkey- oder Redis-OSS-Parameter sind vorhanden:

Name	Details	Description
acl-pubsub-default	allchannels	Standardberechtigungen für Pub/Sub-Kanäle für ACL-Benutzer im Cache.
client-output-buffer-limit	normal 0 0 0 pubsub 32mb 8mb 60	Normale Clients haben kein Pufferlimit. PUB/SUB Clients werden getrennt, wenn sie den 32 MiB-Backlog oder den 8 MiB-Backlog für 60 Sekunden überschreiten.
client-query-buffer-limit	1 GiB	Die maximale Größe eines einzelnen Client-Abfragepuffers. Darüber hinaus können Kunden keine Anfrage mit mehr als 3.999 Argumenten stellen.
cluster-allow-pubsubshard-when-down	yes	Dadurch kann der Cache Pub/Sub-Datenverkehr bereitstellen, während der Cache teilweise ausgefallen ist.
cluster-allow-read	yes	Dadurch kann der Cache Lesedatenverkehr bereitstellen, während der Cache teilweise ausgefallen ist.

Name	Details	Description
<code>s-when-do-wn</code>		
<code>cluster-enabled</code>	yes	Für alle Serverless-Caches ist der Clustermodus aktiviert, sodass sie ihre Daten transparent auf mehrere Backend-Shards partitionieren können. Alle Slots werden den Clients so angezeigt, als gehörten sie einem einzigen virtuellen Knoten.
<code>cluster-require-full-coverage</code>	no	Wenn der Keyspace teilweise ausgefallen ist (d. h. auf mindestens einen Hash-Slot kann nicht zugegriffen werden), akzeptiert der Cache weiterhin Abfragen für den Teil des Keyspace, der noch abgedeckt ist. Der gesamte Keyspace wird immer von einem einzigen virtuellen Knoten in <code>cluster slots</code> „abgedeckt“.
<code>lua-time-limit</code>	5000	Die maximale Ausführungszeit für ein Lua-Skript in Millisekunden, bevor Maßnahmen zum Stoppen des Skripts ergriffen ElastiCache werden. Wenn sie überschritten <i>lua-time-limit</i> wird, geben alle Valkey- oder Redis OSS-Befehle möglicherweise einen Fehler der Form <code>____-BUSY</code> zurück. Da dieser Status viele wichtige Valkey- oder Redis-OSS-Operationen stören kann, ElastiCache wird zunächst ein <code>SCRIPT KILL</code>-Befehl ausgegeben. Wenn dies nicht erfolgreich ist, ElastiCache wird Valkey oder Redis OSS zwangsweise neu gestartet.

Name	Details	Description
maxclients	65000	Die maximale Anzahl von Clients, die zu jedem beliebigen Zeitpunkt mit dem Cache verbunden sein können. Weitere Verbindungen können erfolgreich hergestellt werden oder auch nicht.
maxmemory-policy	volatile-lru	Elemente mit einem TTL-Satz werden nach einer Schätzung least-recently-used (LRU) gelöscht, wenn das Speicherlimit eines Caches erreicht ist.
notify-keyspace-events	(eine leere Zeichenfolge)	Keyspace-Ereignisse werden in Serverless-Caches derzeit nicht unterstützt.
port	Primärer Port: 6379 Leseport: 6380	Serverless-Caches kündigen zwei Ports mit demselben Hostnamen an. Der primäre Port lässt Schreib- und Lesevorgänge zu, wohingegen der Leseport mithilfe des Befehls READONLY letztendlich konsistente Lesevorgänge mit niedrigerer Latenz ermöglicht.
proto-max-bulk-len	512 MiB	Die maximale Größe einer einzelnen Elementanforderung.
timeout	0	Zu einer bestimmten Leerlaufzeit werden die Verbindungen mit Clients nicht gewaltsam getrennt, sie können jedoch aus Gründen des Lastausgleichs im stabilen Zustand getrennt werden.

Darüber hinaus gelten die folgenden Limits:

Name	Details	Description
Größe pro Cache	5.000 GiB	Maximale Datenmenge, die pro serverlosem Cache gespeichert werden kann.
Größe pro Steckplatz	32 GiB	Die maximale Größe eines einzelnen Valkey- oder Redis OSS-Hash-Slots. Clients, die versuchen, mehr Daten auf einem einzelnen Valkey- oder Redis-OSS-Steckplatz zu speichern, lösen die Räumungsrichtlinie für den Steckplatz aus, und wenn keine Schlüssel entfernt werden können, wird ein Fehler wegen Speichermangel () angezeigt. OOM
ECPU pro Cache	15.000.000 ECPU/Sekunde	ElastiCache Metrisch für Verarbeitungseinheiten (ECPU) Die Anzahl der von Ihren Anfragen ECPU's verbrauchten Daten hängt von der vCPU vCPU-Zeit und der Menge der übertragenen Daten ab.
ECPU pro Steckplatz	30.000 bis 90.000 ECPU/Sekunde	Maximal 30.000 ECPU's/second pro Steckplatz oder 90.000 ECPU's/second bei Verwendung von Read from Replica mit READONLY-Verbindungen.
Argumente pro Anfrage	3.999	Maximale Anzahl von Argumenten pro Anfrage. Clients, die mehr Argumente pro Anfrage senden, erhalten eine Fehlermeldung.
Länge des Schlüsselnamens	4 KiB	Die maximale Größe für einen einzelnen Valkey- oder Redis-OSS-Schlüssel oder Kanalnamen. Clients, die auf Schlüssel verweisen, die größer sind, wird eine Fehlermeldung angezeigt.
Größe des Lua-Skripts	4 MiB	Die maximale Größe eines einzelnen Valkey- oder Redis OSS-Lua-Skripts. Versuche, ein

Name	Details	Description
		Lua-Skript zu laden, das größer ist als dieses, führen zu einer Fehlermeldung.

Knotenbasierte Cluster

Informationen zu knotenbasierten Clustern finden Sie unter [Valkey- und Redis OSS-Parameter](#). Die Standardwerte der Konfigurationsparameter und die konfigurierbaren Werte. Die Standardwerte werden generell empfohlen, es sei denn, Sie haben einen bestimmten Anwendungsfall, bei dem sie außer Kraft gesetzt werden müssen.

IPv6 Kundenbeispiele für Valkey, Memcached und Redis OSS

ElastiCache ist mit Valkey, Memcached und Redis OSS kompatibel. Das bedeutet, dass Clients, die IPv6 Verbindungen unterstützen, in der Lage sein sollten, eine Verbindung zu Clustern herzustellen, die für Memcached IPv6 aktiviert sind ElastiCache. Bei der Interaktion mit IPv6 aktivierten Ressourcen gibt es einige Vorsichtsmaßnahmen, die es zu beachten gilt.

Empfehlungen zur Konfiguration von [Valkey- und Redis OSS-Clients für Ressourcen finden Sie im Blogbeitrag Best Practices](#) für Valkey- und Redis-Clients im AWS Datenbank-Blog. ElastiCache

Im Folgenden finden Sie bewährte Methoden für die Interaktion mit IPv6 aktivierten ElastiCache Ressourcen mithilfe häufig verwendeter Open-Source-Clientbibliotheken.

Validierte Kunden mit Valkey und Redis OSS

ElastiCache ist mit Valkey und dem Open-Source-Redis OSS kompatibel. Das bedeutet, dass Valkey- und Open-Source-Redis-OSS-Clients, die IPv6 Verbindungen unterstützen, in der Lage sein sollten, eine Verbindung zu für Redis IPv6 aktivierten ElastiCache OSS-Clustern herzustellen. Darüber hinaus wurden mehrere der beliebtesten Python- und Java-Clients speziell getestet und validiert, sodass sie mit allen unterstützten Netzwerktypkonfigurationen (IPv4 nur, IPv6 nur und Dual Stack) funktionieren.

Die folgenden Clients wurden speziell dafür validiert, dass sie mit allen unterstützten Netzwerktypkonfigurationen für Valkey und Redis OSS funktionieren.

Validierte Clients:

- [Redis Py \(\) – 4.1.2](#)
- [Lettuce – Version: 6.1.6.RELEASE](#)
- [Jedis – Version: 3.6.0](#)

Bewährte Methoden für Kunden (Valkey und Redis OSS)

Lernen Sie bewährte Methoden für gängige Szenarien kennen und folgen Sie den Codebeispielen einiger der beliebtesten Open-Source-Clientbibliotheken von Valkey und Redis OSS (redis-py und Lettuce) sowie von Best Practices für die Interaktion mit ElastiCache Ressourcen mit häufig verwendeten Open-Source-Memcached-Clientbibliotheken. PHPRedis

Themen

- [Große Anzahl von Verbindungen \(Valkey und Redis OSS\)](#)
- [Cluster-Client-Erkennung und exponentielles Backoff \(Valkey und Redis OSS\)](#)
- [Konfigurieren Sie ein clientseitiges Timeout \(Valkey und Redis OSS\)](#)
- [Konfigurieren Sie ein serverseitiges Leerlauf-Timeout \(Valkey und Redis OSS\)](#)
- [Lua-Skripte](#)
- [Speichern großer zusammengesetzter Artikel \(Valkey und Redis OSS\)](#)
- [Konfiguration des Lettuce-Clients \(Valkey und Redis OSS\)](#)
- [Konfiguration eines bevorzugten Protokolls für Dual-Stack-Cluster \(Valkey und Redis OSS\)](#)

Große Anzahl von Verbindungen (Valkey und Redis OSS)

Serverlose Caches und individuelle Caches ElastiCache für Redis OSS-Knoten unterstützen bis zu 65.000 gleichzeitige Client-Verbindungen. Zur Optimierung der Leistung empfehlen wir jedoch, dass Client-Anwendungen nicht ständig mit diesem Verbindungsaufkommen arbeiten. Valkey und Redis OSS verfügen jeweils über einen Single-Thread-Prozess, der auf einer Ereignisschleife basiert, in der eingehende Client-Anfragen sequentiell bearbeitet werden. Das bedeutet, dass die Antwortzeit eines bestimmten Clients mit zunehmender Anzahl verbundener Clients länger wird.

Sie können die folgenden Maßnahmen ergreifen, um einen Verbindungsengpass auf einem Valkey- oder Redis OSS-Server zu vermeiden:

- Führen Sie Lesevorgänge von Read Replicas aus durch. Dies kann durch die Verwendung der ElastiCache Leser-Endpunkte im deaktivierten Clustermodus oder durch die Verwendung von

Replikaten für Lesevorgänge im aktivierten Clustermodus, einschließlich eines serverlosen Caches, erreicht werden.

- Verteilen Sie den Schreibdatenverkehr auf mehrere Primärknoten. Es gibt zwei Methoden dafür: Sie können einen Valkey- oder Redis-OSS-Cluster mit mehreren Shards und einem Client verwenden, der den Clustermodus unterstützt. Sie können auch im deaktivierten Cluster-Modus mit clientseitigem Sharding auf mehrere Primärknoten schreiben. Dies erfolgt automatisch in einem Serverless-Cache.
- Verwenden Sie einen Verbindungspool, sofern dieser in Ihrer Client-Bibliothek verfügbar ist.

Im Allgemeinen ist das Erstellen einer TCP-Verbindung im Vergleich zu typischen Valkey- oder Redis OSS-Befehlen ein rechenintensiver Vorgang. Beispielsweise ist die Bearbeitung einer SET/GET Anfrage um eine Größenordnung schneller, wenn eine bestehende Verbindung wiederverwendet wird. Durch die Verwendung eines Client-Verbindungspools mit begrenzter Größe wird der Aufwand für die Verbindungsverwaltung reduziert. Außerdem wird dadurch die Anzahl der gleichzeitig eingehenden Verbindungen von der Client-Anwendung begrenzt.

Das folgende Codebeispiel PHPRedis zeigt, dass für jede neue Benutzeranfrage eine neue Verbindung erstellt wird:

```
$redis = new Redis();
if ($redis->connect($HOST, $PORT) != TRUE) {
    //ERROR: connection failed
    return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Wir haben diesen Code in einer Schleife auf einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance verglichen, die mit einem Graviton2 (m6g.2xlarge) für Redis OSS-Knoten verbunden ist. ElastiCache Client und Server wurden in derselben Availability Zone platziert. Die durchschnittliche Latenz des gesamten Vorgangs betrug 2,82 Millisekunden.

Beim Aktualisieren des Codes und bei Verwendung persistenter Verbindungen sowie eines Verbindungspools betrug die durchschnittliche Latenz des gesamten Vorgangs 0,21 Millisekunden:

```
$redis = new Redis();
if ($redis->pconnect($HOST, $PORT) != TRUE) {
```

```
// ERROR: connection failed
return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Erforderliche Konfigurationen für redis.ini:

- `redis.pconnect.pooling_enabled=1`
- `redis.pconnect.connection_limit=10`

Der folgende Code ist ein Beispiel für einen [Redis-py-Verbindungspool](#):

```
conn = Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10))
conn.set(key, value)
```

Der folgende Code ist ein Beispiel für einen [Lettuce-Verbindungspool](#):

```
RedisClient client = RedisClient.create(RedisURI.create(HOST, PORT));
GenericObjectPool<StatefulRedisConnection> pool =
    ConnectionPoolSupport.createGenericObjectPool(() -> client.connect(), new
    GenericObjectPoolConfig());
pool.setMaxTotal(10); // Configure max connections to 10
try (StatefulRedisConnection connection = pool.borrowObject()) {
    RedisCommands syncCommands = connection.sync();
    syncCommands.set(key, value);
}
```

Cluster-Client-Erkennung und exponentielles Backoff (Valkey und Redis OSS)

Wenn Sie im aktivierten Clustermodus eine Verbindung zu einem ElastiCache Valkey- oder Redis-OSS-Cluster herstellen, muss die entsprechende Clientbibliothek clusterfähig sein. Die Clients müssen eine Zuordnung der Hash-Slots zu den entsprechenden Knoten im Cluster abrufen, um Anfragen an die richtigen Knoten zu senden und den zusätzlichen Leistungsaufwand bei der Bearbeitung von Cluster-Umleitungen zu vermeiden. Aus diesem Grund muss der Client in zwei verschiedenen Situationen eine vollständige Liste der Slots und der zugewiesenen Knoten erkennen:

- Der Client ist initialisiert und muss die anfängliche Slot-Konfiguration auffüllen.

- Eine MOVED-Umleitung wird vom Server empfangen, z. B. im Fall eines Failovers, bei dem alle vom ehemaligen Primärknoten bereitgestellten Slots vom Replikat übernommen werden, oder beim Re-Sharding, wenn Slots vom primären Quellknoten zum primären Zielknoten verschoben werden.

Die Client-Erkennung erfolgt normalerweise durch Ausgabe eines CLUSTER SLOT- oder CLUSTER NODE-Befehls an den Valkey- oder Redis OSS-Server. Wir empfehlen die CLUSTER SLOT-Methode, da sie das Set der Slot-Bereiche und die zugehörigen Primär- und Replikatknoten an den Client zurückgibt. Dies erfordert kein zusätzliches Parsen durch den Client und ist effizienter.

Abhängig von der Cluster-Topologie kann die Größe der Antwort auf den Befehl CLUSTER SLOT je nach Clustergröße variieren. Größere Cluster mit mehr Knoten erzeugen eine größere Antwort. Daher muss sichergestellt werden, dass die Anzahl der Clients, die die Cluster-Topologie ermitteln, nicht unbegrenzt zunimmt. Wenn beispielsweise die Client-Anwendung startet oder die Verbindung mit dem Server unterbrochen wird und die Cluster-Erkennung durchgeführt werden muss, besteht ein häufiger Fehler darin, dass die Client-Anwendung mehrere Wiederverbindungs- und Erkennungsanforderungen auslöst, ohne dass bei einem erneuten Versuch ein exponentielles Backoff erfolgt. Dies kann dazu führen, dass der Valkey- oder Redis OSS-Server bei einer CPU-Auslastung von 100% über einen längeren Zeitraum nicht reagiert. Der Ausfall verlängert sich, wenn jeder CLUSTER SLOT-Befehl eine große Anzahl von Knoten im Cluster-Bus verarbeiten muss. Aufgrund dieses Verhaltens haben wir in der Vergangenheit mehrere Client-Ausfälle in verschiedenen Sprachen beobachtet, darunter Python (redis-py-cluster) und Java (Lettuce und Redisson).

In einem Serverless-Cache werden viele der Probleme automatisch behoben, da die angekündigte Cluster-Topologie statisch ist und aus zwei Einträgen besteht: einem Schreib- und einem Lese-Endpunkt. Die Cluster-Erkennung wird außerdem automatisch auf mehrere Knoten verteilt, wenn der Cache-Endpunkt verwendet wird. Die folgenden Empfehlungen sind jedoch nach wie vor nützlich.

Wir empfehlen Folgendes, um die Auswirkungen eines plötzlichen Zustroms von Verbindungs- und Erkennungsanforderungen zu minimieren:

- Implementieren Sie einen Client-Verbindungspool mit einer begrenzten Größe, um die Anzahl der gleichzeitig eingehenden Verbindungen von der Client-Anwendung zu begrenzen.
- Wenn der Client aufgrund eines Timeouts die Verbindung mit dem Server unterbricht, versuchen Sie es erneut mit exponentiellem Backoff mit Jitter. Dadurch wird vermieden, dass mehrere Clients den Server gleichzeitig überlasten.
- Verwenden Sie die Anleitung unter [Verbindungsendpunkte finden in ElastiCache](#), um den Cluster-Endpunkt zur Durchführung der Cluster-Erkennung zu finden. Dadurch verteilen Sie die

Erkennungslast auf alle Knoten im Cluster (bis zu 90), anstatt einige fest codierte Seed-Knoten im Cluster zu erhalten.

Im Folgenden finden Sie einige Codebeispiele für die exponentielle Backoff-Wiederholungslogik in Redis-py, und Lettuce. PHPRedis

Beispiel 1 für Backoff-Logik: redis-py

Redis-py verfügt über einen integrierten Wiederholungsmechanismus, der unmittelbar nach einem Fehler einen erneuten Versuch vornimmt. [Dieser Mechanismus kann durch das `retry\_on\_timeout` Argument aktiviert werden, das beim Erstellen eines Redis-OSS-Objekts angegeben wird.](#) Hier demonstrieren wir einen benutzerdefinierten Wiederholungsmechanismus mit exponentiellem Backoff und Jitter. Wir haben eine Pull-Anfrage gesendet, um exponentielles Backoff in [redis-py \(#1494\)](#) nativ zu implementieren. Künftig ist eine manuelle Implementierung möglicherweise nicht mehr erforderlich.

```
def run_with_backoff(function, retries=5):
    base_backoff = 0.1 # base 100ms backoff
    max_backoff = 10 # sleep for maximum 10 seconds
    tries = 0
    while True:
        try:
            return function()
        except (ConnectionError, TimeoutError):
            if tries >= retries:
                raise
            backoff = min(max_backoff, base_backoff * (pow(2, tries) + random.random()))
            print(f"sleeping for {backoff:.2f}s")
            sleep(backoff)
            tries += 1
```

Sie können dann den folgenden Code verwenden, um einen Wert festzulegen:

```
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10))
res = run_with_backoff(lambda: client.set("key", "value"))
print(res)
```

Abhängig von Ihrem Workload möchten Sie möglicherweise den Basis-Backoff-Wert von 1 Sekunde auf einige Dutzende oder Hunderte von Millisekunden für latenzempfindliche Workloads ändern.

Beispiel 2 für Backoff-Logik: PHPRedis

PHPRedis verfügt über einen integrierten Wiederholungsmechanismus, der (nicht konfigurierbar) maximal zehnmal versucht. Es gibt eine konfigurierbare Verzögerung zwischen den Wiederholungsversuchen (mit einem Jitter ab dem zweiten Versuch). Weitere Informationen finden Sie im folgenden [Codebeispiel](#). [Wir haben eine Pull-Anfrage zur nativen Implementierung des exponentiellen Backoffs in PHPRedis \(#1986\) eingereicht, die inzwischen zusammengeführt und dokumentiert wurde](#). Für diejenigen, die die neueste Version von verwenden PHPRedis, ist eine manuelle Implementierung nicht erforderlich, aber wir haben hier die Referenz für die Versionen früherer Versionen aufgenommen. Im Folgenden finden Sie zunächst ein Codebeispiel, das die Verzögerung des Wiederholungsmechanismus konfiguriert:

```
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, $timeout, NULL, $retry_interval) != TRUE) {
    return; // ERROR: connection failed
}
$client->set($key, $value);
```

Beispiel 3 für Backoff-Logik: Lettuce

Lettuce verfügt über integrierte Wiederholungsmechanismen, die auf den exponentiellen Backoff-Strategien basieren, wie im Beitrag [Exponentielles Backoff und Jitter](#) beschrieben. Im Folgenden finden Sie einen Codeauszug, der den vollständigen Jitter-Ansatz zeigt:

```
public static void main(String[] args)
{
    ClientResources resources = null;
    RedisClient client = null;

    try {
        resources = DefaultClientResources.builder()
            .reconnectDelay(Delay.fullJitter(
                Duration.ofMillis(100),    // minimum 100 millisecond delay
                Duration.ofSeconds(5),     // maximum 5 second delay
                100, TimeUnit.MILLISECONDS) // 100 millisecond base
            ).build();

        client = RedisClient.create(resources, RedisURI.create(HOST, PORT));
        client.setOptions(ClientOptions.builder()
            .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
            100 millisecond connection timeout
        ).build());
    }
}
```

```
.timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(5)).build()) //  
5 second command timeout  
.build());  
  
    // use the connection pool from above example  
} finally {  
    if (connection != null) {  
        connection.close();  
    }  
  
    if (client != null){  
        client.shutdown();  
    }  
  
    if (resources != null){  
        resources.shutdown();  
    }  
  
}  
}
```

Konfigurieren Sie ein clientseitiges Timeout (Valkey und Redis OSS)

Konfiguration des clientseitigen Timeouts

Konfigurieren Sie das clientseitige Timeout so, dass der Server genügend Zeit hat, die Anfrage zu verarbeiten und die Antwort zu generieren. Dies ermöglicht einen Fail-Fast, wenn die Verbindung zum Server nicht hergestellt werden kann. Bestimmte Valkey- oder Redis OSS-Befehle können rechenintensiver sein als andere. Zum Beispiel Lua-Skripte oder MULTI/EXEC Transaktionen, die mehrere Befehle enthalten, die atomar ausgeführt werden müssen. Im Allgemeinen wird ein höheres clientseitiges Timeout empfohlen, um ein Timeout des Clients zu vermeiden, bevor die Antwort vom Server empfangen wird. Dies umfasst:

- Ausführen von Befehlen über mehrere Tasten
- Ausführen von MULTI/EXEC Transaktionen oder Lua-Skripten, die aus mehreren einzelnen Valkey- oder Redis-OSS-Befehlen bestehen
- Lesen von großen Werten
- Durchführen von Blockiervorgängen wie BLPOP

Im Falle eines Blockiervorgangs wie BLPOP empfiehlt es sich, das Befehls-Timeout auf einen Wert festzulegen, der unter dem Socket-Timeout liegt.

Im Folgenden finden Sie Codebeispiele für die Implementierung eines clientseitigen Timeouts in redis-py, und Lettuce. PHPRedis

Beispiel 1 für eine Timeout-Konfiguration: redis-py

Im Folgenden finden Sie ein Codebeispiel für redis-py:

```
# connect to Redis server with a 100 millisecond timeout
# give every Redis command a 2 second timeout
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10,socket_connect_timeout=0.1,socket_timeout=2))

res = client.set("key", "value") # will timeout after 2 seconds
print(res)                       # if there is a connection error

res = client.blpop("list", timeout=1) # will timeout after 1 second
                                     # less than the 2 second socket timeout
print(res)
```

Beispiel 2 für eine Timeout-Konfiguration: PHPRedis

Das Folgende ist ein Codebeispiel mit PHPRedis:

```
// connect to Redis server with a 100ms timeout
// give every Redis command a 2s timeout
$client = new Redis();
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, 0.1, NULL, 100, $read_timeout=2) != TRUE){
    return; // ERROR: connection failed
}
$client->set($key, $value);

$res = $client->set("key", "value"); // will timeout after 2 seconds
print "$res\n";                    // if there is a connection error

$res = $client->blpop("list", 1); // will timeout after 1 second
print "$res\n";                  // less than the 2 second socket timeout
```

Beispiel 3 für eine Timeout-Konfiguration: Lettuce

Im Folgenden finden Sie ein Codebeispiel für Lettuce:

```
// connect to Redis server and give every command a 2 second timeout
public static void main(String[] args)
{
    RedisClient client = null;
    StatefulRedisConnection<String, String> connection = null;
    try {
        client = RedisClient.create(RedisURI.create(HOST, PORT));
        client.setOptions(ClientOptions.builder()
            .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
            100 millisecond connection timeout
            .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(2)).build()) //
            2 second command timeout
            .build());

        // use the connection pool from above example

        commands.set("key", "value"); // will timeout after 2 seconds
        commands.blpop(1, "list"); // BLP0P with 1 second timeout
    } finally {
        if (connection != null) {
            connection.close();
        }

        if (client != null){
            client.shutdown();
        }
    }
}
```

Konfigurieren Sie ein serverseitiges Leerlauf-Timeout (Valkey und Redis OSS)

Wir haben Fälle beobachtet, in denen mit der Anwendung eines Kunden eine hohe Anzahl inaktiver Clients verbunden war, diese aber keine aktiven Befehle sendet. In solchen Szenarien können Sie alle 65 000 Verbindungen mit einer hohen Anzahl inaktiver Clients ausschöpfen. Zur Vermeidung solcher Szenarien konfigurieren Sie die Timeout-Einstellung auf dem Server entsprechend über [Valkey- und Redis OSS-Parameter](#). Dadurch wird sichergestellt, dass der Server die Verbindung inaktiver Clients aktiv trennt, um eine Erhöhung der Anzahl von Verbindungen zu vermeiden. Diese Einstellung ist für Serverless-Caches nicht verfügbar.

Lua-Skripte

Valkey und Redis OSS unterstützen mehr als 200 Befehle, einschließlich solcher zum Ausführen von Lua-Skripten. Bei Lua-Skripten gibt es jedoch mehrere Fallstricke, die sich auf den Arbeitsspeicher und die Verfügbarkeit von Valkey oder Redis OSS auswirken können.

Nicht parametrisierte Lua-Skripte

Jedes Lua-Skript wird auf dem Valkey- oder Redis-OSS-Server zwischengespeichert, bevor es ausgeführt wird. Unparametrisierte Lua-Skripten sind einzigartig, was dazu führen kann, dass der Valkey- oder Redis-OSS-Server eine große Anzahl von Lua-Skripten speichert und mehr Speicher verbraucht. Damit dies vermieden wird, stellen Sie sicher, dass alle Lua-Skripte parametrisiert sind, und führen Sie bei Bedarf regelmäßig `SCRIPT FLUSH` aus, um zwischengespeicherte Lua-Skripte zu bereinigen.

Beachten Sie auch, dass Schlüssel bereitgestellt werden müssen. Wenn kein Wert für den `KEY`-Parameter angegeben wird, schlägt das Skript fehl. Das wird zum Beispiel nicht funktionieren:

```
serverless-test-1st4hg.serverless.use1.cache.amazonaws.com:6379> eval 'return "Hello World"' 0
(error) ERR Lua scripts without any input keys are not supported.
```

Das wird funktionieren:

```
serverless-test-1st4hg.serverless.use1.cache.amazonaws.com:6379> eval 'return redis.call("get", KEYS[1])' 1 mykey-2
"myvalue-2"
```

Das folgende Codebeispiel zeigt, wie parametrisierte Skripte verwendet werden. Zunächst haben wir ein Beispiel für einen nicht parametrisierten Ansatz, der zu drei verschiedenen zwischengespeicherten Lua-Skripten führt. Dieser Ansatz wird nicht empfohlen:

```
eval "return redis.call('set','key1','1')" 0
eval "return redis.call('set','key2','2')" 0
eval "return redis.call('set','key3','3')" 0
```

Verwenden Sie stattdessen das folgende Muster, um ein einzelnes Skript zu erstellen, das übergebene Parameter akzeptieren kann:

```
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key1 1
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key2 2
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key3 3
```

Lang andauernde Lua-Skripte

Lua-Skripte können mehrere Befehle atomar ausführen, sodass die Ausführung länger dauern kann als bei einem normalen Valkey- oder Redis OSS-Befehl. Wenn das Lua-Skript nur schreibgeschützte Operationen ausführt, können Sie es zwischendurch beenden. Sobald das Lua-Skript jedoch einen Schreibvorgang ausführt, kann es nicht mehr beendet werden und muss vollständig ausgeführt werden. Ein mutierendes Lua-Skript mit langer Laufzeit kann dazu führen, dass der Valkey- oder Redis-OSS-Server lange Zeit nicht reagiert. Zur Behebung dieses Problems vermeiden Sie lang andauernde Lua-Skripte und testen Sie das Skript in einer Vorproduktionsumgebung.

Lua-Skript mit Stealth-Schreibvorgängen

Es gibt mehrere Möglichkeiten, wie ein Lua-Skript weiterhin neue Daten in Valkey oder Redis OSS schreiben kann, selbst wenn Valkey oder Redis OSS beendet ist: `maxmemory`

- Das Skript startet, wenn sich der Valkey- oder Redis-OSS-Server unten `maxmemory` befindet, und enthält mehrere Schreiboperationen
- Der erste Schreibbefehl des Skripts verbraucht keinen Speicher (wie `DEL`), gefolgt von weiteren Schreibvorgängen, die Speicher belegen
- Sie können dieses Problem beheben, indem Sie auf einem anderen Valkey- oder Redis-OSS-Server eine geeignete Räumungsrichtlinie konfigurieren als `noeviction`. Auf diese Weise kann Redis OSS Objekte entfernen und Speicherplatz zwischen Lua-Skripten freigeben.

Speichern großer zusammengesetzter Artikel (Valkey und Redis OSS)

In einigen Szenarien kann eine Anwendung große zusammengesetzte Elemente in Valkey oder Redis OSS speichern (z. B. einen Hash-Datensatz mit mehreren GB). Diese Vorgehensweise wird nicht empfohlen, da sie häufig zu Leistungsproblemen in Valkey oder Redis OSS führt. Der Client kann beispielsweise den Befehl `HGETALL` ausführen, um die gesamte Hash-Sammlung mit mehreren GB abzurufen. Dies kann zu erheblichem Speicherdruck auf den Valkey- oder Redis-OSS-Server führen, der das große Objekt im Client-Ausgabepuffer zwischenspeichert. Außerdem werden bei der Steckplatzmigration im Clustermodus ElastiCache keine Steckplätze migriert, die Elemente mit einer serialisierten Größe von mehr als 256 MB enthalten.

Zur Lösung der Probleme mit großen Elementen wird Folgendes empfohlen:

- Teilen Sie das große zusammengesetzte Element in mehrere kleinere Objekte auf. Teilen Sie beispielsweise eine große Hash-Sammlung in einzelne Schlüssel-Wert-Felder auf, wobei das Schlüsselnamensschema die Sammlung angemessen widerspiegelt, z. B. indem Sie ein gemeinsames Präfix im Schlüsselnamen verwenden, um die Sammlung von Elementen zu identifizieren. Wenn Sie atomar auf mehrere Felder in derselben Sammlung zugreifen müssen, können Sie den Befehl MGET verwenden, um mehrere Schlüsselwerte in demselben Befehl abzurufen.
- Wenn Sie alle Optionen geprüft haben und den großen Sammlungsdatensatz immer noch nicht aufteilen können, versuchen Sie, Befehle zu verwenden, die auf eine Teilmenge der Daten in der Sammlung statt auf die gesamte Sammlung angewendet werden. Vermeiden Sie einen Anwendungsfall, bei dem Sie die gesamte Sammlung mit mehreren GB mit demselben Befehl atomar abrufen müssen. Ein Beispiel ist die Verwendung von HGET- oder HMGET-Befehlen anstelle von HGETALL für Hash-Sammlungen.

Konfiguration des Lettuce-Clients (Valkey und Redis OSS)

In diesem Abschnitt werden die empfohlenen Konfigurationsoptionen für Java und Lettuce sowie deren Anwendung auf Cluster beschrieben. ElastiCache

Die Empfehlungen in diesem Abschnitt wurden mit Lettuce Version 6.2.2 getestet.

Themen

- [Beispiel: Lettuce-Konfiguration für den Clustermodus, TLS aktiviert](#)
- [Beispiel: Die Lettuce-Konfiguration für den Cluster-Modus ist deaktiviert, TLS aktiviert](#)

Java-DNS-Cache-TTLs

Die Java Virtual Machine (JVM) speichert DNS-Namensauflösungen zwischen. Wenn die JVM einen Hostnamen in eine IP-Adresse auflöst, speichert sie die IP-Adresse für einen bestimmten Zeitraum, der als (TTL) bezeichnet wird. time-to-live

Die Wahl des TTL-Werts ist ein Kompromiss zwischen Latenz auf der einen und Reaktionsfähigkeit auf Änderungen auf der anderen Seite. Bei einer kürzeren TTLs Variante erkennen DNS-Resolver Aktualisierungen im DNS des Clusters schneller. Dadurch kann Ihre Anwendung schneller auf Ersetzungen oder andere Workflows reagieren, denen Ihr Cluster unterzogen wird. Ist die TTL jedoch

zu niedrig, erhöht sich das Abfragevolumen, was die Latenz Ihrer Anwendung erhöhen kann. Zwar gibt es keinen korrekten TTL-Wert, ist es bei der Festlegung des TTL-Werts beachtenswert, zu berücksichtigen, wie lange Sie darauf warten können, dass eine Änderung wirksam wird.

Da ElastiCache Knoten DNS-Namenseinträge verwenden, die sich ändern können, empfehlen wir Ihnen, Ihre JVM mit einer niedrigen TTL von 5 bis 10 Sekunden zu konfigurieren. Auf diese Weise wird bei Änderung der IP-Adresse eines Knotens sichergestellt, dass Ihre Anwendung die neue IP-Adresse der Ressource durch erneute Abfrage des DNS-Eintrags abrufen und nutzen kann.

Bei einigen Java-Konfigurationen ist die JVM-Standard-TTL so festgelegt, dass DNS-Einträge nie aktualisiert werden, bis die JVM neu gestartet wird.

Einzelheiten zum Festlegen der JVM-TTL finden Sie unter [So legen Sie die JVM-TTL fest](#).

Lettuce-Version

Wir empfehlen die Verwendung der Lettuce-Version 6.2.2 oder höher.

Endpunkte

Wenn Sie Cluster verwenden, für die der Cluster-Modus aktiviert ist, legen Sie den `redisUri` auf den Endpunkt der Cluster-Konfiguration fest. Die DNS-Suche für diesen URI gibt eine Liste aller verfügbaren Knoten im Cluster zurück und wird während der Cluster-Initialisierung nach dem Zufallsprinzip auf einen davon aufgelöst. Weitere Informationen zur Funktionsweise der Topologieaktualisierung finden Sie weiter unten in diesem `dynamicRefreshResourcesThema`.

SocketOption

Aktivieren Sie [KeepAlive](#). Durch die Aktivierung dieser Option wird die Notwendigkeit verringert, während der Befehlslaufzeit ausgefallene Verbindungen zu behandeln.

Stellen Sie sicher, dass Sie das [Verbindungs-Timeout](#) entsprechend Ihren Anwendungsanforderungen und Ihrer Workload festlegen. Weitere Informationen finden Sie im Abschnitt „Timeouts“ weiter unten in diesem Thema.

ClusterClientOption: Client-Optionen mit aktiviertem Clustermodus

Aktiviert [AutoReconnect](#), wenn die Verbindung unterbrochen wird.

Set [CommandTimeout](#). Weitere Einzelheiten finden Sie im Abschnitt „Timeouts“ weiter unten in diesem Thema.

Legen Sie [nodeFilter](#) fest, um ausgefallene Knoten aus der Topologie herauszufiltern. Lettuce speichert alle Knoten, die sich in der Ausgabe „Cluster-Knoten“ befinden (einschließlich der Knoten mit PFAIL/FAIL-Status), in den „Partitionen“ des Clients (auch als Shards bekannt). Während der Erstellung der Cluster-Topologie wird versucht, eine Verbindung mit allen Knoten der Partition herzustellen. Dieses Verhalten von Lettuce, bei dem ausgefallene Knoten hinzugefügt werden, kann Verbindungsfehlern (oder Warnungen) verursachen, wenn Knoten aus einem beliebigen Grund ersetzt werden.

Wenn beispielsweise ein Failover abgeschlossen ist und der Cluster den Wiederherstellungsprozess startet, während das clusterTopologie-Element aktualisiert wird, weist die Zuordnung der Cluster-Busknoten einen kurzen Zeitraum auf, in dem der ausgefallene Knoten als FAIL-Knoten aufgeführt wird, bevor er vollständig aus der Topologie entfernt wird. Während dieser Zeit betrachtet der Lettuce-Client ihn als fehlerfreien Knoten und stellt kontinuierlich eine Verbindung zu ihm her. Dies führt zu einem Fehler, nachdem alle Wiederholungsversuche durchgeführt wurden.

Zum Beispiel:

```
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
        ... // other options
        .nodeFilter(it ->
            ! (it.is(RedisClusterNode.NodeFlag.FAIL)
            || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
            || it.is(RedisClusterNode.NodeFlag.HANDSHAKE)
            || it.is(RedisClusterNode.NodeFlag.NOADDR)))
        .validateClusterNodeMembership(false)
        .build();
redisClusterClient.setOptions(clusterClientOptions);
```

Note

Die Knotenfilterung wird am besten verwendet, wenn sie auf true DynamicRefreshSources gesetzt ist. Andernfalls wird der Primärknoten eines Shards herausgefiltert, wenn die Topologieansicht von einem einzelnen problematischen Seed-Knoten übernommen wird, der diesen primären Knoten als ausgefallen ansieht. Dies führt dazu, dass die Slots nicht abgedeckt werden. Wenn mehrere Seed-Knoten vorhanden sind (wenn wahr), verringert DynamicRefreshSources sich die Wahrscheinlichkeit, dass dieses Problem auftritt, da zumindest einige der Seed-Knoten nach einem Failover mit dem neu hochgestuften Primärknoten über eine aktualisierte Topologieansicht verfügen sollten.

ClusterTopologyRefreshOptions: Optionen zur Steuerung der Aktualisierung der Clustertopologie auf dem Client mit aktiviertem Clustermodus

 Note

Cluster mit deaktivierten Cluster-Modus unterstützen die Befehle zur Cluster-Ermittlung nicht und sind nicht mit allen Funktionen zur dynamischen Topologieermittlung des Clients kompatibel.

Der Clustermodus ist deaktiviert mit ElastiCache ist nicht mit dem von Lettuce kompatibel. MasterSlaveTopologyRefresh Stattdessen können Sie für einen deaktivierten Cluster-Modus einen StaticMasterReplicaTopologyProvider konfigurieren und die Lese- und Schreibendpunkte des Clusters bereitstellen.

Weitere Informationen zum Herstellen einer Verbindung mit Clustern mit deaktiviertem Cluster-Modus finden Sie unter [Finden Sie die Endpunkte eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

Wenn Sie die Funktion zur dynamischen Topologieermittlung von Lettuce verwenden möchten, können Sie einen Cluster mit aktiviertem Cluster-Modus mit derselben Shard-Konfiguration wie Ihr vorhandener Cluster erstellen. Für Cluster mit aktiviertem Cluster-Modus empfehlen wir jedoch, mindestens 3 Shards mit mindestens einem Replikat zu konfigurieren, um schnelles Failover zu unterstützen.

Aktivieren Sie [enablePeriodicRefresh](#). Dies ermöglicht eine regelmäßige Aktualisierung der Cluster-Topologie, sodass der Client die Cluster-Topologie in den Intervallen des refreshPeriod-Elements aktualisiert (Standard: 60 Sekunden). Wenn das Element deaktiviert ist, aktualisiert der Client die Cluster-Topologie nur, falls es bei dem Versuch, Befehle für den Cluster auszuführen, zu Fehlern kommt.

Wenn diese Option aktiviert ist, können Sie die mit der Aktualisierung der Cluster-Topologie verbundene Latenz reduzieren, indem Sie diesen Auftrag zu einer Hintergrundaufgabe hinzufügen. Die Aktualisierung der Topologie erfolgt zwar im Hintergrund, kann jedoch bei Clustern mit vielen Knoten etwas langsam sein. Dies liegt daran, dass alle Knoten nach ihren Ansichten abgefragt werden, um die aktuellste Cluster-Ansicht zu erhalten. Wenn Sie einen großen Cluster betreiben, sollten Sie den Zeitraum erhöhen.

Aktivieren Sie [enableAllAdaptiveRefreshTriggers](#). Dies ermöglicht eine adaptive Aktualisierung der Topologie, die alle [Trigger](#) verwendet: MOVED_REDIRECT, ASK_REDIRECT, PERSISTENT_RECONNECTS, UNCOVERED_SLOT, UNKNOWN_NODE. Adaptive

Aktualisierungsauslöser initiieren Aktualisierungen der Topologieansicht auf der Grundlage von Ereignissen, die während Valkey- oder Redis OSS-Clustervorgängen auftreten. Das Aktivieren dieser Option führt zu einer sofortigen Aktualisierung der Topologie, wenn einer der zuvor genannten Trigger auftritt. Die Rate der adaptiv ausgelöste Aktualisierungen ist durch einen Timeout begrenzt, da Ereignisse in großem Umfang auftreten können (Standard-Timeout zwischen Updates: 30).

Aktivieren Sie [closeStaleConnections](#). Dadurch können alte Verbindungen beim Aktualisieren der Cluster-Topologie geschlossen werden. [Sie tritt nur in Kraft, wenn. ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled \(\)](#) ist wahr. Wenn es aktiviert ist, kann der Client alte Verbindungen schließen und im Hintergrund neue erstellen. Dadurch wird die Notwendigkeit verringert, während der Befehlslaufzeit ausgefallene Verbindungen zu behandeln.

Aktivieren Sie [dynamicRefreshResources](#). Wir empfehlen, die `dynamicRefreshResources` Option für kleine Cluster zu aktivieren und für große Cluster zu deaktivieren. `dynamicRefreshResources` ermöglicht die Erkennung von Clusterknoten anhand des bereitgestellten Seed-Knotens (z. B. vom Cluster-Konfigurationsendpunkt). Es verwendet alle erkannten Knoten als Quellen für die Aktualisierung der Cluster-Topologie.

Bei der Verwendung der dynamischen Aktualisierung werden alle erkannten Knoten nach der Cluster-Topologie abgefragt und es wird versucht, die genaueste Cluster-Ansicht auszuwählen. Wenn der Wert auf „falsch“ festgelegt ist, werden nur die anfänglichen Seed-Knoten als Quellen für die Topologieerkennung verwendet, und die Anzahl der Clients wird nur für die ersten Seed-Knoten ermittelt. Ist er deaktiviert und der Endpunkt der Cluster-Konfiguration wird in einen ausgefallenen Knoten aufgelöst, schlägt der Versuch, die Cluster-Ansicht zu aktualisieren, fehl und führt zu Ausnahmen. Dieses Szenario kann eintreten, da es einige Zeit dauert, bis der Eintrag eines ausgefallenen Knotens vom Endpunkt der Cluster-Konfiguration entfernt wird. Daher kann der Konfigurationsendpunkt für einen kurzen Zeitraum immer noch nach dem Zufallsprinzip auf einen ausgefallenen Knoten aufgelöst werden.

Ist er aktiviert, verwenden wir jedoch alle Cluster-Knoten, die von der Cluster-Ansicht empfangen werden, um ihre aktuelle Ansicht abzufragen. Da wir ausgefallene Knoten aus dieser Ansicht herausfiltern, ist die Topologieaktualisierung erfolgreich. Wenn dies jedoch `dynamicRefreshSources` zutrifft, fragt Lettuce alle Knoten ab, um die Clusteransicht abzurufen, und vergleicht dann die Ergebnisse. Daher kann dieses Vorgehen für Cluster mit vielen Knoten teuer sein. Wir empfehlen, dass Sie diese Funktion für Cluster mit vielen Knoten deaktivieren.

```
final ClusterTopologyRefreshOptions topologyOptions =  
    ClusterTopologyRefreshOptions.builder()
```

```
.enableAllAdaptiveRefreshTriggers()  
.enablePeriodicRefresh()  
.dynamicRefreshSources(true)  
.build();
```

ClientResources

Konfigurieren Sie [DnsResolver](#) mit [DirContextDnsResolver](#). Der DNS-Resolver basiert auf Javas `com.sun.jndi.dns.DnsContextFactory`.

Konfigurieren Sie [reconnectDelay](#) mit exponentiellem Backoff und vollständigem Jitter. Lettuce verfügt über integrierte Wiederholungsmechanismen, die auf den exponentiellen Backoff-Strategien basieren. Einzelheiten finden Sie unter [Exponential Backoff and Jitter](#) im Architecture-Blog. AWS [Weitere Informationen zur Bedeutung einer Backoff-Strategie mit wiederholten Versuchen finden Sie in den Abschnitten zur Backoff-Logik im Blogbeitrag Best Practices im Datenbank-Blog.](#) AWS

```
ClientResources clientResources = DefaultClientResources.builder()  
.dnsResolver(new DirContextDnsResolver())  
.reconnectDelay(  
    Delay.fullJitter(  
        Duration.ofMillis(100),    // minimum 100 millisecond delay  
        Duration.ofSeconds(10),    // maximum 10 second delay  
        100, TimeUnit.MILLISECONDS)) // 100 millisecond base  
.build();
```

Timeouts

Verwenden Sie einen niedrigeren Wert für das Verbindungs-Timeout als für das Befehls-Timeout. Lettuce verwendet einen verzögerten Verbindungsaufbau. Wenn also das Verbindungs-Timeout höher als das Befehls-Timeout ist, kann es nach einer Topologieaktualisierung zu einem Zeitraum mit anhaltendem Ausfall kommen, falls Lettuce versucht, eine Verbindung mit einem fehlerhaften Knoten herzustellen, und das Befehls-Timeout immer überschritten wird.

Verwenden Sie ein dynamisches Befehls-Timeout für verschiedene Befehle. Wir empfehlen, das Befehls-Timeout auf der Grundlage der erwarteten Dauer des Befehls festzulegen. Verwenden Sie beispielsweise ein längeres Timeout für Befehle, die über mehrere Schlüssel hinweg iterieren, z. B. FLUSHDB-, FLUSHALL-, KEYS-, SMEMBERS- oder Lua-Skripts. Verwenden Sie kürzere Timeouts für einzelne Schlüsselbefehle wie SET, GET und HSET.

Note

Die im folgenden Beispiel konfigurierten Timeouts gelten für Tests, bei denen SET/GET-Befehle mit Schlüsseln und Werten von bis zu 20 Byte ausgeführt wurden. Die Verarbeitungszeit kann bei komplexen Befehlen oder größeren Schlüsseln und Werten länger sein. Sie sollten die Timeouts je nach Anwendungsfall Ihrer Anwendung festlegen.

```
private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);
```

```
SocketOptions socketOptions = SocketOptions.builder()
    .connectTimeout(CONNECT_TIMEOUT)
    .build();
```

```
class DynamicClusterTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
        ImmutableSet.<ProtocolKeyword>builder()
            .add(CommandType.FLUSHDB)
            .add(CommandType.FLUSHALL)
            .add(CommandType.CLUSTER)
            .add(CommandType.INFO)
            .add(CommandType.KEYS)
            .build();

    private final Duration defaultCommandTimeout;
    private final Duration metaCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
    }
}
```

```

        return defaultCommandTimeout.toMillis();
    }
}

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(
        new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
    .build();

```

Beispiel: Lettuce-Konfiguration für den Clustermodus, TLS aktiviert

Note

Timeouts im folgenden Beispiel beziehen sich auf Tests, bei denen SET/GET Befehle mit Schlüsseln und Werten ausgeführt wurden, die bis zu 20 Byte lang waren. Die Verarbeitungszeit kann bei komplexen Befehlen oder größeren Schlüsseln und Werten länger sein. Sie sollten die Timeouts je nach Anwendungsfall Ihrer Anwendung festlegen.

```

// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the cluster configuration endpoint
clusterConfigurationEndpoint = <cluster-configuration-endpoint> // TODO: add your
    cluster configuration endpoint
final RedisURI redisUriCluster =
    RedisURI.Builder.redis(clusterConfigurationEndpoint)
        .withPort(6379)
        .withSsl(true)
        .build();

// Configure the client's resources
ClientResources clientResources = DefaultClientResources.builder()

```

```

    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100),    // minimum 100 millisecond delay
            Duration.ofSeconds(10),    // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .dnsResolver(new DirContextDnsResolver())
    .build();

// Create a cluster client instance with the URI and resources
RedisClusterClient redisClusterClient =
    RedisClusterClient.create(clientResources, redisUriCluster);

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
class DynamicClusterTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
        ImmutableSet.<ProtocolKeyword>builder()
            .add(CommandType.FLUSHDB)
            .add(CommandType.FLUSHALL)
            .add(CommandType.CLUSTER)
            .add(CommandType.INFO)
            .add(CommandType.KEYS)
            .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()

```



```
.timeoutSource(new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT,
META_COMMAND_TIMEOUT))
    .build();

// Configure the topology refreshment options
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
        .enableAllAdaptiveRefreshTriggers()
        .enablePeriodicRefresh()
        .dynamicRefreshSources(true)
        .build();

// Configure the socket options
final SocketOptions socketOptions =
    SocketOptions.builder()
        .connectTimeout(CONNECT_TIMEOUT)
        .keepAlive(true)
        .build();

// Configure the client's options
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
        .topologyRefreshOptions(topologyOptions)
        .socketOptions(socketOptions)
        .autoReconnect(true)
        .timeoutOptions(timeoutOptions)
        .nodeFilter(it ->
            ! (it.is(RedisClusterNode.NodeFlag.FAIL)
                || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
                || it.is(RedisClusterNode.NodeFlag.NOADDR)))
        .validateClusterNodeMembership(false)
        .build();

redisClusterClient.setOptions(clusterClientOptions);

// Get a connection
final StatefulRedisClusterConnection<String, String> connection =
    redisClusterClient.connect();

// Get cluster sync/async commands
RedisAdvancedClusterCommands<String, String> sync = connection.sync();
RedisAdvancedClusterAsyncCommands<String, String> async = connection.async();
```

Beispiel: Die Lettuce-Konfiguration für den Cluster-Modus ist deaktiviert, TLS aktiviert

Note

Timeouts im folgenden Beispiel beziehen sich auf Tests, bei denen SET/GET Befehle mit Schlüsseln und Werten ausgeführt wurden, die bis zu 20 Byte lang waren. Die Verarbeitungszeit kann bei komplexen Befehlen oder größeren Schlüsseln und Werten länger sein. Sie sollten die Timeouts je nach Anwendungsfall Ihrer Anwendung festlegen.

```
// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the primary/reader endpoint
clusterEndpoint = <primary/reader-endpoint> // TODO: add your node endpoint
RedisURI redisUriStandalone =

    RedisURI.Builder.redis(clusterEndpoint).withPort(6379).withSsl(true).withDatabase(0).build();

ClientResources clientResources =
    DefaultClientResources.builder()
        .dnsResolver(new DirContextDnsResolver())
        .reconnectDelay(
            Delay.fullJitter(
                Duration.ofMillis(100), // minimum 100 millisecond delay
                Duration.ofSeconds(10), // maximum 10 second delay
                100,
                TimeUnit.MILLISECONDS)) // 100 millisecond base
        .build();

// Use a dynamic timeout for commands, to avoid timeouts during
// slow operations.
class DynamicTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
        ImmutableSet.<ProtocolKeyword>builder()
```

```

        .add(CommandType.FLUSHDB)
        .add(CommandType.FLUSHALL)
        .add(CommandType.INFO)
        .add(CommandType.KEYS)
        .build();

private final Duration metaCommandTimeout;
private final Duration defaultCommandTimeout;

DynamicTimeout(Duration defaultTimeout, Duration metaTimeout)
{
    defaultCommandTimeout = defaultTimeout;
    metaCommandTimeout = metaTimeout;
}

@Override
public long getTimeout(RedisCommand<?, ?, ?> command) {
    if (META_COMMAND_TYPES.contains(command.getType())) {
        return metaCommandTimeout.toMillis();
    }
    return defaultCommandTimeout.toMillis();
}
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
    .build();

final SocketOptions socketOptions =
    SocketOptions.builder().connectTimeout(CONNECT_TIMEOUT).keepAlive(true).build();

ClientOptions clientOptions =

    ClientOptions.builder().timeoutOptions(timeoutOptions).socketOptions(socketOptions).build();

RedisClient redisClient = RedisClient.create(clientResources, redisUriStandalone);
redisClient.setOptions(clientOptions);

```

Konfiguration eines bevorzugten Protokolls für Dual-Stack-Cluster (Valkey und Redis OSS)

Bei Valkey- oder Redis OSS-Clustern, die im Clustermodus aktiviert sind, können Sie mit dem IP Discovery-Parameter steuern, welches Protokoll Clients verwenden, um eine Verbindung zu den

Knoten im Cluster herzustellen. Der IP Discovery-Parameter kann entweder auf oder IPv4 gesetzt werden. IPv6

Für Valkey- oder Redis-OSS-Cluster legt der IP-Discovery-Parameter das IP-Protokoll fest, das in der Ausgabe von [Cluster-Slots \(\)](#), [Cluster-Shards \(\)](#) und [Clusterknoten \(\)](#) verwendet wird. Diese Befehle werden von Clients verwendet, um die Cluster-Topologie zu ermitteln. Clients verwenden die IPs in diesen Befehlen enthaltenen Befehle, um eine Verbindung zu den anderen Knoten im Cluster herzustellen.

Eine Änderung vom IP-Discovery führt zu keinen Ausfallzeiten für verbundene Clients. Die Verarbeitung und Weiterleitung der Änderungen wird jedoch einige Zeit dauern. Um festzustellen, wann sich die Änderungen für einen Valkey- oder Redis-OSS-Cluster vollständig verbreitet haben, überwachen Sie die Ausgabe von `cluster slots`. Sobald alle vom Befehl `Cluster-Slots` zurückgegebenen Knoten das neue Protokoll gemeldet haben, ist die Übertragung der Änderungen abgeschlossen.

Beispiel mit Redis-Py:

```
cluster = RedisCluster(host="xxxx", port=6379)
target_type = IPv6Address # Or IPv4Address if changing to IPv4

nodes = set()
while len(nodes) == 0 or not all((type(ip_address(host)) is target_type) for host in
    nodes):
    nodes = set()

    # This refreshes the cluster topology and will discovery any node updates.
    # Under the hood it calls cluster slots
    cluster.nodes_manager.initialize()
    for node in cluster.get_nodes():
        nodes.add(node.host)
    self.logger.info(nodes)

    time.sleep(1)
```

Beispiel mit Lettuce:

```
RedisClusterClient clusterClient = RedisClusterClient.create(RedisURI.create("xxxx",
    6379));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
    switching to IPv4
```

```
Set<String> nodes;

do {
    // Check for any changes in the cluster topology.
    // Under the hood this calls cluster slots
    clusterClient.refreshPartitions();
    Set<String> nodes = new HashSet<>();

    for (RedisClusterNode node : clusterClient.getPartitions().getPartitions()) {
        nodes.add(node.getUri().getHost());
    }

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
    } catch (UnknownHostException ignored) {}
    return false;
})));
```

Bewährte Methoden für Kunden (Memcached)

Erfahren Sie mehr über bewährte Methoden für gängige Szenarien mit ElastiCache Memcached-Clustern.

Themen

- [Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich \(Memcached\)](#)
- [Validierte Kunden mit Memcached](#)
- [Konfiguration eines bevorzugten Protokolls für Dual-Stack-Cluster \(Memcached\)](#)

Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich (Memcached)

Note

Dieser Abschnitt bezieht sich auf knotenbasierte Memcached-Cluster mit mehreren Knoten.

Um mehrere ElastiCache Memcached-Knoten effektiv nutzen zu können, müssen Sie in der Lage sein, Ihre Cache-Schlüssel auf die Knoten zu verteilen. Eine einfache Methode zum Lastenausgleich eines Clusters mit n Knoten besteht darin, den Hash des Objektschlüssels zu berechnen und das Ergebnis mit n zu modifizieren. $\text{hash}(\text{key}) \bmod n$ Der resultierende Wert (0 bis $n-1$) ist die Nummer des Knotens, auf dem Sie das Objekt platzieren.

Diese Methode ist einfach und funktioniert gut, solange die Anzahl der Knoten (n) konstant bleibt. Wenn Sie einen Knoten zum Cluster hinzufügen oder daraus entfernen, beträgt die Anzahl der zu verschiebenden Schlüssel jedoch jedes Mal $(n - 1) / n$ (dabei entspricht n der neuen Anzahl von Knoten). Diese Methode führt daher dazu, dass eine große Anzahl von Schlüsseln verschoben werden. Dies ist mit einer großen Anzahl anfänglicher Cache-Fehlelemente verbunden, insbesondere bei einer umfangreichen Anzahl von Knoten. Die Skalierung von 1 auf 2 Knoten führt bestenfalls dazu, dass $(2-1) / 2$ (50 %) der Schlüssel verschoben werden. Eine Skalierung von 9 auf 10 Knoten hat bestenfalls zur Folge, dass $(10-1) / 10$ (90 %) der Schlüssel verschoben werden. Ist die Aufwärtsskalierung durch einen sprunghaften Anstieg im Datenverkehr bedingt, ist eine große Anzahl von Cache-Fehlelementen nicht wünschenswert. Eine große Anzahl von Cache-Fehlelementen führt zu Anfragen bei der Datenbank, die aufgrund des sprunghaften Anstiegs im Datenverkehr bereits überlastet ist.

Die Lösung zu diesem Dilemma ist konsistentes Hashing. Beim konsistenten Hashing wird ein Algorithmus verwendet, bei dem jedes Mal, wenn ein Knoten zu einem Cluster hinzugefügt oder daraus entfernt wird, die Anzahl der zu verschiebenden Schlüssel ungefähr $1 / n$ beträgt (wobei n die neue Anzahl von Knoten ist). Eine Skalierung von 1 auf 2 Knoten hat schlimmstenfalls zur Folge, dass $1/2$ (50 Prozent) der Schlüssel verschoben werden. Eine Skalierung von 9 auf 10 Knoten hat zur Folge, dass $1/10$ (10 Prozent) der Schlüssel verschoben werden.

Sie als Benutzer steuern, welcher Hashalgorithmus für Mehrknoten-Cluster verwendet wird. Wir empfehlen, dass Sie Ihre Clients zur Verwendung von konsistentem Hashing konfigurieren. Glücklicherweise sind viele Memcached-Clientbibliotheken in den meisten gängigen Sprachen verfügbar, die konsistentes Hashing implementieren. Überprüfen Sie die Dokumentation der von Ihnen verwendeten Bibliothek darauf, ob sie konsistentes Hashing unterstützt und wie es implementiert wird.

Wenn Sie in Java, PHP oder .NET arbeiten, empfehlen wir Ihnen, eine der ElastiCache Amazon-Clientbibliotheken zu verwenden.

Konsistentes Hashing mithilfe von Java

Der ElastiCache Memcached Java-Client basiert auf dem Open-Source-SpyMemcached-Java-Client, der über integrierte konsistente Hashing-Funktionen verfügt. Die Bibliothek umfasst eine Klasse, die konsistentes Hashing implementiert. KetamaConnectionFactory Konsistentes Hashing ist in spyMemcached standardmäßig deaktiviert.

Weitere Informationen finden Sie in der KetamaConnectionFactory Dokumentation unter [KetamaConnectionFactory](#).

Konsistentes Hashing mithilfe von PHP mit Memcached

Der ElastiCache Memcached PHP-Client ist ein Wrapper rund um die integrierte Memcached-PHP-Bibliothek. Konsistentes Hashing wird von der Memcached-PHP-Bibliothek standardmäßig deaktiviert.

Verwenden Sie den folgenden Code, um konsistentes Hashing zu aktivieren.

```
$m = new Memcached();  
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

Zusätzlich zum obigen Code wird empfohlen, ebenfalls `memcached.sess_consistent_hash` in der Datei `php.ini` einzuschalten.

Weitere Informationen finden Sie in der Dokumentation zur Laufzeitkonfiguration für Memcached PHP unter <http://php.net/manual/en/memcached.configuration.php>. Beachten Sie insbesondere den Parameter `memcached.sess_consistent_hash`.

Konsistentes Hashing mit .NET mit Memcached

Der ElastiCache Memcached.NET-Client ist ein Wrapper für Enyim Memcached. Konsistentes Hashing wird vom Enyim-Memcached-Client standardmäßig aktiviert.

Weitere Informationen finden Sie in der Dokumentation unter [-Configuration#. memcached/locator](#) <https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-user-content-memcachedlocator>

Validierte Kunden mit Memcached

Die folgenden Clients wurden speziell daraufhin geprüft, ob sie mit allen unterstützten Netzwerktypkonfigurationen für Memcached funktionieren.

Validierte Clients:

- [AWS ElastiCache Cluster-Client Memcached für Php — Version *3.6.2](#)
- [AWS ElastiCache Cluster Client Memcached für Java](#) — Neuester Master auf Github

Konfiguration eines bevorzugten Protokolls für Dual-Stack-Cluster (Memcached)

Bei Memcached-Clustern können Sie mit dem IP-Discovery-Parameter das Protokoll steuern, das Clients für die Verbindung mit den Knoten im Cluster verwenden. Der IP Discovery-Parameter kann entweder auf `IPv4` oder `IPv6` gesetzt werden.

Der IP-Erkennungsparameter steuert das IP-Protokoll, das in der Cluster-Ausgabe der Konfiguration verwendet wird. Dies wiederum bestimmt das IP-Protokoll, das von Clients verwendet wird, die automatische Erkennung ElastiCache für Memcached-Cluster unterstützen.

Eine Änderung vom IP-Discovery führt zu keinen Ausfallzeiten für verbundene Clients. Die Weiterleitung der Änderungen wird jedoch einige Zeit dauern.

Überwachen Sie die Ausgabe von `getAvailableNodeEndpoints` für Java und für Php überwachen Sie die Ausgabe von `getServerList`. Sobald die Ausgabe dieser Funktionen Berichte IPs für alle Knoten im Cluster enthält, die das aktualisierte Protokoll verwenden, ist die Übertragung der Änderungen abgeschlossen.

Java-Beispiel:

```
MemcachedClient client = new MemcachedClient(new InetSocketAddress("xxxx", 11211));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
switching to IPv4

Set<String> nodes;

do {
    nodes =
        client.getAvailableNodeEndpoints().stream().map(NodeEndPoint::getIpAddress).collect(Collectors.toSet());

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByAddress(node));
    } catch (UnknownHostException ignored) {}
}));
```



```
        return false;
    }));
```

Php-Beispiel:

```
$client = new Memcached;
$client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);
$client->addServer("xxxx", 11211);

$nodes = [];
$target_ips_count = 0;
do {
    # The PHP memcached client only updates the server list if the polling interval has
    expired and a
    # command is sent
    $client->get('test');

    $nodes = $client->getServerList();

    sleep(1);
    $target_ips_count = 0;

    // For IPv4 use FILTER_FLAG_IPV4
    $target_ips_count = count(array_filter($nodes, function($node) { return
    filter_var($node["ipaddress"], FILTER_VALIDATE_IP, FILTER_FLAG_IPV6); }));
} while (count($nodes) !== $target_ips_count);
```

Alle vorhandenen Client-Verbindungen, die vor der Aktualisierung von IP Discovery erstellt wurden, werden weiterhin mit dem alten Protokoll verbunden. Alle validierten Clients stellen mithilfe des neuen IP-Protokolls automatisch wieder eine Verbindung mit dem Cluster her, sobald die Änderungen in der Ausgabe der Cluster-Erkennungsbefehle erkannt wurden. Dies hängt jedoch von der Implementierung des Clients ab.

TLS-fähige ElastiCache Dual-Stack-Cluster

Wenn TLS für ElastiCache Cluster aktiviert ist, geben die Cluster-Erkennungsfunktionen (`cluster slotscluster shards`, und `cluster nodes` für Redis) oder `config get cluster` für Memcached statt Hostnamen zurück. IPs Die Hostnamen werden dann verwendet, anstatt eine Verbindung IPs zum ElastiCache Cluster herzustellen und einen TLS-Handshake durchzuführen. Das bedeutet, dass Clients nicht vom IP-Discovery-Parameter betroffen sind. Bei TLS-fähigen Clustern

hat der IP Discovery-Parameter keine Auswirkung auf das bevorzugte IP-Protokoll. Stattdessen wird das verwendete IP-Protokoll dadurch bestimmt, welches IP-Protokoll der Client bei der Auflösung von DNS-Hostnamen bevorzugt.

Java-Clients

Wenn Sie eine Verbindung von einer Java-Umgebung aus herstellen, die IPv4 sowohl als auch unterstützt IPv6, wird Java aus IPv6 Gründen der Abwärtskompatibilität standardmäßig IPv4 den Vorzug geben. Die IP-Protokollpräferenz ist jedoch über die JVM-Argumente konfigurierbar. Zu bevorzugen IPv4, akzeptiert die JVM `-Djava.net.preferIPv4Stack=true` und bevorzugt IPv6 gesetzt. `-Djava.net.preferIPv6Stack=true` Die Einstellung `-Djava.net.preferIPv4Stack=true` bedeutet, dass die JVM keine Verbindungen mehr herstellt. IPv6 Für Valkey oder Redis OSS umfasst dies auch die Verbindungen zu anderen Nicht-Valkey- und Nicht-Redis-OSS-Anwendungen.

Einstellungen auf Host-Ebene

Wenn vom Client oder von der Client-Laufzeit keine Konfigurationsoptionen zum Festlegen einer IP-Protokollpräferenz bereitgestellt wird, hängt das IP-Protokoll bei der DNS-Auflösung im Allgemeinen von der Konfiguration des Hosts ab. Standardmäßig bevorzugen die meisten Hosts den Vorzug, IPv4 aber diese Einstellung kann IPv6 auf Host-Ebene konfiguriert werden. Dies wirkt sich auf alle DNS-Anfragen von diesem Host aus, nicht nur auf Anfragen an ElastiCache Cluster.

Linux-Hosts

Für Linux kann eine IP-Protokollpräferenz konfiguriert werden, indem die `gai.conf`-Datei geändert wird. Sie finden die `gai.conf`-Datei unter `/etc/gai.conf`. Wenn keine `gai.conf` angegeben ist, sollte ein Beispiel davon `/usr/share/doc/glibc-common-x.xx/gai.conf` verfügbar sein, das nach `/etc/gai.conf` kopiert werden kann. Die Standardkonfiguration sollte dann unkommentiert sein. Um die Konfiguration zu aktualisieren, die IPv4 bei der Verbindung zu einem ElastiCache Cluster bevorzugt wird, aktualisieren Sie die Priorität für den CIDR-Bereich, der den Cluster umfasst, so dass IPs sie über der Priorität für Standardverbindungen liegt. IPv6 Standardmäßig haben IPv6 Verbindungen eine Priorität von 40. Angenommen, der Cluster befindet sich in einem Subnetz mit dem CIDR `172.31.0.0/16`, würde die folgende Konfiguration dazu führen, dass Clients Verbindungen zu diesem Cluster bevorzugen. IPv4

```
label ::1/128      0
label ::/0         1
label 2002::/16    2
label ::/96        3
```

```

label ::ffff:0:0/96 4
label fec0::/10      5
label fc00::/7       6
label 2001:0::/32    7
label ::ffff:172.31.0.0/112 8
#
#   This default differs from the tables given in RFC 3484 by handling
#   (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
#   The reason for this difference is that these addresses are never
#   NATed while IPv4 site-local addresses most probably are. Given
#   the precedence of IPv6 over IPv4 (see below) on machines having only
#   site-local IPv4 and IPv6 addresses a lookup for a global address would
#   see the IPv6 be preferred. The result is a long delay because the
#   site-local IPv6 addresses cannot be used while the IPv4 address is
#   (at least for the foreseeable future) NATed. We also treat Teredo
#   tunnels special.
#
# precedence <mask>    <value>
#   Add another rule to the RFC 3484 precedence table. See section 2.1
#   and 10.3 in RFC 3484. The default is:
#
precedence ::1/128      50
precedence ::/0         40
precedence 2002::/16    30
precedence ::/96        20
precedence ::ffff:0:0/96 10
precedence ::ffff:172.31.0.0/112 100

```

[Weitere Informationen dazu finden Sie auf der Linux-Manpage `gai.conf`](#)

Windows-Hosts

Der Prozess für Windows-Hosts ist ähnlich. Für Windows-Hosts können Sie `netsh interface ipv6 set prefix CIDR_CONTAINING_CLUSTER_IPS PRECEDENCE LABEL` ausführen. Dies hat den gleichen Effekt wie das Ändern der `gai.conf`-Datei für Linux-Hosts.

Dadurch werden die Präferenzrichtlinien aktualisiert, sodass Verbindungen IPv4 IPv6 Verbindungen für den angegebenen CIDR-Bereich vorgezogen werden. Wenn beispielsweise angenommen wird, dass sich der Cluster in einem Subnetz befindet, in dem `172.31.0. 0:0 /16` CIDR ausgeführt wird, würde zu der folgenden Rangfolgetabelle führen, was dazu führen `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` würde, dass Clients eine Verbindung mit dem Cluster bevorzugen. IPv4

```
C:\Users\Administrator>netsh interface ipv6 show prefixpolicies
Querying active state...
```

```
Precedence Label Prefix
```

```
-----
```

```
100 15 ::ffff:172.31.0.0:0/112
```

```
20 4 ::ffff:0:0/96
```

```
50 0 ::1/128
```

```
40 1 ::/0
```

```
30 2 2002::/16
```

```
5 5 2001::/32
```

```
3 13 fc00::/7
```

```
1 11 fec0::/10
```

```
1 12 3ffe::/16
```

```
1 3 ::/96
```

Verwaltung von reserviertem Speicher für Valkey und Redis OSS

Reservierter Speicher ist Speicher, der nicht für Daten verwendet werden soll. Bei der Durchführung eines Backups oder Failovers verwenden Valkey und Redis OSS den verfügbaren Speicher, um Schreibvorgänge in Ihrem Cluster aufzuzeichnen, während die Daten des Clusters in die RDB-Datei geschrieben werden. Wenn nicht genügend Speicher für alle Schreibvorgängen zur Verfügung steht, tritt ein Fehler auf. Im Folgenden finden Sie Informationen zu Optionen für die Verwaltung von reserviertem Speicher ElastiCache für Redis OSS und zur Anwendung dieser Optionen.

Themen

- [Wie viel reservierten Speicher benötigen Sie?](#)
- [Parameter zum Veralten von reserviertem Speicher](#)
- [Angabe Ihres Parameters für die Verwaltung reservierten Arbeitsspeichers](#)

Wie viel reservierten Speicher benötigen Sie?

Wenn Sie eine Version von Redis OSS vor 2.8.22 ausführen, reservieren Sie mehr Speicher für Backups und Failover, als wenn Sie Redis OSS 2.8.22 oder höher ausführen. Diese Anforderung ist auf die unterschiedlichen Arten zurückzuführen, mit denen der Backup-Prozess ElastiCache für Redis OSS implementiert wird. Als Faustregel gilt, die Hälfte des `maxmemory` Werts eines Knotentyps für Redis OSS-Overhead für Versionen vor 2.8.22 und ein Viertel für Redis OSS-Versionen 2.8.22 und höher zu reservieren.

Aufgrund der unterschiedlichen ElastiCache Implementierungsmethoden des Sicherungs- und Replikationsprozesses lautet die Faustregel, 25% des Werts eines Knotentyps mithilfe des Parameters zu reservieren. `maxmemory reserved-memory-percent` Dies ist der Standardwert und wird in den meisten Fällen empfohlen.

Wenn die Typen `Burstable Micro` und `Small` Instances in der Nähe der `maxmemory` Grenzwerte arbeiten, kann es zu Swap-Nutzung kommen. Um die Betriebssicherheit dieser Instance-Typen bei Backups, Replikationen und hohem Datenaufkommen zu verbessern, empfehlen wir, den Wert des `reserved-memory-percent` Parameters bei kleinen Instance-Typen um bis zu 30% und bei `Micro`-Instance-Typen um bis zu 50% zu erhöhen.

Für schreibintensive Workloads auf ElastiCache Clustern mit Daten-Tiering empfehlen wir, den verfügbaren Speicher des Knotens auf `reserved-memory-percent` bis zu 50% zu erhöhen.

Weitere Informationen finden Sie hier:

- [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#)
- [So werden Synchronisation und Backup implementiert](#)
- [Datenzuweisung ElastiCache](#)

Parameter zum Veralten von reserviertem Speicher

Seit dem 16. März 2017 bietet Amazon zwei sich gegenseitig ausschließende Parameter für die Verwaltung Ihres Valkey- oder Redis-OSS-Speichers `undreserved-memory`. `reserved-memory-percent` Keiner dieser Parameter ist Teil der Valkey- oder Redis-OSS-Distribution.

Je nachdem, wann Sie ElastiCache Kunde wurden, ist der eine oder andere dieser Parameter der Standardparameter für die Speicherverwaltung. Dieser Parameter gilt, wenn Sie einen neuen Valkey- oder Redis OSS-Cluster oder eine neue Replikationsgruppe erstellen und eine Standardparametergruppe verwenden.

- Für Kunden, die vor dem 16. März 2017 damit angefangen haben: Wenn Sie einen Redis OSS-Cluster oder eine Redis-OSS-Cluster oder eine Redis-Replikationsgruppe mithilfe der Standardparametergruppe erstellen, lautet Ihr Speicherverwaltungsparameter `reserved-memory`. In diesem Fall werden null (0) Byte Speicher reserviert.
- Für Kunden, die am oder nach dem 16. März 2017 angefangen haben: Wenn Sie einen Valkey- oder Redis OSS-Cluster oder eine Replikationsgruppe mithilfe der Standardparametergruppe erstellen, lautet Ihr Speicherverwaltungsparameter `reserved-memory-percent`. In diesem Fall sind 25 Prozent des `maxmemory`-Wertes Ihres Knotens für Nichtdatenzwecke reserviert.

Nachdem Sie sich mit den beiden Speicherverwaltungsparametern Valkey oder Redis OSS vertraut gemacht haben, ziehen Sie es vielleicht vor, den zu verwenden, der nicht Ihr Standard ist oder nicht die Standardwerte enthält. Wenn dies der Fall ist, können Sie zu dem anderen Verwaltungsparameter für reservierten Speicher wechseln.

Um den Wert dieses Parameters zu ändern, können Sie eine benutzerdefinierte Parametergruppe erstellen und sie ändern, um Ihren bevorzugten Speicherverwaltungsparameter und -wert zu verwenden. Sie können die benutzerdefinierte Parametergruppe dann immer dann verwenden, wenn Sie einen neuen Valkey- oder Redis OSS-Cluster oder eine neue Replikationsgruppe erstellen. Existierende Cluster und Replikationsgruppen können Sie so ändern, dass sie die benutzerdefinierte Parametergruppe verwenden.

Weitere Informationen finden Sie hier:

- [Angabe Ihres Parameters für die Verwaltung reservierten Arbeitsspeichers](#)
- [Eine ElastiCache Parametergruppe erstellen](#)
- [Änderung einer ElastiCache Parametergruppe](#)
- [Einen ElastiCache Cluster ändern](#)
- [Ändern einer Replikationsgruppe](#)

Der Parameter "reserved-memory"

Vor dem 16. März 2017 wurde die gesamte Verwaltung des reservierten Speichers ElastiCache für Redis OSS mithilfe des Parameters durchgeführt. `reserved-memory` Der Standardwert von `reserved-memory` ist 0. Diese Standardeinstellung reserviert keinen Speicher für Valkey- oder Redis OSS-Overhead und ermöglicht es Valkey oder Redis OSS, den gesamten Speicher eines Knotens mit Daten zu verbrauchen.

Wenn Sie `reserved-memory` ändern, damit ausreichend Speicher für Sicherungen und Failovers zur Verfügung steht, müssen Sie eine benutzerdefinierte Parametergruppe erstellen. In dieser benutzerdefinierten Parametergruppe legen `reserved-memory` Sie einen Wert fest, der für die Valkey- oder Redis OSS-Version, die auf Ihrem Cluster ausgeführt wird, und für den Knotentyp des Clusters geeignet ist. Weitere Informationen finden Sie unter [Wie viel reservierten Speicher benötigen Sie?](#).

Der Parameter `reserved-memory` ist spezifisch für die allgemeine Redis OSS-Distribution ElastiCache und nicht Teil dieser.

Das folgende Verfahren zeigt, wie Sie reserved-memory den Speicher auf Ihrem Valkey- oder Redis OSS-Cluster verwalten.

So reservieren Sie Speicher mithilfe von reserviertem Speicher

1. Erstellen Sie eine benutzerdefinierte Parametergruppe, indem Sie die Parametergruppenfamilie angeben, die der von Ihnen ausgeführten Engine-Version entspricht, z. B. indem Sie die `redis2.8`-Parametergruppenfamilie angeben. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

```
aws elasticache create-cache-parameter-group \
  --cache-parameter-group-name redis6x-m3x1 \
```

```
--description "Redis OSS 2.8.x for m3.xlarge node type" \  
--cache-parameter-group-family redis6.x
```

2. Berechnen Sie, wie viele Byte Speicher für den OSS-Overhead von Valkey oder Redis reserviert werden müssen. Sie finden den Wert von maxmemory für Ihren Knotentyp unter [Redis OSS-Knotentyp-spezifische Parameter](#).
3. Ändern Sie die benutzerdefinierte Parametergruppe, sodass der Parameter reserved-memory der Anzahl von Bytes entspricht, die Sie im vorherigen Schritt berechnet haben. Im folgenden AWS CLI Beispiel wird davon ausgegangen, dass Sie eine Version von Redis OSS vor 2.8.22 ausführen und die Hälfte der Knoten reservieren müssen. maxmemory Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis28-m3x1 \  
  --parameter-name-values "ParameterName=reserved-memory,  
  ParameterValue=7130316800"
```

Sie benötigen für jeden verwendeten Knotentyp eine eigene benutzerdefinierte Parametergruppe, da jeder Knotentyp einen anderen maxmemory-Wert hat. Daher benötigt jeder Knotentyp einen anderen Wert für reserved-memory.

4. Ändern Sie Ihren Redis OSS-Cluster oder Ihre Redis-Replikationsgruppe so, dass sie Ihre benutzerdefinierte Parametergruppe verwendet.

Mit dem folgenden CLI-Beispiel wird der Cluster `my-redis-cluster` geändert, damit er ab sofort die benutzerdefinierte Parametergruppe `redis28-m3x1` verwendet. Weitere Informationen finden Sie unter [Einen ElastiCache Cluster ändern](#).

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cluster \  
  --cache-parameter-group-name redis28-m3x1 \  
  --apply-immediately
```

Mit dem folgenden CLI-Beispiel wird die Replikationsgruppe `my-redis-repl-grp` geändert, damit sie ab sofort die benutzerdefinierte Parametergruppe `redis28-m3x1` verwendet. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-repl-grp \  
  --cache-parameter-group-name redis28-m3x1 \  
  --apply-immediately
```



```
--apply-immediately
```

Der Parameter reserved-memory-percent

Am 16. März 2017 ElastiCache führte Amazon den Parameter ein `reserved-memory-percent` und stellte ihn in allen Versionen von ElastiCache for Redis OSS zur Verfügung. Der Zweck von `reserved-memory-percent` besteht darin, die Verwaltung des reservierten Speichers für alle Ihre Cluster zu vereinfachen. Dies wird dadurch erreicht, dass Sie eine einzige Parametergruppe für jede Parametergruppenfamilie (wie `redis2.8`) zur Verfügung haben, um den reservierten Speicher Ihrer Cluster unabhängig vom Knotentyp zu verwalten. Der Standardwert für `reserved-memory-percent` ist 25 (25 Prozent).

Der Parameter `reserved-memory-percent` ist spezifisch für die allgemeine Redis OSS-Distribution ElastiCache und nicht Teil davon.

Wenn Ihr Cluster einen Knotentyp aus der R6gd-Familie verwendet und Ihre Speicherauslastung 75 Prozent erreicht, wird Daten-Tiering automatisch ausgelöst. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

Um Speicher zu reservieren mit reserved-memory-percent

Gehen Sie wie folgt vor `reserved-memory-percent`, um den Speicher auf Ihrem ElastiCache for Redis OSS-Cluster zu verwalten:

- Wenn Sie Redis OSS 2.8.22 oder höher ausführen, weisen Sie Ihrem Cluster die Standardparametergruppe zu. Der Standardwert 25 Prozent sollte ausreichen. Falls nicht, führen Sie die folgenden Schritte aus, um den Wert zu ändern.
- Wenn Sie eine Version von Redis OSS vor 2.8.22 ausführen, müssen Sie wahrscheinlich mehr Speicher als die standardmäßigen 25 Prozent reservieren. `reserved-memory-percent` Führen Sie dazu die folgenden Schritte aus.

Um den Prozentwert von zu ändern reserved-memory-percent

1. Erstellen Sie eine benutzerdefinierte Parametergruppe, indem Sie die Parametergruppenfamilie angeben, die der von Ihnen ausgeführten Engine-Version entspricht, z. B. indem Sie die `redis2.8`-Parametergruppenfamilie angeben. Eine benutzerdefinierte Parametergruppe ist erforderlich, da Sie eine Standardparametergruppe nicht ändern können. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name redis28-50 \  
  --description "Redis OSS 2.8.x 50% reserved" \  
  --cache-parameter-group-family redis2.8
```

Da `reserved-memory-percent` Speicher als Prozentanteil von `maxmemory` eines Knotens reserviert, benötigen Sie keine benutzerdefinierte Parametergruppe für jeden Knotentyp.

2. Ändern Sie die benutzerdefinierte Parametergruppe, sodass `reserved-memory-percent` 50 (50 Prozent) beträgt. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis28-50 \  
  --parameter-name-values "ParameterName=reserved-memory-percent,  
  ParameterValue=50"
```

3. Verwenden Sie diese benutzerdefinierte Parametergruppe für alle Redis OSS-Cluster oder Replikationsgruppen, auf denen eine Version von Redis OSS ausgeführt wird, die älter als 2.8.22 ist.

Das folgende CLI-Beispiel ändert den Redis OSS-Cluster `somy-redis-cluster`, dass er `redis28-50` ab sofort die benutzerdefinierte Parametergruppe verwendet. Weitere Informationen finden Sie unter [Einen ElastiCache Cluster ändern](#).

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cluster \  
  --cache-parameter-group-name redis28-50 \  
  --apply-immediately
```

Das folgende CLI-Beispiel ändert die Redis OSS-Replikationsgruppe `somy-redis-repl-grp`, dass sie `redis28-50` ab sofort die benutzerdefinierte Parametergruppe verwendet. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-repl-grp \  
  --cache-parameter-group-name redis28-50 \  
  --apply-immediately
```

Angabe Ihres Parameters für die Verwaltung reservierten Arbeitsspeichers

Wenn Sie am 16. März 2017 ein aktueller ElastiCache Kunde waren, lautet Ihr Standardparameter für die Verwaltung von reserviertem Speicher `reserved-memory` mit null (0) Byte reserviertem Speicher. Wenn Sie nach dem 16. März 2017 ElastiCache Kunde wurden, ist `reserved-memory-percent` Ihr Standardparameter für die Verwaltung von reserviertem Speicher so, dass 25 Prozent des Speichers des Knotens reserviert sind. Dies gilt unabhängig davon, wann Sie Ihren OSS-Cluster oder Ihre Replikationsgruppe ElastiCache für Redis erstellt haben. Sie können Ihren Parameter für die Verwaltung des reservierten Speichers jedoch mithilfe der ElastiCache API AWS CLI oder ändern.

Die Parameter `reserved-memory` und `reserved-memory-percent` schließen sich gegenseitig aus. Eine Parametergruppe verfügt immer über einen dieser Parameter, niemals jedoch über beide. Sie können den Parameter ändern, den eine Parametergruppe für die Verwaltung von reserviertem Speicher verwendet, indem Sie die Parametergruppe ändern. Die Parametergruppe muss eine benutzerdefinierte Parametergruppe sein, da Sie die Standardparametergruppen nicht ändern können. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

Um zu spezifizieren `reserved-memory-percent`

Um `reserved-memory-percent` als Verwaltungsparameter für reservierten Speicher zu verwenden, ändern Sie eine benutzerdefinierte Parametergruppe mit dem `modify-cache-parameter-group`-Befehl. Verwenden Sie den `parameter-name-values`-Parameter zum Angeben von `reserved-memory-percent` sowie einen Wert dafür.

Mit dem folgenden CLI-Beispiel wird die benutzerdefinierte Parametergruppe `redis32-cluster-on` geändert, sodass sie `reserved-memory-percent` zum Verwalten von reserviertem Speicher verwendet. Für die Parametergruppe muss `ParameterValue` ein Wert zugewiesen werden, um den `ParameterName`-Parameter für die Verwaltung des reservierten Speichers zu verwenden. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis32-cluster-on \  
  --parameter-name-values "ParameterName=reserved-memory-percent, ParameterValue=25"
```

So geben Sie `reserved-memory` an

Um `reserved-memory` als Verwaltungsparameter für reservierten Speicher zu verwenden, ändern Sie eine benutzerdefinierte Parametergruppe mit dem `modify-cache-parameter-group`-Befehl.

Verwenden Sie den `parameter-name-values`-Parameter zum Angeben von `reserved-memory` sowie einen Wert dafür.

Mit dem folgenden CLI-Beispiel wird die benutzerdefinierte Parametergruppe `redis32-m3x1` geändert, sodass sie `reserved-memory` zum Verwalten von reserviertem Speicher verwendet. Für die Parametergruppe muss `ParameterValue` ein Wert zugewiesen werden, um den `ParameterName`-Parameter für die Verwaltung des reservierten Speichers zu verwenden. Da die Engine-Version höher als 2.8.22 ist, wird der Wert auf `3565158400` gesetzt. Dieser Wert entspricht 25 % des Werts von `maxmemory` für `cache.m3.xlarge`. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis32-m3x1 \  
  --parameter-name-values "ParameterName=reserved-memory, ParameterValue=3565158400"
```

Bewährte Methoden bei der Arbeit mit nodenbasierten Valkey- und Redis OSS-Clustern

Multi-AZ-Nutzung, ausreichender Arbeitsspeicher, Clustergrößenänderung und Minimierung von Ausfallzeiten sind alles nützliche Konzepte, die Sie bei der Arbeit mit knotenbasierten Clustern in Valkey oder Redis OSS berücksichtigen sollten. Wir empfehlen Ihnen, sich mit diesen bewährten Methode vertraut zu machen und sie zu befolgen.

Themen

- [Minimierung von Ausfallzeiten mit Multi-AZ](#)
- [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#)
- [Online-Größenanpassung von Clustern](#)
- [Minimieren der Ausfallzeit während der Wartung](#)

Minimierung von Ausfallzeiten mit Multi-AZ

Es gibt eine Reihe von Fällen, in denen ElastiCache Valkey oder Redis OSS möglicherweise einen Primärknoten ersetzen müssen. Dazu gehören bestimmte Arten von geplanten Wartungsarbeiten und der unwahrscheinliche Fall eines Ausfalls eines Primärknotens oder einer Availability Zone.

Dieser Austausch führt zu einer gewissen Ausfallzeit für den Cluster, aber wenn Multi-AZ aktiviert ist, wird die Ausfallzeit minimiert. Die Rolle des primären Knotens wird automatisch auf eines der Read Replicas übertragen. Es ist nicht erforderlich, einen neuen Primärknoten zu erstellen und bereitzustellen, da dies transparent ElastiCache gehandhabt wird. Dieser Failover und die Replikatheraufstufung stellen sicher, dass Sie weiter in den neuen primären Knoten schreiben können, sobald die Heraufstufung abgeschlossen wurde.

Weitere Informationen [Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS](#) über Multi-AZ und die Minimierung von Ausfallzeiten finden Sie unter.

Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen

Snapshots und Synchronisationen in Valkey 7.2 und höher sowie Redis OSS Version 2.8.22 und höher

Valkey unterstützt standardmäßig Snapshots und Synchronisationen. Redis OSS 2.8.22 führt einen forkless-Speicherprozess ein, der es Ihnen ermöglicht, einen größeren Teil Ihres Speichers für die Nutzung durch Ihre Anwendung zuzuweisen, ohne dass es bei Synchronisationen und Speichervorgängen zu einer erhöhten Swap-Nutzung kommt. Weitere Informationen finden Sie unter [So werden Synchronisation und Backup implementiert](#).

Redis OSS-Snapshots und Synchronisationen vor Version 2.8.22

Wenn Sie mit ElastiCache for Redis OSS arbeiten, ruft Redis OSS in einer Reihe von Fällen einen Schreibbefehl im Hintergrund auf:

- Beim Erstellen eines Snapshots für eine Sicherung
- Beim Synchronisieren von Replikaten mit dem primären Cluster in einer Replikationsgruppe
- Wenn Sie die Funktion „Nur Dateien anhängen“ (AOF) für Redis OSS aktivieren.
- Beim Hochstufen eines Replikats zu einem primären (was eine primäre/Replikat-Synchronisierung verursacht).

Immer wenn Redis OSS einen Schreibvorgang im Hintergrund ausführt, müssen Sie über ausreichend verfügbaren Speicher verfügen, um den Prozessaufwand zu bewältigen. Wenn nicht genügend Arbeitsspeicher verfügbar ist, schlägt der Vorgang fehl. Aus diesem Grund ist es wichtig, bei der Erstellung Ihres Redis OSS-Clusters einen Knoteninstanztyp auszuwählen, der über ausreichend Speicher verfügt.

Schreibvorgang im Hintergrund und Speichernutzung mit Valkey und Redis OSS

Immer wenn ein Schreibvorgang im Hintergrund aufgerufen wird, forken Valkey und Redis OSS seinen Prozess ab (denken Sie daran, dass diese Engines Single-Threading verwenden). Ein Fork speichert Ihre Daten auf der Festplatte in einer Redis OSS .rdb-Snapshot-Datei. Die andere Vergabelung führt alle Lese- und Schreibvorgänge durch. Um sicherzustellen, dass es sich bei Ihrem point-in-time Snapshot um einen Snapshot handelt, werden alle Datenaktualisierungen und Ergänzungen in einen vom Datenbereich getrennten Bereich des verfügbaren Speichers geschrieben.

Solange genügend Arbeitsspeicher zum Aufzeichnen aller Schreibvorgänge verfügbar ist, während die Daten dauerhaft auf dem Datenträger erhalten bleiben, treten keine Probleme aufgrund von Speichermangel auf. Wenn vermehrt Probleme aufgrund von Speichermangel auftreten, treffen beliebige der folgenden Situationen ein:

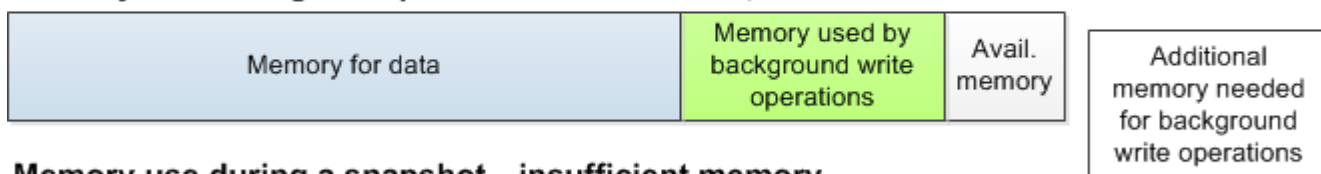
- Ihre Anwendung führt viele Schreibvorgänge aus und benötigt daher eine große Menge an verfügbarem Arbeitsspeicher zum Akzeptieren neuer oder aktualisierter Daten.
- Es ist sehr wenig Arbeitsspeicher zum Schreiben neuer oder aktualisierter Daten verfügbar.
- Die dauerhafte Erhaltung eines großen Datensatzes auf der Festplatte dauert eine lange Zeit und erfordert eine große Anzahl von Schreibvorgängen.

Das folgende Diagramm veranschaulicht die Speichernutzung beim Ausführen eines Hintergrundschreibvorgangs.

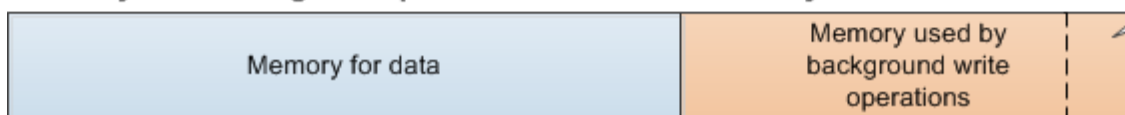
Memory use prior to a snapshot



Memory use during a snapshot—sufficient memory



Memory use during a snapshot—insufficient memory



Informationen zu den Auswirkungen einer Sicherung auf die Leistung finden Sie unter [Auswirkungen von Backups knotenbasierter Cluster auf die Leistung](#).

Weitere Informationen darüber, wie Valkey und Redis OSS Snapshots durchführen, finden Sie unter <http://valkey.io>.

Weitere Informationen zu Regionen und Availability Zones finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

Vermeidung von Speichermangel beim Ausführen eines Hintergrundschreibvorgangs

Immer wenn ein Schreibvorgang im Hintergrund wie BGSAVE oder aufgerufen BGREWRITEAOF wird, muss mehr Speicher verfügbar sein, als durch Schreibvorgänge während des Vorgangs verbraucht wird, um zu verhindern, dass der Prozess fehlschlägt. Im schlimmsten Fall wird während des Schreibvorgangs im Hintergrund jeder Datensatz aktualisiert und einige neue Datensätze werden dem Cache hinzugefügt. Aus diesem Grund empfehlen wir, den Wert `reserved-memory-percent` auf 50 (50 Prozent) für Redis OSS-Versionen vor 2.8.22 oder 25 (25 Prozent) für Valkey und alle Redis OSS-Versionen 2.8.22 und höher festzulegen.

Der Wert `maxmemory` gibt den für die Daten und den Betriebsaufwand verfügbaren Arbeitsspeicher an. Da der Parameter `reserved-memory` in der Standardparametergruppe nicht geändert werden kann, müssen Sie eine benutzerdefinierte Parametergruppe für den Cluster erstellen. Der Standardwert für `reserved-memory` ist 0, was es Redis OSS ermöglicht, den gesamten maximalen Speicher mit Daten zu verbrauchen, sodass möglicherweise zu wenig Speicher für andere Zwecke übrig bleibt, z. B. für einen Schreibvorgang im Hintergrund. Informationen zu `maxmemory`-Werten nach Knoten-Instance-Typ finden Sie unter [Redis OSS-Knotentyp-spezifische Parameter](#).

Sie können `reserved-memory` Parameter auch verwenden, um den Speicherverbrauch auf der Box zu reduzieren.

Weitere Informationen zu Valkey- und Redis-spezifischen Parametern finden Sie unter [ElastiCache Valkey- und Redis OSS-Parameter](#)

Weitere Informationen zum Erstellen und Ändern von Parametergruppen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#) und [Änderung einer ElastiCache Parametergruppe](#).

Online-Größenanpassung von Clustern

Resharding umfasst das Hinzufügen und Entfernen von Shards oder Knoten für den Cluster sowie die Neuverteilung von Schlüsselräumen. Daher haben viele Aspekte Einfluss auf die Resharding-

Operation, z. B. Workload des Clusters, Speichernutzung und allgemeine Datengröße. Für optimale Ergebnisse empfehlen wir, dass Sie die allgemeinen bewährten Methoden zu Clustern für eine gleichmäßige Verteilung von Workload-Verteilung befolgen. Außerdem empfehlen wir, die folgenden Schritte durchzuführen.

Vor dem Beginn des Resharding sollten Sie Folgendes durchführen:

- Testen Sie Ihre Anwendung – Testen Sie das Verhalten Ihrer Anwendung während des Reshardings nach Möglichkeit in einer Staging-Umgebung.
- Erhalten Sie frühzeitige Benachrichtigungen bei Skalierungsproblemen – Resharding ist ein rechenintensiver Vorgang. Aus diesem Grund empfehlen wir, beim Resharding die CPU-Auslastung bei Multicore-Instances unter 80 Prozent und bei Single-Core-Instances unter 50 Prozent zu halten. Überwachen Sie ElastiCache die OSS-Metriken von Redis und initiieren Sie das Resharding, bevor Ihre Anwendung Skalierungsprobleme beobachtet. Die Überwachung folgender Metriken ist nützlich: `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `NewConnections`, `FreeableMemory`, `SwapUsage` und `BytesUsedForCacheItems`.
- Stellen Sie vor dem Hochskalieren sicher, dass ausreichend freier Speicher verfügbar ist – Stellen Sie beim Hochskalieren sicher, dass der freie Speicher auf den beizubehaltenden Shards mindestens das 1,5-fache des Arbeitsspeichers beträgt, der auf den Shards verwendet wird, die Sie entfernen möchten.
- Initiieren Sie Resharding außerhalb der Spitzenzeiten – Diese Vorgehensweise hilft, die Auswirkungen auf die Latenz und den Durchsatz auf den Client während des Resharding-Vorgangs zu reduzieren. Außerdem wird das Resharding schneller abgeschlossen, da bei der Slot-Verteilung mehr Ressourcen verwendet werden können.
- Überprüfen Sie das Client-Timeout-Verhalten – Einige Clients stellen möglicherweise eine höhere Latenz während der Online-Cluster-Größenänderung fest. Es kann helfen, bei Ihrer Client-Bibliothek einen höheren Timeout zu konfigurieren, da dem System so Zeit zur Verbindungsherstellung unter höheren Lastbedingungen auf dem Server gegeben wird. Manchmal wird eine große Anzahl an Verbindungen zum Server geöffnet. Fügen Sie in diesen Fällen exponentielles Backoff hinzu, um Logik erneut zu verbinden. Hierdurch wird verhindert, dass ein Schub neuer Verbindungen den Server gleichzeitig erreicht.
- Laden Sie Ihre Funktionen auf jeden Shard — Beim Skalieren Ihres Clusters ElastiCache werden die Funktionen, die in einem der vorhandenen Knoten geladen wurden (zufällig ausgewählt), automatisch auf die neuen Knoten repliziert. Wenn Ihr Cluster über Valkey 7.2 und höher oder Redis OSS 7.0 oder höher verfügt und Ihre Anwendung [Functions](#) verwendet, empfehlen wir,

alle Ihre Funktionen vor dem Skalieren auf alle Shards zu laden, damit Ihr Cluster nicht mit unterschiedlichen Funktionen auf verschiedenen Shards endet.

Beachten Sie nach dem Resharding Folgendes:

- Die Skalierung nach oben ist möglicherweise nur zum Teil erfolgreich, wenn auf den Ziel-Shards nicht ausreichend Arbeitsspeicher verfügbar ist. In diesem Fall prüfen Sie den verfügbaren Speicher und wiederholen Sie ggf. die Operation. Die Daten auf den Ziel-Shards werden nicht gelöscht.
- Die Befehle FLUSHALL und FLUSHDB werden in Lua-Skripten während eines Resharding-Vorgangs nicht unterstützt. Vor Redis OSS 6 wurde der BRPOPLPUSH Befehl nicht unterstützt, wenn er auf dem zu migrierenden Steckplatz ausgeführt wird.

Minimieren der Ausfallzeit während der Wartung

Die Konfiguration des Cluster-Modus ist am besten im Rahmen verwalteter oder nicht verwalteter Operationen verfügbar. Es wird empfohlen, einen im Cluster-Modus unterstützten Client zu verwenden, der eine Verbindung mit dem Clusterermittlungsendpunkt herstellt. Bei deaktiviertem Cluster-Modus empfehlen wir, den primären Endpunkt für alle Schreiboperationen zu verwenden.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpunkt werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren. Bei deaktiviertem Cluster-Modus empfehlen wir daher, den Reader-Endpunkt für Leseaktivitäten zu verwenden.

Wenn im Cluster aktiviert AutoFailover ist, kann sich der primäre Knoten ändern. Daher sollte die Anwendung die Rolle des Knotens bestätigen und alle Leseendpunkte aktualisieren. Dadurch wird sichergestellt, dass Sie keine große Belastung des primären Knotens verursachen. Bei AutoFailover deaktivierter Option ändert sich die Rolle des Knotens nicht. Die Ausfallzeiten bei verwalteten oder nicht verwalteten Vorgängen sind jedoch höher als bei Clustern mit AutoFailover aktivierter Option.

Vermeiden Sie es, Leseanforderungen an einen einzelnen Read-Replica-Knoten weiterzuleiten, da dessen Nichtverfügbarkeit zu einem Leseausfall führen könnte. Entweder Sie greifen auf das Lesen vom Primärknoten zurück oder stellen sicher, dass Sie über mindestens zwei Read Replicas verfügen, um Leseunterbrechungen während der Wartung zu vermeiden.

Caching-Strategien für Memcached

Im folgenden Thema finden Sie Strategien zum Auffüllen und Verwalten Ihres Memcached-Caches.

Welche Strategien Sie zum Auffüllen und Verwalten Ihres Cache implementieren müssen, hängt von den zwischengespeicherten Daten und den Zugriffsmustern auf diese Daten ab. Zum Beispiel möchten Sie wahrscheinlich nicht dieselbe Strategie für eine Top-10-Bestenliste auf einer Spieleseite und für trendige Nachrichten verwenden. Im Rest dieses Abschnitts besprechen wir gängige Cache-Wartungsstrategien und ihre Vor- und Nachteile.

Themen

- [Read Replicas](#)
- [Lazy Loading](#)
- [Write-Through](#)
- [Hinzufügen von TTL](#)
- [Verwandte Themen](#)

Read Replicas

Sie können die Leistung ElastiCache serverloser Caches häufig erheblich verbessern, indem Sie Replikate erstellen und aus ihnen lesen, anstatt sie vom primären Cache-Knoten aus zu lesen. Weitere Informationen finden Sie unter [Bewährte Methoden für die Verwendung von Read Replicas](#).

Lazy Loading

Wie der Name schon sagt, ist Lazy Loading ("langsames Laden") eine Caching-Strategie, die Daten nur bei Bedarf in den Cache lädt. Es funktioniert folgendermaßen.

Amazon ElastiCache ist ein speicherinterner Schlüsselwertspeicher, der sich zwischen Ihrer Anwendung und dem Datenspeicher (Datenbank) befindet, auf den sie zugreift. Immer wenn Ihre Anwendung Daten anfordert, sendet sie die Anfrage zuerst an den Cache. ElastiCache Wenn die Daten im Cache vorhanden und aktuell sind, gibt ElastiCache die Daten an Ihre Anwendung zurück. Wenn die Daten nicht im Cache vorhanden sind oder abgelaufen sind, fordert Ihre Anwendung die Daten von Ihrem Datenspeicher an. Ihr Datenspeicher gibt die Daten dann an Ihre Anwendung zurück. Als nächstes schreibt Ihre Anwendung die vom Speicher empfangenen Daten in den Cache. Auf diese Weise können sie bei der nächsten Anforderung schneller abgerufen werden.

Ein Cache-Treffer tritt auf, wenn sich Daten im Cache befinden und nicht abgelaufen sind:

1. Die Anwendung fordert Daten aus dem Cache an.
2. Der Cache gibt die Daten an die Anwendung zurück.

Ein Cache-Fehlertreffer tritt auf, wenn sich Daten nicht im Cache befinden oder abgelaufen sind:

1. Die Anwendung fordert Daten vom Cache an.
2. Der Cache enthält die angeforderten Daten nicht und gibt daher `null` zurück.
3. Ihre Anwendung fordert die Daten an und erhält sie aus der Datenbank.
4. Ihre Anwendung aktualisiert den Cache mit den neuen Daten.

Vor- und Nachteile von Lazy Loading

Die Vorteile von Lazy Loading sind:

- Es werden nur die angeforderten Daten im Cache abgelegt.

Da die meisten Daten nie angefordert werden, wird durch Lazy Loading vermieden, dass der Cache mit nicht angeforderten Daten gefüllt wird.

- Knotenfehler sind für Ihre Anwendung nicht fatal.

Wenn ein Knoten ausfällt und durch einen neuen, leeren Knoten ersetzt wird, funktioniert Ihre Anwendung weiterhin, allerdings mit erhöhter Latenz. Wenn Anforderungen an den neuen Knoten gestellt werden, führt jeder Cache-Fehlertreffer zu einer Abfrage der Datenbank. Gleichzeitig wird die Datenkopie dem Cache hinzugefügt, so dass nachfolgende Anforderungen aus dem Cache abgerufen werden.

Die Nachteile von Lazy Loading sind:

- Bei Cache-Fehlschlägen gibt es Verzögerungen. Jeder Cache-Fehlertreffer führt zu drei Übertragungsvorgängen:
 1. Anfängliche Anforderung von Daten aus dem Cache
 2. Abfrage der Daten aus der Datenbank
 3. Schreiben der Daten in den Cache

Diese Fehlertreffer können zu einer merklichen Verzögerung beim Eingang der Daten in die Anwendung führen.

- Veraltete Daten.

Wenn Daten nur bei einem Cache-Fehlertreffer in den Cache geschrieben werden, können Daten im Cache veraltet sein. Dieses Ergebnis tritt auf, weil der Cache nicht aktualisiert wird, wenn Daten in der Datenbank geändert werden. Um dieses Problem zu beheben, können Sie die [Write-Through](#) und [Hinzufügen von TTL](#)-Strategien anwenden.

Beispiel für Lazy Loading Pseudocode

Im Folgenden finden Sie ein Pseudocode-Beispiel für Lazy Loading-Logik.

```
// *****
// function that returns a customer's record.
// Attempts to retrieve the record from the cache.
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
//   retrieved from the database,
//   added to the cache, and
//   returned to the application
// *****
get_customer(customer_id)

    customer_record = cache.get(customer_id)
    if (customer_record == null)

        customer_record = db.query("SELECT * FROM Customers WHERE id = {0}",
customer_id)
        cache.set(customer_id, customer_record)

    return customer_record
```

In diesem Beispiel sieht der Anwendungscode, der die Daten abrufen, wie folgt aus.

```
customer_record = get_customer(12345)
```

Write-Through

Die Write-Through-Strategie fügt dem Cache Daten hinzu oder aktualisiert Daten im Cache, wenn Daten in die Datenbank geschrieben werden.

Vor- und Nachteile von Write-Through

Die Vorteile von Write-Through sind:

- Die Daten im Cache sind nie veraltet.

Da die Daten im Cache jedes Mal aktualisiert werden, wenn sie in die Datenbank geschrieben werden, sind die Daten im Cache immer aktuell.

- Schreibstrafe im Vergleich zu Lesestrafe.

Jeder Schreibvorgang umfasst zwei Übertragungsvorgänge:

1. Ein Schreibvorgang in den Cache
2. Ein Schreibvorgang in die Datenbank

Dadurch wird die Latenz des Prozesses erhöht. Endbenutzer tolerieren im Allgemeinen Latenz beim Aktualisieren von Daten eher als beim Abrufen. Aktualisierungen werden als arbeits- und zeitintensiver wahrgenommen.

Die Nachteile von Write-Through sind:

- Fehlende Daten.

Wenn Sie einen neuen Knoten hochfahren, sei es aufgrund eines Knotenfehlers oder einer horizontalen Skalierung, fehlen Daten. Diese Daten fehlen weiterhin, bis sie in der Datenbank hinzugefügt oder aktualisiert werden. Sie können dies minimieren, indem Sie [Lazy Loading](#) mit Write-Through implementieren.

- Cache-Änderung.

Die meisten Daten werden nie gelesen, was eine Verschwendung von Ressourcen darstellt. Durch [Hinzufügen eines Time-to-Live-Werts \(TTL\)](#) können Sie verschwendeten Speicherplatz minimieren.

Beispiel für Write-Through-Pseudocode

Das Folgende ist ein Pseudocode-Beispiel für Write-Through-Logik.

```
// *****  
// function that saves a customer's record.  
// *****  
save_customer(customer_id, values)
```

```
customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)
cache.set(customer_id, customer_record)
return success
```

In diesem Beispiel sieht der Anwendungscode, der die Daten abrufen, wie folgt aus.

```
save_customer(12345, {"address": "123 Main"})
```

Hinzufügen von TTL

Lazy Loading ermöglicht veraltete Daten, schlägt jedoch nicht wegen leeren Knoten fehl. Write-Through stellt sicher, dass die Daten immer frisch sind, kann aber bei leeren Knoten fehlschlagen und den Cache mit überflüssigen Daten füllen. Indem Sie jedem Schreibvorgang einen Time-to-Live-Wert (TTL) hinzufügen, können Sie die Vorteile jeder Strategie nutzen. Gleichzeitig können Sie das Überladen des Caches mit zusätzlichen Daten weitgehend vermeiden.

Time to Live (TTL) ist ein ganzzahliger Wert, der die Anzahl der Sekunden angibt, bis der Schlüssel abläuft. Valkey oder Redis OSS können Sekunden oder Millisekunden für diesen Wert angeben. Memcached gibt diesen Wert in Sekunden an. Wenn eine Anwendung versucht, einen abgelaufenen Schlüssel zu lesen, gilt der Schlüssel als nicht gefunden. Die Datenbank wird nach dem Schlüssel abgefragt und der Cache aktualisiert. Dieser Ansatz garantiert nicht, dass ein Wert nicht veraltet ist. Er verhindert jedoch, dass die Daten zu veraltet werden, und erfordert, dass Werte im Cache gelegentlich aus der Datenbank aktualisiert werden.

[Weitere Informationen finden Sie unter den OSS-Befehlen Valkey und Redis oder den Memcached-Befehlen. `set`](#)

Beispiele für TTL Pseudocode

Das Folgende ist ein Pseudocode-Beispiel für Write-Through-Logik mit TTL.

```
// *****
// function that saves a customer's record.
// The TTL value of 300 means that the record expires
//   300 seconds (5 minutes) after the set command
//   and future reads will have to query the database.
// *****
save_customer(customer_id, values)

customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)
```

```
cache.set(customer_id, customer_record, 300)

return success
```

Das Folgende ist ein Pseudocode-Beispiel für Lazy-Loading-Logik mit TTL.

```
// *****
// function that returns a customer's record.
// Attempts to retrieve the record from the cache.
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
//   retrieved from the database,
//   added to the cache, and
//   returned to the application.
// The TTL value of 300 means that the record expires
//   300 seconds (5 minutes) after the set command
//   and subsequent reads will have to query the database.
// *****
get_customer(customer_id)

    customer_record = cache.get(customer_id)

    if (customer_record != null)
        if (customer_record.TTL < 300)
            return customer_record          // return the record and exit function

    // do this only if the record did not exist in the cache OR
    //   the TTL was >= 300, i.e., the record in the cache had expired.
    customer_record = db.query("SELECT * FROM Customers WHERE id = {0}", customer_id)
    cache.set(customer_id, customer_record, 300) // update the cache
    return customer_record                    // return the newly retrieved record and exit
function
```

In diesem Beispiel sieht der Anwendungscode, der die Daten abrufen, wie folgt aus.

```
save_customer(12345, {"address": "123 Main"})
```

```
customer_record = get_customer(12345)
```

Verwandte Themen

- [In-Memory-Datastore](#)

- [Auswählen einer Engine und einer Version](#)
- [Skalierung ElastiCache](#)

Verwaltung Ihres knotenbasierten Clusters in ElastiCache

ElastiCache bietet zwei Bereitstellungsoptionen: serverlose Caches und knotenbasierte Cluster. Jeder hat seine eigenen Fähigkeiten und Anforderungen.

Dieser Abschnitt enthält Themen, die Sie bei der Verwaltung Ihrer knotenbasierten Cluster unterstützen.

Note

Diese Themen gelten nicht für Serverless. ElastiCache

Themen

- [Auto Scaling Valkey- und Redis OSS-Cluster](#)
- [Ändern des Clustermodus](#)
- [AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#)
- [Hohe Verfügbarkeit mit Replikationsgruppen](#)
- [Verwaltung der ElastiCache Cluster-Wartung](#)
- [Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen](#)

Auto Scaling Valkey- und Redis OSS-Cluster

Voraussetzungen

ElastiCache Auto Scaling ist auf Folgendes beschränkt:

- Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert), auf denen Valkey 7.2 oder Redis OSS 6.0 oder höher ausgeführt wird
- Cluster mit Datenklassierung (Clustermodus aktiviert), auf denen Valkey 7.2 oder Redis OSS 7.0.7 oder höher ausgeführt wird
- Instanzgrößen: Groß, 2 XLarge XLarge

- Instance-Familien – R7g, R6g, R6gd, R5, M7g, M6g, M5, C7gn
- Auto Scaling in ElastiCache wird nicht für Cluster unterstützt, die in globalen Datenspeichern, Outposts oder Local Zones ausgeführt werden.

Automatisches Kapazitätsmanagement mit ElastiCache Auto Scaling mit Valkey oder Redis OSS

ElastiCache Auto Scaling mit Valkey oder Redis OSS ist die Möglichkeit, die gewünschten Shards oder Replikate in Ihrem Service automatisch zu erhöhen oder zu verringern. ElastiCache nutzt den Application Auto Scaling Scaling-Dienst, um diese Funktionalität bereitzustellen. Weitere Informationen finden Sie unter [Application Auto Scaling](#). Um die automatische Skalierung zu verwenden, definieren und wenden Sie eine Skalierungsrichtlinie an, die von Ihnen CloudWatch zugewiesene Metriken und Zielwerte verwendet. ElastiCache Auto Scaling verwendet die Richtlinie, um die Anzahl der Instanzen als Reaktion auf tatsächliche Workloads zu erhöhen oder zu verringern.

Sie können die verwenden AWS-Managementkonsole, um eine Skalierungsrichtlinie anzuwenden, die auf einer vordefinierten Metrik basiert. Eine `predefined metric` ist in einer Aufzählung definiert, sodass Sie sie im Code durch einen Namen angeben oder in der AWS-Managementkonsole verwenden können. Benutzerdefinierte Metriken können nicht über die AWS-Managementkonsole ausgewählt werden. Alternativ können Sie entweder die AWS CLI oder die Application Auto Scaling-API verwenden, um eine Skalierungsrichtlinie anzuwenden, die auf einer vordefinierten oder benutzerdefinierten Metrik basiert.

ElastiCache für Valkey und Redis unterstützt OSS die Skalierung für die folgenden Dimensionen:

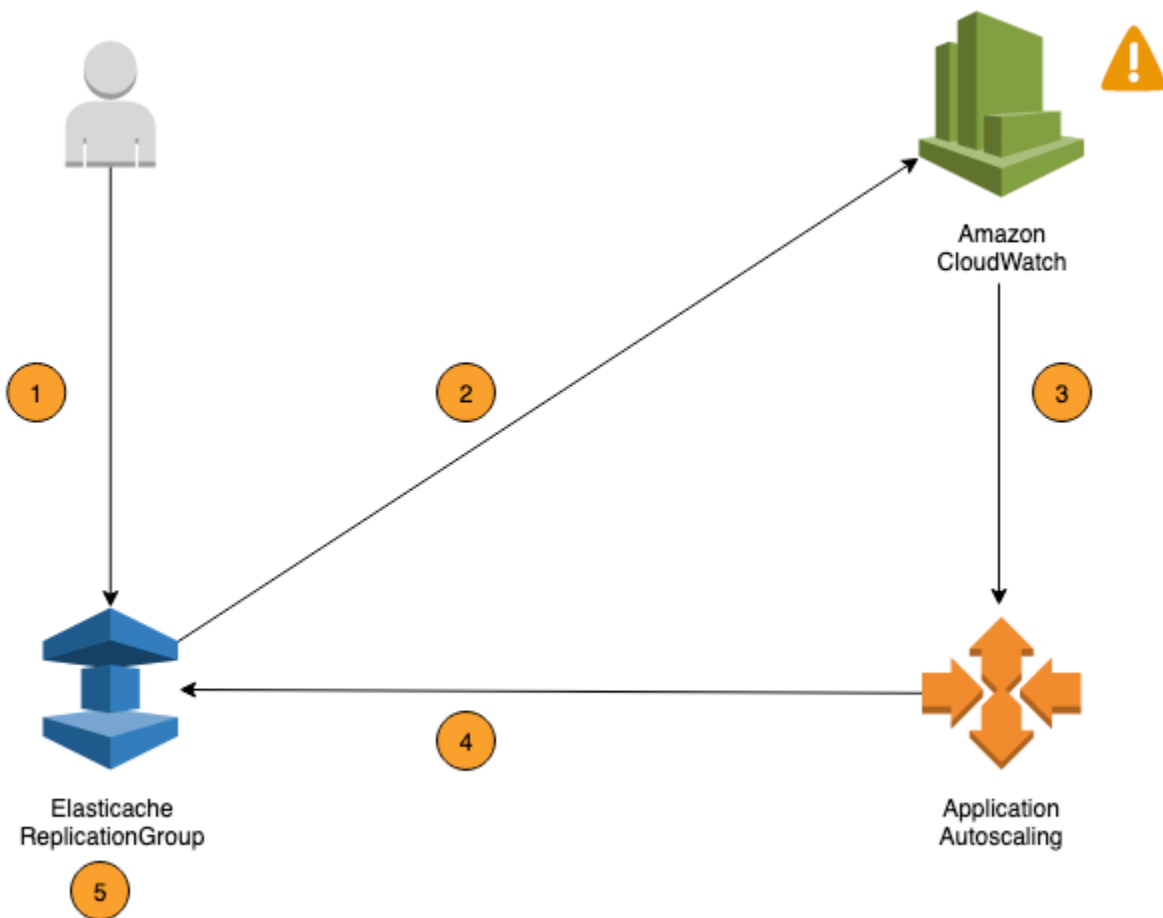
- Shards— Automatisches Hinzufügen/Entfernen von Shards im Cluster ähnlich wie beim manuellen Online-Resharding. In diesem Fall löst ElastiCache Auto Scaling die Skalierung in Ihrem Namen aus.
- Replikate — Automatisch und add/remove replicas in the cluster similar to manual Increase/Decrease replica operations. ElastiCache auto scaling for Valkey and Redis OSS adds/removes einheitlich auf allen Shards im Cluster repliziert.

ElastiCache für Valkey und Redis unterstützt OSS die folgenden Arten von Richtlinien für die automatische Skalierung:

- [Skalierungsrichtlinien für die Ziel-Nachverfolgung](#)— Erhöhen oder verringern Sie die Anzahl shards/replicas , mit der Ihr Service ausgeführt wird, basierend auf einem Zielwert für eine

bestimmte Metrik. Dies ähnelt der Art und Weise, wie ein Thermostat die Temperatur in Ihrem Zuhause konstant hält. Sie wählen eine Temperatur aus und der Thermostat erledigt den Rest.

- [Geplante Skalierung für Ihre Anwendung](#). — ElastiCache Für Valkey und Redis kann OSS Auto Scaling die Anzahl der ausgeführten shards/replicas Dienste je nach Datum und Uhrzeit erhöhen oder verringern.



Die folgenden Schritte fassen den auto Skalierungsprozess ElastiCache für Valkey und Redis OSS zusammen, wie im vorherigen Diagramm dargestellt:

1. Sie erstellen eine ElastiCache Auto Scaling-Richtlinie für Ihre Replikationsgruppe.
2. ElastiCache Auto Scaling erstellt in Ihrem Namen ein Paar CloudWatch Alarme. Jedes Paar stellt die Ober- und Untergrenze für Metriken dar. Diese CloudWatch Alarme werden ausgelöst, wenn

die tatsächliche Auslastung des Clusters über einen längeren Zeitraum von Ihrer Zielauslastung abweicht. Sie können jetzt -Alarme in der -Konsole anzeigen.

3. Wenn der konfigurierte Metrikwert Ihre Zielauslastung für einen bestimmten Zeitraum überschreitet (oder unter das Ziel fällt), wird ein Alarm CloudWatch ausgelöst, der Auto Scaling zur Bewertung Ihrer Skalierungsrichtlinie auslöst.
4. ElastiCache Auto Scaling gibt eine Modifizierungsanforderung aus, um Ihre Clusterkapazität anzupassen.
5. ElastiCache verarbeitet die Modifizierungsanforderung und erhöht (oder verringert) die Shards/Replicas Clusterkapazität dynamisch, sodass sie sich Ihrer Zielauslastung annähert.

Um zu verstehen, wie ElastiCache Auto Scaling funktioniert, nehmen wir an, Sie haben einen Cluster mit dem Namen `UsersCluster`. Durch die Überwachung der CloudWatch Metriken bestimmen Sie die maximale Anzahl an Shards, die der Cluster benötigt, wenn der Verkehr seinen Höhepunkt erreicht, und die Mindestanzahl an Shards, wenn der Verkehr am niedrigsten Punkt ist. `UsersCluster` Sie legen auch einen Zielwert für die CPU-Auslastung für den `UsersCluster` Cluster fest. ElastiCache Auto Scaling verwendet seinen Target-Tracking-Algorithmus, um sicherzustellen, dass die bereitgestellten Shards von `UsersCluster` nach Bedarf angepasst werden, sodass die Auslastung auf oder nahe dem Zielwert bleibt.

Note

Die Skalierung kann viel Zeit in Anspruch nehmen und erfordert zusätzliche Cluster-Ressourcen, damit die Shards wieder ausgeglichen werden können. ElastiCache Auto Scaling ändert die Ressourceneinstellungen nur, wenn die tatsächliche Arbeitslast über einen längeren Zeitraum von mehreren Minuten erhöht (oder reduziert) bleibt. Der Auto Scaling Target-Tracking-Algorithmus versucht, die Zielauslastung langfristig auf oder nahe dem von Ihnen gewählten Wert zu halten.

Auto-Scaling-Richtlinien

Eine Skalierungsrichtlinie enthält die folgenden Komponenten:

- Eine Zielmetrik — Die CloudWatch Metrik, die ElastiCache für Valkey und Redis OSS Auto Scaling verwendet, um zu bestimmen, wann und wie stark skaliert werden soll.
- Minimale und maximale Kapazität — Die minimale und die maximale Anzahl an Shards oder Replikaten, die für die Skalierung verwendet werden sollen.

⚠ Important

Wenn bei der Erstellung der Auto Scaling-Richtlinie die aktuelle Kapazität höher als die konfigurierte maximale Kapazität ist, skalieren wir MaxCapacity bei der Richtlinienerstellung entsprechend. Wenn die aktuelle Kapazität unter der konfigurierten Mindestkapazität liegt, skalieren wir ebenfalls auf die. MinCapacity

- Eine Cooldown-Phase: Die Zeitspanne (in Sekunden) zwischen dem Ende einer Scale-In- bzw. einer Scale-Out-Aktivität und dem Beginn einer neuen Scale-Out-Aktivität.
- Eine dienstverknüpfte Rolle — Eine AWS Identity and Access Management (IAM) - Rolle, die mit einem bestimmten AWS Dienst verknüpft ist. Eine dienstbezogene Rolle umfasst alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen. ElastiCache Auto Scaling generiert diese Rolle automatisch für Sie. `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`
- Scale-In-Aktivitäten aktivieren oder deaktivieren: Die Möglichkeit, Scale-In-Aktivitäten für eine Richtlinie zu aktivieren oder zu deaktivieren.

Themen

- [Zielmetrik für Auto Scaling](#)
- [Minimale und maximale Kapazität](#)
- [Abkühlungszeit](#)
- [Aktivieren oder Deaktivieren von Scale-In-Aktivitäten](#)

Zielmetrik für Auto Scaling

Bei diesem Richtlinientyp werden eine vordefinierte oder benutzerdefinierte Metrik und ein Zielwert für die Metrik in einer Skalierungsrichtlinienkonfiguration zur Zielverfolgung angegeben. ElastiCache für Valkey und Redis OSS Auto Scaling erstellt und verwaltet CloudWatch Alarme, die die Skalierungsrichtlinie auslösen, und berechnet die Skalierungsanpassung auf der Grundlage der Metrik und des Zielwerts. Die Skalierungsrichtlinie fügt nach shards/replicas Bedarf hinzu oder entfernt sie, um die Metrik auf oder nahe dem angegebenen Zielwert zu halten. Abgesehen davon, dass eine Skalierungsrichtlinie für die Ziel-Nachverfolgung die Metrik nahe an dem Zielwert hält, passt sie sich auch an die Schwankungen in der Metrik aufgrund eines sich ändernden Workloads an. Eine solche Richtlinie minimiert auch schnelle Schwankungen bei der Anzahl der shards/replicas für Ihren Cluster verfügbaren Geräte.

Nehmen wir zum Beispiel eine Skalierungsrichtlinie, die die vordefinierte durchschnittliche `ElastiCachePrimaryEngineCPUUtilization`-Metrik verwendet. Eine solche Richtlinie kann die CPU-Auslastung bei einem bestimmten Prozentsatz der Auslastung halten, z. B. bei 70 Prozent.

 Note

Für jeden DB-Cluster können Sie nur eine Auto-Scaling-Richtlinie für jede Zielmetrik erstellen.

Minimale und maximale Kapazität

Shards

Sie können die maximale Anzahl von Shards angeben, auf die skaliert werden kann, indem Sie Valkey und Redis ElastiCache OSS Auto Scaling verwenden. Dieser Wert muss kleiner oder gleich 250 sein und mindestens 1 sein. Sie können auch die Mindestanzahl von Shards angeben, die mit Auto Scaling verwaltet werden sollen. Dieser Wert muss mindestens 1 betragen und gleich oder kleiner sein als der Wert, der für die maximale Anzahl von Scherben 250 angegeben ist.

Replikas

Sie können die maximale Anzahl von Replikaten angeben, mit denen Valkey und Redis OSS Auto ElastiCache Scaling verwaltet werden sollen. Dieser Wert muss gleich oder kleiner 5 sein. Sie können auch die Mindestanzahl von Replikaten angeben, die mit Auto Scaling verwaltet werden sollen. Dieser Wert muss mindestens 1 betragen und gleich oder kleiner als der für die maximale Anzahl von Replikaten angegebene Wert 5 sein.

Um die minimale und maximale Anzahl davon zu ermitteln shards/replicas , die Sie für den typischen Datenverkehr benötigen, testen Sie Ihre Auto Scaling Scaling-Konfiguration mit der erwarteten Datenverkehrsrate zu Ihrem Modell.

 Note

ElastiCache Auto Scaling-Richtlinien erhöhen die Clusterkapazität, bis sie Ihre definierte Maximalgröße erreicht hat oder bis Servicebeschränkungen gelten. Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

 Important

Scaling-in findet statt, wenn es keinen Verkehr gibt. Wenn der Traffic einer Variante Null wird, wird ElastiCache automatisch auf die angegebene Mindestanzahl von Instanzen skaliert.

Abkühlungszeit

Sie können die Reaktionsfähigkeit einer Skalierungsrichtlinie für die Zielverfolgung anpassen, indem Sie Ruhephasen hinzufügen, nachdem eine Skalierung Ihres DB-Clusters erfolgt ist. Eine Ruhephase blockiert nachfolgende Scale-in- oder Scale-out-Anforderungen bis zum Ablauf der Frist. Dies verlangsamt das Löschen von shards/replicas in Ihrem ElastiCache for Valkey- und Redis OSS-Cluster für Scale-In-Anfragen und das Erstellen von für Scale-Out-Anfragen. shards/replicas Sie können die folgenden Ruhephasen angeben:

- Eine Scale-In-Aktivität reduziert die Anzahl der in Ihrem Cluster. shards/replicas Eine Ruhephase zum Abskalieren gibt die Zeitspanne in Sekunden an, nach der eine Abskalierung abgeschlossen sein muss, bevor eine weitere Abskalierung gestartet werden kann.
- Eine Scale-Out-Aktivität erhöht die Anzahl der shards/replicas in Ihrem Cluster. Eine Ruhephase zum Aufskalieren gibt die Zeitspanne in Sekunden an, nach der eine Aufskalierung abgeschlossen sein muss, bevor eine weitere Aufskalierung gestartet werden kann.

Wenn keine Ruhephase für das Abskalieren oder Aufskalieren angegeben ist, beträgt der Standardwert für die Aufskalierung 600 Sekunden und für das Abskalieren 900 Sekunden.

Aktivieren oder Deaktivieren von Scale-In-Aktivitäten

Sie können die Scale-In-Aktivitäten für eine Richtlinie aktivieren oder deaktivieren. Durch die Aktivierung von Scale-In-Aktivitäten kann die Skalierungsrichtlinie gelöscht werden. shards/replicas. When scale-in activities are enabled, the scale-in cooldown period in the scaling policy applies to scale-in activities. Disabling scale-in activities prevents the scaling policy from deleting shards/replicas

 Note

Scale-Out-Aktivitäten sind immer aktiviert, sodass die Skalierungsrichtlinie nach Bedarf ElastiCache Shards oder Replikate erstellen kann.

Für Auto Scaling sind IAM-Berechtigungen erforderlich

ElastiCache für Valkey und Redis OSS wird Auto Scaling durch eine Kombination aus ElastiCache CloudWatch, und Application Auto Scaling ermöglicht. APIs Cluster werden mit Application Auto Scaling erstellt und aktualisiert ElastiCache CloudWatch, Alarme werden mit erstellt und Skalierungsrichtlinien werden erstellt. Zusätzlich zu den Standard-IAM-Berechtigungen für das Erstellen und Aktualisieren von Clustern muss der IAM-Benutzer, der auf die ElastiCache Auto Scaling Scaling-Einstellungen zugreift, über die entsprechenden Berechtigungen für die Dienste verfügen, die dynamische Skalierung unterstützen. In dieser neuesten Richtlinie haben wir mit der Aktion Unterstützung für die vertikale Skalierung von Memcached hinzugefügt. `elasticache:ModifyCacheCluster` IAM-Benutzer müssen die Berechtigung haben, die Aktionen in der folgenden Beispielrichtlinie zu verwenden:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:*",
        "elasticache:DescribeReplicationGroups",
        "elasticache:ModifyReplicationGroupShardConfiguration",
        "elasticache:IncreaseReplicaCount",
        "elasticache:DecreaseReplicaCount",
        "elasticache:DescribeCacheClusters",
        "elasticache:DescribeCacheParameters",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DisableAlarmActions",
        "cloudwatch:EnableAlarmActions",
        "iam:CreateServiceLinkedRole",
        "sns:CreateTopic",
        "sns:Subscribe",
        "sns:Get*"
      ]
    }
  ]
}
```

```

        "sns:List*"
    ],
    "Resource": "arn:aws:iam::123456789012:role/autoscaling-roles-for-
cluster"
}
]
}

```

Servicegebundene Rolle

Der auto Skalierungsdienst ElastiCache für Valkey und Redis OSS benötigt außerdem die Erlaubnis, Ihre Cluster und CloudWatch Alarme zu beschreiben, sowie Berechtigungen, Ihre ElastiCache Zielkapazität in Ihrem Namen zu ändern. Wenn Sie Auto Scaling für Ihren Cluster aktivieren, wird eine dienstverknüpfte Rolle mit dem Namen `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG` erstellt. Diese dienstbezogene Rolle gewährt ElastiCache Auto Scaling-Berechtigungen, um die Alarme für Ihre Richtlinien zu beschreiben, die aktuelle Kapazität der Flotte zu überwachen und die Kapazität der Flotte zu ändern. Die serviceverknüpfte Rolle ist die Standardrolle für ElastiCache Auto Scaling. Weitere Informationen finden Sie unter [Serviceverknüpfte Rollen ElastiCache für Redis OSS Auto Scaling im Application Auto Scaling](#) Scaling-Benutzerhandbuch.

Bewährte Methoden für die Auto Scaling

Wir empfehlen vor der Registrierung für Auto Scaling Folgendes:

1. Verwenden Sie nur eine Tracking-Metrik – Ermitteln Sie, ob Ihr Cluster über CPU- oder datenintensive Workloads verfügt, und verwenden Sie eine entsprechende vordefinierte Metrik, um die Skalierungsrichtlinie zu definieren.
 - Engine-CPU: `ElastiCachePrimaryEngineCPUUtilization` (Shard-Dimension) oder `ElastiCacheReplicaEngineCPUUtilization` (Replikatdimension)
 - Datenbanknutzung: `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` Diese Skalierungsrichtlinie funktioniert am besten, wenn „maxmemory-policy“ im Cluster auf „noeviction“ festgelegt ist.

Wir empfehlen Ihnen, mehrere Richtlinien pro Dimension auf dem Cluster zu vermeiden. ElastiCache Für Valkey und Redis OSS skaliert Auto Scaling das skalierbare Ziel, wenn irgendwelche Richtlinien zur Zielverfolgung bereit für die horizontale Skalierung sind. Die

Skalierung erfolgt jedoch nur, wenn alle Zielverfolgungsrichtlinien (mit aktiviertem Scale-In-Teil) skalierbar sind. Wenn mehrere Richtlinien das skalierbare Ziel anweisen, gleichzeitig zu herauf oder herunter zu skalieren, skaliert auf Grundlage der Richtlinie, die die größte Kapazität für das Herauf- und Herunterskalieren bietet.

2. Benutzerdefinierte Metriken für die Zielverfolgung – Seien Sie vorsichtig, wenn Sie benutzerdefinierte Metriken für die Zielverfolgung verwenden, da die automatische Skalierung am besten geeignet ist, um proportional zu Änderungen der für die Richtlinie ausgewählten Metriken auf-/abzuskalieren. Wenn solche Metriken, die sich nicht proportional zu den Skalierungsaktionen ändern, zur Richtlinienerstellung verwendet werden, kann dies zu kontinuierlichen Auf- oder Abskalierungsaktionen führen, was sich auf Verfügbarkeit oder Kosten auswirken kann.

Vermeiden Sie bei Daten-Tiering-Clustern (Instance-Familie r6gd) die Verwendung speicherbasierter Metriken für die Skalierung.

3. Geplante Skalierung — Wenn Sie feststellen, dass Ihre Arbeitslast deterministisch ist (sie erreicht high/low zu einem bestimmten Zeitpunkt), empfehlen wir die Verwendung von Scheduled Scaling und die Konfiguration Ihrer Zielkapazität entsprechend den Anforderungen. Target Tracking eignet sich am besten für nicht-deterministische Workloads und für einen Cluster für den Betrieb mit der erforderlichen Zielmetrik, indem er aufskaliert, wenn Sie mehr Ressourcen benötigen, und abskaliert, wenn Sie weniger benötigen.
4. Scale-In deaktivieren — Die automatische Skalierung bei Target Tracking eignet sich am besten für Cluster mit schrittweisen Workloads, da In-Metriken increase/decrease aufeinanderfolgende spikes/dip Scale-Out-/In-Oszillationen auslösen können. Um solche Schwankungen zu vermeiden, können Sie mit deaktivierter Abskalierung beginnen. Später können Sie jederzeit manuell nach Ihren Bedürfnissen abskalieren.
5. Testen Sie Ihre Anwendung — Wir empfehlen Ihnen, Ihre Anwendung mit Ihren geschätzten Min/Max Workloads zu testen, um die für den Cluster shards/replicas erforderlichen Mindest-/Höchstwerte zu ermitteln und gleichzeitig Skalierungsrichtlinien zu erstellen, um Verfügbarkeitsprobleme zu vermeiden. Die automatische Skalierung kann bis zum Maximal- und Minimalwert auf- bzw. abskalieren, der für das Ziel konfiguriert wurde.
6. Definition des Zielwerts — Sie können die entsprechenden CloudWatch Metriken für die Clusterauslastung über einen Zeitraum von vier Wochen analysieren, um den Schwellenwert für den Zielwert zu ermitteln. Wenn Sie sich immer noch nicht sicher sind, welchen Wert Sie wählen möchten, empfehlen wir, mit dem minimal unterstützten vordefinierten Metrikwert zu beginnen.
7. AutoScaling On Target Tracking eignet sich am besten für Cluster mit einheitlicher Verteilung der Workloads über alle shards/replicas Dimensionen. Eine ungleichmäßige Verteilung kann zu folgenden Faktoren führen:

- Skalierung, wenn sie aufgrund der Arbeitslast spike/dip bei einigen Hot-Shards/Replicas nicht erforderlich ist.
- Keine Skalierung, wenn erforderlich, da insgesamt durchschnittlich nahe am Ziel liegt, obwohl Hot-Shards/Replikate vorhanden sind.

Note

Bei der Skalierung Ihres Clusters ElastiCache werden die Funktionen, die in einem der vorhandenen Knoten geladen wurden (zufällig ausgewählt), automatisch auf die neuen Knoten repliziert. Wenn Ihr Cluster über Valkey oder Redis OSS 7.0 oder höher verfügt und Ihre Anwendung [Functions](#) verwendet, empfehlen wir, alle Ihre Funktionen vor dem Skalieren auf alle Shards zu laden, damit Ihr Cluster nicht mit unterschiedlichen Funktionen auf verschiedenen Shards endet.

Beachten Sie nach der Registrierung Folgendes AutoScaling:

- Es gibt Einschränkungen bei den unterstützten Konfigurationen für die automatische Skalierung, daher empfehlen wir, die Konfiguration einer Replikationsgruppe, die für die automatische Skalierung registriert ist, nicht zu ändern. Im Folgenden sind einige Beispiele aufgeführt:
 - Manuelles Ändern des Instance-Typs in nicht unterstützte Typen.
 - Zuordnen der Replikationsgruppe zu einem globalen Datenspeicher.
 - ÄndernReservedMemoryPercent-Parameter.
 - Manuell konfigurierte increasing/decreasing shards/replicas beyond the Min/Max Kapazität bei der Richtlinienerstellung.

Verwenden von Auto Scaling mit Shards

Damit ElastiCache können AutoScaling Sie Tracking- und geplante Richtlinien mit Ihrer Valkey- oder Redis OSS-Engine verwenden.

Im Folgenden finden Sie Einzelheiten zur Zielverfolgung und zu geplanten Richtlinien sowie dazu, wie Sie diese mithilfe von und anwenden können. AWS-Managementkonsole AWS CLI APIs

Themen

- [Skalierungsrichtlinien für die Ziel-Nachverfolgung](#)

- [Hinzufügen einer Skalierungsrichtlinie](#)
- [Registrieren eines skalierbaren Ziels](#)
- [Definieren einer Skalierungsrichtlinie](#)
- [Deaktivieren der Scale-In-Aktivität](#)
- [Anwenden einer Skalierungsrichtlinie](#)
- [Bearbeiten einer Skalierungsrichtlinie](#)
- [Löschen einer Skalierungsrichtlinie](#)
- [CloudFormation Für Auto Scaling Scaling-Richtlinien verwenden](#)
- [Geplante Skalierung](#)

Skalierungsrichtlinien für die Ziel-Nachverfolgung

Bei Skalierungsrichtlinien für die Zielverfolgung wählen Sie eine Metrik aus und legen einen Zielwert fest. ElastiCache für Valkey und Redis erstellt und verwaltet OSS Auto Scaling die CloudWatch Alarmer, die die Skalierungsrichtlinie auslösen, und berechnet die Skalierungsanpassung auf der Grundlage der Metrik und des Zielwerts. Durch die Skalierungsrichtlinie wird so viel Kapazität wie erforderlich hinzugefügt oder entfernt, damit die Metrik auf oder nahe an dem Zielwert gehalten wird. Abgesehen davon, dass eine Skalierungsrichtlinie für die Ziel-Nachverfolgung die Metrik nahe an dem Zielwert hält, passt sie sich auch an die Schwankungen in der Metrik aufgrund eines schwankenden Lastmusters an und verringert schnelle Schwankungen der Kapazität der Flotte.

Nehmen wir zum Beispiel eine Skalierungsrichtlinie, die die vordefinierte `ElastiCachePrimaryEngineCPUUtilization`-Durchschnittsmetrik mit konfiguriertem Zielwert verwendet. Eine solche Richtlinie kann die CPU-Nutzung auf oder nahe an dem Zielwert halten.

Vordefinierte Metriken

Eine vordefinierte Metrik ist eine Struktur, die sich auf einen bestimmten Namen, eine Dimension und eine Statistik (average) einer bestimmten Metrik bezieht. CloudWatch Ihre Auto-Scaling-Richtlinie definiert eine der folgenden vordefinierten Metriken für Ihren Cluster:

Vordefinierter Metrikname	CloudWatch Metrikname	CloudWatch Metrische Dimension	Nicht geeignete Instance-Typen
ElastiCachePrimaryEngineCPUUtilization	EngineCPUUtilization	ReplicationGroupId, Rolle = Primär	Keine
ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage	DatabaseCapacityUsageCountedForEvictPercentage	Metriken für Valkey- oder Redis OSS-Replikationsgruppen	Keine
ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage	DatabaseMemoryUsageCountedForEvictPercentage	Metriken für die OSS-Replikationsgruppe von Valkey oder Redis	R6gd

Instance-Typen mit Daten-Tiering können

ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage nicht verwenden, da diese Instance-Typen Daten sowohl im Arbeitsspeicher als auch im SSD-Speicher ablegen. Der erwartete Anwendungsfall für Instances mit Daten-Tiering ist eine 100-prozentige Speichernutzung und das Auffüllen des SSD-Speichers nach Bedarf.

Kriterien für die Auto Scaling für Shards

Wenn der Service feststellt, dass Ihre vordefinierte Metrik gleich oder größer als die Zieleinstellung ist, erhöht er die Kapazität Ihrer Shards automatisch. ElastiCache für Valkey und Redis OSS skaliert OSS Ihre Cluster-Shards um eine Anzahl, die der größeren von zwei Zahlen entspricht: prozentuale

Abweichung von Target und 20 Prozent der aktuellen Shards. Bei Scale-In erfolgt ElastiCache keine auto Skalierung, es sei denn, der Gesamtmetrikwert liegt unter 75 Prozent Ihres definierten Ziels.

Wenn Sie für ein Scal-Out-Beispiel 50 Shards und

- Wenn Ihr Target die Sicherheitslücken um 30 Prozent überschreitet, wird eine ElastiCache Skalierung um 30 Prozent vorgenommen, was zu 65 Shards pro Cluster führt.
- Wenn Ihr Target die Sicherheitslücken um 10 Prozent überschreitet, wird standardmäßig ein Minimum von 20 Prozent ElastiCache skaliert, was zu 60 Shards pro Cluster führt.

Wenn Sie beispielsweise einen Zielwert von 60 Prozent ausgewählt haben, erfolgt die auto Skalierung ElastiCache erst, wenn die Metrik 45 Prozent oder weniger beträgt (25 Prozent unter dem Ziel von 60 Prozent).

Überlegungen zum Auto Scaling

Beachten Sie folgende Überlegungen:

- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung geht davon aus, dass sie eine horizontale Skalierung nach oben vornehmen soll, wenn die angegebene Metrik über dem Zielwert liegt. Sie können keine Skalierungsrichtlinie für die Zielverfolgung verwenden, um eine Skalierung vorzunehmen, wenn die angegebene Metrik unter dem Zielwert liegt. ElastiCache für Valkey und Redis skaliert OSS die Shards um eine Abweichung von mindestens 20 Prozent vom Ziel vorhandener Shards im Cluster.
- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung nimmt keine Skalierung vor, wenn die angegebene Metrik unzureichende Daten aufweist. Es wird keine Abskalierung vorgenommen, da unzureichende Daten nicht als geringe Auslastung interpretiert werden.
- Möglicherweise werden Lücken zwischen den Datenpunkten für den Zielwert und die aktuelle Metrik angezeigt. Das liegt daran, dass ElastiCache Auto Scaling immer konservativ agiert, indem es auf- oder abrundet, wenn es bestimmt, wie viel Kapazität hinzugefügt oder entfernt werden soll. Dadurch wird verhindert, dass zu wenig Kapazität hinzugefügt oder zu viel Kapazität entfernt wird.
- Um die Verfügbarkeit der Anwendung sicherzustellen, wird der Service schnellstmöglich proportional zur Metrik hochskaliert, jedoch etwas langsamer herunterskaliert.
- Sie können mehrere Skalierungsrichtlinien zur Zielverfolgung für einen OSS-Cluster ElastiCache für Valkey und Redis verwenden, vorausgesetzt, dass jeder von ihnen eine andere Metrik verwendet. Die Absicht von ElastiCache Auto Scaling besteht darin, der Verfügbarkeit immer Priorität einzuräumen. Daher unterscheidet sich das Verhalten je nachdem, ob die

Zielverfolgungsrichtlinien für die horizontale Skalierung oder für die Skalierung bereit sind.

Sofern Richtlinien für die Ziel-Nachverfolgung für die Skalierung nach oben bereit sind, findet eine Skalierung des Service nach oben statt. Eine Skalierung nach unten wird jedoch nur vorgenommen, wenn alle Richtlinien für die Ziel-Nachverfolgung (mit aktivierter Skalierung nach unten) zur Skalierung nach unten bereit sind.

- Bearbeiten oder löschen Sie nicht die CloudWatch Alarmer, die ElastiCache Auto Scaling für eine Skalierungsrichtlinie zur Zielverfolgung verwaltet. ElastiCache Auto Scaling löscht die Alarmer automatisch, wenn Sie die Skalierungsrichtlinie löschen.
- ElastiCache Auto Scaling verhindert nicht, dass Sie Cluster-Shards manuell ändern. Diese manuellen Anpassungen wirken sich nicht auf bestehende CloudWatch Alarmer aus, die mit der Skalierungsrichtlinie verknüpft sind, können sich jedoch auf Metriken auswirken, die diese CloudWatch Alarmer auslösen können.
- Diese von Auto Scaling verwalteten CloudWatch Alarmer werden anhand der AVG-Metrik für alle Shards im Cluster definiert. So kann Hot-Shards zu einem beliebigen Szenario führen:
 - Skalierung, wenn sie nicht erforderlich ist, weil die Belastung einiger Hot-Shards einen Alarm auslöst CloudWatch
 - nicht skalieren, wenn dies erforderlich ist, aufgrund aggregierter AVG über alle Shards, die den Alarm nicht verletzen.
- ElastiCache Die Standardgrenzwerte für Knoten pro Cluster gelten weiterhin. Wenn Sie sich also für Auto Scaling entscheiden und erwarten, dass maximale Knoten mehr als die Standardgrenze sind, fordern Sie eine Limiterhöhung bei [AWS -Service-Limits](#) an und wählen Sie den Limit-Typ Knoten pro Cluster pro Instance-Typ.
- Stellen Sie sicher, dass in Ihrer VPC genügend ENIs (Elastic Network Interfaces) verfügbar sind, die beim Scale-Out benötigt werden. Weitere Informationen finden Sie unter [Elastic-Network-Schnittstellen](#).
- Wenn nicht genügend Kapazität verfügbar ist EC2, wird ElastiCache Auto Scaling nicht skaliert und verzögert, bis die Kapazität verfügbar ist.
- ElastiCache für Redis OSS Auto Scaling entfernt beim Scale-In keine Shards mit Steckplätzen, die nach der Serialisierung eine Objektgröße von mehr als 256 MB haben.
- Während des Scale-Ins werden Shards nicht entfernt, wenn nicht genügend Speicher für die resultierende Shard-Konfiguration verfügbar ist.

Hinzufügen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie hinzufügen, indem Sie die verwenden AWS-Managementkonsole.

So fügen Sie einem OSS-Cluster ElastiCache für Valkey und Redis eine Auto Scaling Scaling-Richtlinie hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS aus.
3. Wählen Sie den Cluster aus, dem Sie eine Richtlinie hinzufügen möchten (wählen Sie den Cluster-Namen und nicht die Schaltfläche links davon aus).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie Add dynamic scaling (Dynamische Skalierung hinzufügen) aus.
6. Geben Sie unter Policy Name (Richtliniennamen) einen Namen für diese Richtlinie ein.
7. Für Skalierbare Dimension wählen Sie Shards aus.
8. Wählen Sie für die Zielmetrik eine der nachstehenden Optionen aus:
 - Primäre CPU-Auslastung, um eine Richtlinie auf der Grundlage der durchschnittlichen CPU-Auslastung zu erstellen.
 - Memory (Arbeitsspeicher) Um eine Richtlinie zu erstellen, die auf dem durchschnittlichen Datenbankspeicher basiert.
 - Kapazität zum Erstellen einer Richtlinie auf der Grundlage der durchschnittlichen Datenbankkapazitätsauslastung. Die Kapazitätsmetrik umfasst die Speicher- und SSD-Auslastung für Daten-Tiering-Instances und die Speichernutzung für alle anderen Instance-Typen.
9. Wählen Sie für den Zielwert einen Wert größer oder gleich 35 und kleiner oder gleich 70 aus. Bei der automatischen Skalierung wird dieser Wert für die ausgewählte Zielmetrik auf allen Ihren ElastiCache Shards beibehalten:
 - Primäre CPU-Auslastung: behält den Zielwert für die EngineCPUUtilization-Metrik auf den Primärknoten bei.
 - Speicher: behält den Zielwert für die DatabaseMemoryUsageCountedForEvictPercentage-Metrik bei

- Kapazität behält den Zielwert für die DatabaseCapacityUsageCountedForEvictPercentage-Metrik bei

Cluster-Shards werden hinzugefügt oder entfernt, um die Metrik in der Nähe des angegebenen Wertes zu halten.

10. (Optional) Die Ruhephase für Scale-In- oder Scale-Out-Ruhephase werden von der Konsole nicht unterstützt. Verwenden Sie die AWS CLI , um die Abklingzeitwerte zu ändern.
11. Geben Sie für Mindestkapazität die Mindestanzahl von Shards ein, für deren Verwaltung die ElastiCache Auto Scaling Scaling-Richtlinie erforderlich ist.
12. Geben Sie für Maximale Kapazität die maximale Anzahl von Shards ein, für deren Verwaltung die ElastiCache Auto Scaling Scaling-Richtlinie erforderlich ist. Dieser Wert muss gleich oder kleiner 250 sein.
13. Wählen Sie Erstellen aus.

Registrieren eines skalierbaren Ziels

Bevor Sie Auto Scaling mit einem OSS-Cluster ElastiCache für Valkey und Redis verwenden können, registrieren Sie Ihren Cluster bei ElastiCache Auto Scaling. Sie tun dies, um die Skalierungsdimension und die Grenzwerte zu definieren, die auf diesen Cluster angewendet werden sollen. ElastiCache Auto Scaling skaliert den Cluster dynamisch entlang der `elasticache:replication-group:NodeGroups` skalierbaren Dimension, die die Anzahl der Cluster-Shards darstellt.

Unter Verwendung der AWS CLI

Verwenden Sie den [register-scalable-target](#) Befehl mit den folgenden Parametern, um Ihren OSS-Cluster ElastiCache für Valkey und Redis zu registrieren:

- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest
- `--resource-id`— Die Ressourcen-ID für den Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des Clusters `replication-group/myscalablecluster`.
- `--scalable-dimension` – Legen Sie diesen Wert auf `elasticache:replication-group:NodeGroups` fest.
- `--max-capacity` — Die maximale Anzahl von Shards, die durch ElastiCache Auto Scaling verwaltet werden können. Informationen zur Beziehung zwischen `--min-capacity`, `--max-`

capacity und der Anzahl der DB-Instances in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

- `--min-capacity` — Die Mindestanzahl von Shards, die durch ElastiCache Auto Scaling verwaltet werden müssen. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der DB-Instances in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

Example

Im folgenden Beispiel registrieren Sie einen ElastiCache Cluster mit dem Namen `myscalablecluster`. Die Registrierung zeigt an, dass der DB-Cluster dynamisch skaliert werden soll, um zwischen einer und acht -Replicas zu haben.

Für Linux, macOS oder Unix:

```
aws application-autoscaling register-scalable-target \
  --service-namespace elasticache \
  --resource-id replication-group/myscalablecluster \
  --scalable-dimension elasticache:replication-group:NodeGroups \
  --min-capacity 1 \
  --max-capacity 10 \
```

Für Windows:

```
aws application-autoscaling register-scalable-target ^
  --service-namespace elasticache ^
  --resource-id replication-group/myscalablecluster ^
  --scalable-dimension elasticache:replication-group:NodeGroups ^
  --min-capacity 1 ^
  --max-capacity 10 ^
```

Verwenden der API

Verwenden Sie den [register-scalable-target](#) Befehl mit den folgenden Parametern, um Ihren ElastiCache Cluster zu registrieren:

- `ServiceNamespace` — Setzen Sie diesen Wert auf `elasticache`.
- `ResourceID` — Die Ressourcen-ID für den ElastiCache Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise `replication-group/myscalablecluster` der Name des Clusters.

- **ScalableDimension** — Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.
- **MinCapacity** — Die Mindestanzahl von Shards, die durch ElastiCache Auto Scaling verwaltet werden müssen. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).
- **MaxCapacity** — Die maximale Anzahl von Shards, die durch ElastiCache Auto Scaling verwaltet werden können. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

Example

Im folgenden Beispiel registrieren Sie einen ElastiCache Cluster, der `myscalecluster` mit der Application Auto Scaling API benannt ist. Diese Registrierung zeigt an, dass der Cluster dynamisch skaliert werden soll, um zwischen 1 und 5 Replikate zu haben.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definieren einer Skalierungsrichtlinie

Eine Konfiguration der Skalierungsrichtlinie für die Zielverfolgung wird durch einen JSON-Block repräsentiert, in dem die Metriken und Zielwerte definiert sind. Sie können die Konfiguration einer Skalierungsrichtlinie als JSON-Block in einer Textdatei speichern. Sie verwenden diese Textdatei, wenn Sie die AWS CLI oder die Application Auto Scaling Scaling-API aufrufen. Weitere Informationen

zur Syntax der Richtlinienkonfiguration finden Sie unter [TargetTrackingScalingPolicyConfiguration](#) in der Application Auto Scaling API Reference.

Die folgenden Optionen stehen zur Verfügung, um eine Konfiguration der Skalierungsrichtlinien für die Zielverfolgung zu definieren:

Themen

- [Verwenden einer vordefinierten Metrik](#)
- [Verwenden einer benutzerdefinierten Metrik](#)
- [Verwenden von Ruhephasen](#)

Verwenden einer vordefinierten Metrik

Mithilfe vordefinierter Metriken können Sie schnell eine Skalierungsrichtlinie für die Zielverfolgung für einen OSS-Cluster ElastiCache für Valkey und Redis definieren, der mit der Zielverfolgung in Auto Scaling funktioniert. ElastiCache

ElastiCache unterstützt derzeit die folgenden vordefinierten Metriken in NodeGroup Auto Scaling:

- `ElastiCachePrimaryEngineCPUUtilization`— Der Durchschnittswert der `EngineCPUUtilization` Metrik CloudWatch auf allen Primärknoten im Cluster.
- `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`— Der Durchschnittswert der `DatabaseMemoryUsageCountedForEvictPercentage` Metrik CloudWatch für alle Primärknoten im Cluster.
- `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`— Der Durchschnittswert der `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` Metrik CloudWatch für alle Primärknoten im Cluster.

Weitere Informationen über die Metriken

`EngineCPUUtilization`, `DatabaseMemoryUsageCountedForEvictPercentage` und `DatabaseCapacityUsageCountedForEvictPercentage` sind Metriken, die Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#) finden. Um eine vordefinierte Metrik in Ihrer Skalierungsrichtlinie zu verwenden, erstellen Sie eine Zielverfolgungskonfiguration für Ihre Skalierungsrichtlinie. Diese Konfiguration muss eine `PredefinedMetricSpecification` für die vordefinierte Metrik und einen `TargetValue` für den Zielwert dieser Metrik enthalten.

Example

Das folgende Beispiel beschreibt eine typische Richtlinienkonfiguration für die Skalierung nach Zielverfolgung für einen OSS-Cluster ElastiCache für Valkey und Redis. In dieser Konfiguration wird die `ElastiCachePrimaryEngineCPUUtilization` vordefinierte Metrik verwendet, um den Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 40 Prozent auf allen Primärknoten im Cluster anzupassen.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
  }
}
```

Verwenden einer benutzerdefinierten Metrik

Durch die Verwendung von benutzerdefinierten Metriken können Sie eine Skalierungsrichtlinie für die Zielverfolgung definieren, die Ihren individuellen Anforderungen entspricht. Sie können eine benutzerdefinierte Metrik basierend auf einer beliebigen ElastiCache -Metrik definieren, die sich proportional zur Skalierung ändert. Nicht alle ElastiCache Metriken eignen sich für die Zielverfolgung. Die Metrik muss eine gültige Auslastungsmetrik sein und beschreiben, wie ausgelastet eine Instance ist. Der Wert der Metrik muss sich proportional zur Anzahl der -Replicas im -DB-Cluster erhöhen oder verringern. Diese proportionale Erhöhung oder Verminderung ist notwendig, um die metrischen Daten zur proportionalen Skalierung oder in der Anzahl der -Repliken zu verwenden.

Example

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt eine benutzerdefinierte Metrik einen ElastiCache für Redis OSS-Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 50 Prozent für alle Shards in einem genannten Cluster an. `my-db-cluster`

```
{
  "TargetValue": 50,
  "CustomizedMetricSpecification":
  {
    "MetricName": "EngineCPUUtilization",
    "Namespace": "AWS/ElastiCache",
    "Dimensions": [
```

```
{
  {
    "Name": "ReplicationGroup","Value": "my-db-cluster"
  },
  {
    "Name": "Role","Value": "PRIMARY"
  }
],
"Statistic": "Average",
"Unit": "Percent"
}
}
```

Verwenden von Ruhephasen

Sie können einen Wert in Sekunden für `ScaleOutCooldown` angeben, um eine Ruhephase für die Aufskalierung Ihres Clusters hinzuzufügen. Ähnlich können Sie einen Wert in Sekunden für `ScaleInCooldown` angeben, um eine Ruhephase für die Abskalierung Ihres Clusters hinzuzufügen. Weitere Informationen finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der API-Referenz für Application Auto Scaling.

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration wird die `ElastiCachePrimaryEngineCPUUtilization` vordefinierte Metrik verwendet, um einen ElastiCache für Redis OSS-Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 40 Prozent auf allen primären Knoten in diesem Cluster anzupassen. Die Konfiguration sieht eine Ruhephase von 10 Minuten zum Abskalieren und eine Ruhephase von 5 Minuten zum Aufskalieren vor.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
  },
  "ScaleInCooldown": 600,
  "ScaleOutCooldown": 300
}
```

Deaktivieren der Scale-In-Aktivität

Sie können verhindern, dass die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung in Ihrem Cluster skaliert, indem Sie die Scale-In-Aktivität deaktivieren. Das Deaktivieren der Scale-

In-Aktivität verhindert das Löschen von -Repliken durch die Skalierungsrichtlinie, und erlaubt der Skalierungsrichtlinie dennoch, Repliken nach Bedarf zu erstellen.

Sie können einen booleschen Wert für `DisableScaleIn` angeben, um die Scale-In-Aktivität für Ihren Cluster zu aktivieren oder zu deaktivieren. Weitere Informationen finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der API-Referenz für Application Auto Scaling.

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt die `ElastiCachePrimaryEngineCPUUtilization` vordefinierte Metrik einen OSS-Cluster ElastiCache für Valkey und Redis auf der Grundlage einer durchschnittlichen CPU-Auslastung von 40 Prozent auf allen primären Knoten in diesem Cluster an. Die Konfiguration deaktiviert die Scale-In-Aktivität für die Skalierungsrichtlinie.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
  },
  "DisableScaleIn": true
}
```

Anwenden einer Skalierungsrichtlinie

Nachdem Sie Ihren Cluster bei ElastiCache for Valkey und Redis OSS Auto Scaling registriert und eine Skalierungsrichtlinie definiert haben, wenden Sie die Skalierungsrichtlinie auf den registrierten Cluster an. Um eine Skalierungsrichtlinie auf einen ElastiCache für Redis OSS-Cluster anzuwenden, können Sie die AWS CLI oder die Application Auto Scaling Scaling-API verwenden.

Anwenden einer Skalierungsrichtlinie mithilfe der AWS CLI

Um eine Skalierungsrichtlinie auf Ihren OSS-Cluster ElastiCache für Valkey und Redis anzuwenden, verwenden Sie den [put-scaling-policy](#) Befehl mit den folgenden Parametern:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--policy-type` – Legen Sie diesen Wert auf `TargetTrackingScaling` fest.
- `--resource-id` — Die Ressourcen-ID. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des Clusters. `replication-group/myscalablecluster`

- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.
- `--target-tracking-scaling-policy-configuration` — Die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung, die für den Cluster verwendet werden soll.

Im folgenden Beispiel wenden Sie eine Skalierungsrichtlinie `myscalablepolicy` zur Zielverfolgung an, die auf einen mit Auto Scaling benannten OSS-Cluster ElastiCache für Valkey und Redis OSS benannt `myscalablecluster` ist. ElastiCache Dazu verwenden Sie eine Richtlinienkonfiguration, die in einer Datei mit dem Namen `config.json` gespeichert ist.

Für Linux, macOS oder Unix:

```
aws application-autoscaling put-scaling-policy \  
  --policy-name myscalablepolicy \  
  --policy-type TargetTrackingScaling \  
  --resource-id replication-group/myscalablecluster \  
  --service-namespace elasticache \  
  --scalable-dimension elasticache:replication-group:NodeGroups \  
  --target-tracking-scaling-policy-configuration file://config.json
```

Für Windows:

```
aws application-autoscaling put-scaling-policy ^  
  --policy-name myscalablepolicy ^  
  --policy-type TargetTrackingScaling ^  
  --resource-id replication-group/myscalablecluster ^  
  --service-namespace elasticache ^  
  --scalable-dimension elasticache:replication-group:NodeGroups ^  
  --target-tracking-scaling-policy-configuration file://config.json
```

Anwendung einer Skalierungsrichtlinie mit der API

Um eine Skalierungsrichtlinie auf Ihren OSS-Cluster ElastiCache für Valkey und Redis anzuwenden, verwenden Sie den [PutScalingPolicy](#) AWS CLI Befehl mit den folgenden Parametern:

- `—policy-name`– Der Name der Skalierungsrichtlinie.

- `--resource-id` — Die Ressourcen-ID. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des Clusters. `replication-group/myscalablecluster`
- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.
- `--target-tracking-scaling-policy-configuration` — Die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung, die für den Cluster verwendet werden soll.

Im folgenden Beispiel wenden Sie eine Zielverfolgungs-Skalierungsrichtlinie mit dem Namen auf einen ElastiCache Cluster `myscalablepolicy` an, der `myscalablecluster` mit ElastiCache Auto Scaling benannt ist. Sie verwenden die Richtlinienkonfiguration `ElastiCachePrimaryEngineCPUUtilization` basierend auf einer vordefinierten Metrik.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "TargetValue": 40.0,
    "PredefinedMetricSpecification": {
      "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
    }
  }
}
```


Bearbeiten einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie mit der AWS-Managementkonsole, der oder der AWS CLI Application Auto Scaling API bearbeiten.

Bearbeiten einer Skalierungsrichtlinie mit dem AWS-Managementkonsole

So bearbeiten Sie eine Auto Scaling Scaling-Richtlinie ElastiCache für einen OSS-Cluster für Valkey und Redis

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich die entsprechende Engine aus.
3. Wählen Sie den Cluster aus, dem Sie eine Richtlinie hinzufügen möchten (wählen Sie den Cluster-Namen und nicht die Schaltfläche links davon aus).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie unter Scaling policies (Skalierungsrichtlinien) die Schaltfläche links neben der Auto-Scaling-Richtlinie aus, die Sie ändern möchten, und klicken Sie dann auf Modify (Ändern).
6. Nehmen Sie die erforderlichen Änderungen an der Richtlinie vor.
7. Wählen Sie Ändern aus.

Bearbeiten einer Skalierungsrichtlinie mithilfe der AWS CLI AND-API

Sie können die AWS CLI oder die Application Auto Scaling Scaling-API verwenden, um eine Skalierungsrichtlinie auf die gleiche Weise zu bearbeiten, wie Sie eine Skalierungsrichtlinie anwenden:

- Wenn Sie die verwenden AWS CLI, geben Sie den Namen der Richtlinie, die Sie bearbeiten möchten, im `--policy-name` Parameter an. Legen Sie neue Werte für die Parameter fest, die Sie ändern möchten.
- Wenn Sie die Application Auto Scaling API verwenden, geben Sie den Namen der zu bearbeitenden Richtlinie im Parameter `PolicyName` an. Legen Sie neue Werte für die Parameter fest, die Sie ändern möchten.

Weitere Informationen finden Sie unter [Anwenden einer Skalierungsrichtlinie](#).

Löschen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie mithilfe der AWS-Managementkonsole, der oder der AWS CLI Application Auto Scaling API löschen.

Löschen einer Skalierungsrichtlinie mithilfe der AWS-Managementkonsole

So löschen Sie eine Auto Scaling Scaling-Richtlinie für einen ElastiCache für Redis OSS-Cluster

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS aus.
3. Wählen Sie den Cluster aus, dessen Auto-Scaling-Richtlinie Sie bearbeiten möchten (wählen Sie den Clusternamen aus, nicht die Schaltfläche links davon).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie im Abschnitt Scaling policies (Skalierungsrichtlinien) die Auto-Scaling-Richtlinie und dann Delete (Löschen) aus.

Löschen einer Skalierungsrichtlinie mit dem AWS CLI

Um eine Skalierungsrichtlinie ElastiCache für Ihren OSS-Cluster für Valkey und Redis zu löschen, verwenden Sie den [delete-scaling-policy](#) AWS CLI Befehl mit den folgenden Parametern:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--resource-id` — Die Ressourcen-ID. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des Clusters. `replication-group/myscalablecluster`
- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.

Im folgenden Beispiel löschen Sie eine Skalierungsrichtlinie für die Zielverfolgung mit dem Namen `myscalablepolicy` aus einem Cluster mit dem Namen `myscalablecluster`

Für Linux, macOS oder Unix:

```
aws application-autoscaling delete-scaling-policy \  
    --policy-name myscalablepolicy \  
    --resource-id replication-group/myscalablecluster
```

```
--resource-id replication-group/myscalablecluster \  
--service-namespace elasticache \  
--scalable-dimension elasticache:replication-group:NodeGroups
```

Für Windows:

```
aws application-autoscaling delete-scaling-policy ^  
  --policy-name myscalablepolicy ^  
  --resource-id replication-group/myscalablecluster ^  
  --service-namespace elasticache ^  
  --scalable-dimension elasticache:replication-group:NodeGroups
```

Löschen einer Skalierungsrichtlinie über die API

Verwenden Sie den [DeleteScalingPolicy](#) AWS CLI Befehl mit den folgenden Parametern, um eine Skalierungsrichtlinie ElastiCache für Ihren OSS-Cluster für Valkey und Redis zu löschen:

- `—policy-name`– Der Name der Skalierungsrichtlinie.
- `--resource-id` — Die Ressourcen-ID. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des Clusters. `replication-group/myscalablecluster`
- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.

Im folgenden Beispiel löschen Sie eine Skalierungsrichtlinie für die Zielverfolgung mit dem Namen `myscalablepolicy` aus einem Cluster mit dem Namen `myscalablecluster`

```
POST / HTTP/1.1  
Host: autoscaling.us-east-2.amazonaws.com  
Accept-Encoding: identity  
Content-Length: 219  
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy  
X-Amz-Date: 20160506T182145Z  
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8  
Content-Type: application/x-amz-json-1.1  
Authorization: AUTHPARAMS  
{  
  "PolicyName": "myscalablepolicy",  
  "ServiceNamespace": "elasticache",
```

```

    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:NodeGroups"
  }

```

CloudFormation Für Auto Scaling Scaling-Richtlinien verwenden

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie zur Zielverfolgung erstellen und sie auf eine [AWS::ElastiCache::ReplicationGroup](#) Ressource anwenden, die die [AWS::ApplicationAutoScaling::ScalableTarget](#) Ressource verwendet. Sie verwendet die intrinsischen Funktionen [Fn::Join](#) und [Ref](#), um die ResourceId-Eigenschaft mit dem logischen Namen der `AWS::ElastiCache::ReplicationGroup`-Ressource zu erstellen, die in derselben Vorlage angegeben ist.

```

ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 3
    MinCapacity: 1
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
  Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
  Properties:
    ScalingTargetId: !Ref ScalingTarget
    ServiceNamespace: elasticache
    PolicyName: testpolicy
    PolicyType: TargetTrackingScaling
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    TargetTrackingScalingPolicyConfiguration:
      PredefinedMetricSpecification:
        PredefinedMetricType: ElastiCachePrimaryEngineCPUUtilization
      TargetValue: 40

```

Geplante Skalierung

Eine Skalierung anhand eines Zeitplans ermöglicht es Ihnen, Ihre Anwendung entsprechend vorhersagbarer Anforderungsänderungen zu skalieren. Um die geplante Skalierung zu

verwenden, erstellen Sie geplante Aktionen, die Valkey und Redis OSS anweisen ElastiCache , Skalierungsaktivitäten zu bestimmten Zeiten durchzuführen. Wenn Sie eine geplante Aktion erstellen, geben Sie einen vorhandenen Cluster an, wann die Skalierungsaktivität stattfinden soll, sowie die Mindestkapazität und die maximale Kapazität. Sie können geplante Aktionen erstellen, die nur einmal skalieren oder wiederholt geplant ausgeführt werden.

Sie können eine geplante Aktion nur für Cluster erstellen, die bereits vorhanden sind. Sie können eine geplante Aktion und einen Cluster nicht gleichzeitig erstellen.

Weitere Informationen zur Terminologie für die Erstellung, Verwaltung und Löschung geplanter Aktionen finden Sie unter [Häufig verwendete Befehle für geplante Aktionen erstellen, verwalten und löschen](#)

So erstellen Sie einen sich wiederholenden Zeitplan:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS aus.
3. Wählen Sie den DB-Cluster aus, zu dem Sie die Richtlinie hinzufügen möchten.
4. Wählen Sie das SymbolVerwalten der Auto Scaling RichtlinieausAktionenDropdown erstellen.
5. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
6. Im Abschnitt Auto-Scaling-Richtlinien wird Skalierungsrichtlinie hinzufügen angezeigt. Klicken Sie auf Geplante Skalierung.
7. Geben Sie unter Policy Name (Richtliniennamen) einen Namen für diese Richtlinie ein.
8. Für Skalierbare Dimension wählen Sie Shards aus.
9. Wählen Sie für Ziel-Shards den Wert aus.
10. Für Wiederholung, wählen Sie Wiederkehrende aus.
11. Für Häufigkeit wählen Sie den entsprechenden Wert aus.
12. FürStartdatumundBeginnzeitwählen Sie den Zeitpunkt aus, ab dem die Richtlinie wirksam wird.
13. Wählen Sie Add Policy (Richtlinie hinzufügen) aus.

Erstellen Sie eine einmalige geplante Aktion wie folgt:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.

2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS aus.
3. Wählen Sie den DB-Cluster aus, zu dem Sie die Richtlinie hinzufügen möchten.
4. Wählen Sie das SymbolVerwalten der Auto Scaling RichtlinieausAktionenDropdown erstellen.
5. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
6. Im Abschnitt Auto-Scaling-Richtlinien wird Skalierungsrichtlinie hinzufügen angezeigt. Klicken Sie auf Geplante Skalierung.
7. Geben Sie unter Policy Name (Richtliniennamen) einen Namen für diese Richtlinie ein.
8. Für Skalierbare Dimension wählen Sie Shards aus.
9. Wählen Sie für Ziel-Shards den Wert aus.
10. Wählen Sie für Wiederholung Einmalig aus.
11. FürStartdatumundBeginnzeitwählen Sie den Zeitpunkt aus, ab dem die Richtlinie wirksam wird.
12. FürEnddatumwählen Sie das Datum, bis zu dem die Richtlinie wirksam sein würde.
13. Wählen Sie Add Policy (Richtlinie hinzufügen) aus.

Löschen einer geplanten Aktion

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS aus.
3. Wählen Sie den DB-Cluster aus, zu dem Sie die Richtlinie hinzufügen möchten.
4. Wählen Sie das SymbolVerwalten der Auto Scaling RichtlinieausAktionenDropdown erstellen.
5. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
6. Wählen Sie im Abschnitt Auto-Scaling-Richtlinien die Auto-Scaling-Richtlinie und dann Löschen im Aktionen-Dialog aus.

Verwalten einer geplanten Skalierung mit AWS CLI

Verwenden Sie die folgende automatische Anwendungsskalierung: APIs

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Verwenden Sie CloudFormation für die Erstellung einer geplanten Aktion

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie zur Zielverfolgung erstellen und sie auf eine [AWS::ElastiCache::ReplicationGroup](#) Ressource anwenden, die die [AWS::ApplicationAutoScaling::ScalableTarget](#) Ressource verwendet. Sie verwendet die intrinsischen Funktionen [Fn::Join](#) und [Ref](#), um die ResourceId-Eigenschaft mit dem logischen Namen der AWS::ElastiCache::ReplicationGroup-Ressource zu erstellen, die in derselben Vorlage angegeben ist.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 3
    MinCapacity: 1
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
    ScheduledActions:
      - EndTime: '2020-12-31T12:00:00.000Z'
        ScalableTargetAction:
          MaxCapacity: '5'
          MinCapacity: '2'
        ScheduledActionName: First
        Schedule: 'cron(0 18 * * ? *)'
```

Verwenden von Auto Scaling mit Replikaten

Eine ElastiCache Replikationsgruppe kann einen oder mehrere Caches so einrichten, dass sie als einzelner logischer Knoten funktionieren.

Im Folgenden finden Sie Einzelheiten zur Zielverfolgung und zu geplanten Richtlinien sowie deren Anwendung mithilfe von AWS-Managementkonsole AWS CLI und APIs.

Skalierungsrichtlinien für die Ziel-Nachverfolgung

Bei Skalierungsrichtlinien für die Zielverfolgung wählen Sie eine Metrik aus und legen einen Zielwert fest. ElastiCache für Valkey und Redis AutoScaling erstellt und verwaltet OSS die CloudWatch Alarmer, die die Skalierungsrichtlinie auslösen, und berechnet die Skalierungsanpassung auf der

Grundlage der Metrik und des Zielwerts. Die Skalierungsrichtlinie fügt -Repliken hinzu oder entfernt sie, wenn erforderlich, um die Metrik auf dem angegebenen Zielwert oder in der Nähe davon zu halten. Abgesehen davon, dass eine Skalierungsrichtlinie für die Ziel-Nachverfolgung die Metrik nahe an dem Zielwert hält, passt sie sich auch an die Schwankungen in der Metrik aufgrund eines schwankenden Lastmusters an und verringert schnelle Schwankungen der Kapazität der Flotte.

Kriterien für die Auto Scaling von Replikaten

Ihre Auto Scaling Richtlinie definiert die folgende vordefinierte Metrik für Ihren Cluster:

ElastiCacheReplicaEngineCPUUtilization: Der Schwellenwert für die CPU-Auslastung von AVG Engine, aggregiert für alle Replikate, der ElastiCache verwendet wird, um einen auto-scaling Skalierungsvorgang auszulösen. Sie können das Auslastungsziel zwischen 35 Prozent und 70 Prozent festlegen.

Wenn der Service feststellt, dass Ihre **ElastiCacheReplicaEngineCPUUtilization** Metrik gleich oder größer als die Zieleinstellung ist, erhöht er automatisch die Anzahl der Replikate auf Ihren Shards. ElastiCache skaliert Ihre Cluster-Replikate um eine Anzahl, die der größeren von zwei Zahlen entspricht: prozentuale Abweichung von Target und ein Replikat. Bei Scale-In erfolgt ElastiCache keine auto Skalierung, es sei denn, der Gesamtmetrikwert liegt unter 75 Prozent Ihres definierten Ziels.

Wenn Sie für ein Scale-Out-Beispiel jeweils 5 Shards und 1 Replikat haben:

Wenn Ihr Target die Sicherheitslücken um 30 Prozent überschreitet, skaliert OSS ElastiCache für Valkey und Redis über alle Shards hinweg um 1 Replikat (max. 0,3, Standard 1). Das Ergebnis sind 5 Shards mit jeweils 2 Replikaten.

Wenn Sie beispielsweise einen Zielwert von 60 Prozent ausgewählt haben, wird OSS ElastiCache für Valkey und Redis erst auto skalieren, wenn die Metrik 45 Prozent oder weniger beträgt (25 Prozent unter dem Ziel von 60 Prozent).

Überlegungen zum Auto Scaling

Beachten Sie folgende Überlegungen:

- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung geht davon aus, dass sie eine horizontale Skalierung nach oben vornehmen soll, wenn die angegebene Metrik über dem Zielwert liegt. Sie können keine Skalierungsrichtlinie für die Zielverfolgung verwenden, um eine Skalierung vorzunehmen, wenn die angegebene Metrik unter dem Zielwert liegt. ElastiCache für Valkey

und Redis skaliert OSS Replikate nach dem Maximum (% Abweichung abgerundet von Target, Standard 1) der vorhandenen Replikate auf allen Shards im Cluster.

- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung nimmt keine Skalierung vor, wenn die angegebene Metrik unzureichende Daten aufweist. Es wird keine horizontale Skalierung nach unten vorgenommen, da unzureichende Daten nicht als geringe Auslastung interpretiert werden.
- Möglicherweise werden Lücken zwischen den Datenpunkten für den Zielwert und die aktuelle Metrik angezeigt. Das liegt daran, dass ElastiCache Auto Scaling immer konservativ agiert, indem es auf- oder abrundet, wenn es bestimmt, wie viel Kapazität hinzugefügt oder entfernt werden soll. Dadurch wird verhindert, dass zu wenig Kapazität hinzufügt oder zu viel Kapazität entfernt wird.
- Um die Verfügbarkeit der Anwendung sicherzustellen, wird der Service schnellstmöglich proportional zur Metrik hochskaliert, jedoch etwas langsamer herunterskaliert.
- Sie können mehrere Skalierungsrichtlinien zur Zielverfolgung für einen OSS-Cluster ElastiCache für Valkey und Redis verwenden, vorausgesetzt, dass jeder von ihnen eine andere Metrik verwendet. Die Absicht von Auto Scaling besteht darin, der Verfügbarkeit immer Priorität einzuräumen. Daher unterscheidet sich das Verhalten je nachdem, ob die Zielverfolgungsrichtlinien für die horizontale Skalierung oder für die Skalierung bereit sind. Sofern Richtlinien für die Ziel-Nachverfolgung für die Skalierung nach oben bereit sind, findet eine Skalierung des Service nach oben statt. Eine Skalierung nach unten wird jedoch nur vorgenommen, wenn alle Richtlinien für die Ziel-Nachverfolgung (mit aktivierter Skalierung nach unten) zur Skalierung nach unten bereit sind.
- Bearbeiten oder löschen Sie nicht die CloudWatch Alarmer, die ElastiCache Auto Scaling für eine Skalierungsrichtlinie zur Zielverfolgung verwaltet. Auto Scaling löscht die Alarmer automatisch, wenn Sie die Skalierungsrichtlinie oder den Cluster löschen.
- ElastiCache Auto Scaling hindert Sie nicht daran, Replikate auf mehreren Shards manuell zu ändern. Diese manuellen Anpassungen wirken sich nicht auf bestehende CloudWatch Alarmer aus, die mit der Skalierungsrichtlinie verknüpft sind, können sich jedoch auf Metriken auswirken, die diese CloudWatch Alarmer auslösen können.
- Diese von Auto Scaling verwalteten CloudWatch Alarmer werden anhand der AVG-Metrik für alle Shards im Cluster definiert. So kann Hot-Shards zu einem beliebigen Szenario führen:
 - Skalierung, wenn sie nicht erforderlich ist, weil die Belastung einiger Hot-Shards einen Alarm auslöst CloudWatch
 - nicht skalieren, wenn dies erforderlich ist, aufgrund aggregierter AVG über alle Shards, die den Alarm nicht verletzen.
- ElastiCache Die Standardgrenzwerte für Knoten pro Cluster gelten weiterhin. Wenn Sie sich also für Auto Scaling entscheiden und erwarten, dass maximale Knoten mehr als die Standardgrenze

sind, fordern Sie eine Limiterhöhung bei [AWS -Service-Limits](#) an und wählen Sie den Limit-Typ Knoten pro Cluster pro Instance-Typ.

- Stellen Sie sicher, dass in Ihrer VPC genügend ENIs (Elastic Network Interfaces) verfügbar sind, die beim Scale-Out benötigt werden. Weitere Informationen finden Sie unter [Elastic-Netzwerk-Schnittstellen](#).
- Wenn nicht genügend Kapazität verfügbar ist EC2, wird ElastiCache Auto Scaling erst skaliert, wenn entweder die Kapazität verfügbar ist oder wenn Sie den Cluster manuell auf die Instance-Typen mit ausreichender Kapazität ändern.
- ElastiCache Auto Scaling unterstützt keine Skalierung von Replikaten mit einem Cluster mit `ReservedMemoryPercent` weniger als 25 Prozent. Weitere Informationen finden Sie unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

Hinzufügen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie hinzufügen, indem Sie die verwenden AWS-Managementkonsole.

Hinzufügen einer Skalierungsrichtlinie mithilfe des AWS-Managementkonsole

Um eine Auto Scaling-Richtlinie ElastiCache für Valkey und Redis OSS hinzuzufügen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS aus.
3. Wählen Sie den Cluster aus, dem Sie eine Richtlinie hinzufügen möchten (wählen Sie den Cluster-Namen und nicht die Schaltfläche links davon aus).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie Add dynamic scaling (Dynamische Skalierung hinzufügen) aus.
6. Wählen Sie unter Scaling policies (Skalierungsrichtlinien) Add dynamic scaling (Dynamische Skalierung hinzufügen) aus.
7. Geben Sie unter Policy Name (Richtliniennamen) einen Namen für diese Richtlinie ein.
8. FürSkalierbare Dimensionauf und danachReplicas-Dialogfeld aus erstellen.
9. Geben Sie als Zielwert den durchschnittlichen Prozentsatz der CPU-Auslastung ein, den Sie auf ElastiCache Replicas beibehalten möchten. Dieser Wert muss größer 35 und kleiner 70 sein. - Repliken werden hinzugefügt oder entfernt, um die Metrik in der Nähe des angegebenen Wertes zu halten.

10. (Optional) Die Ruhephase für Scale-In- oder Scale-Out-Ruhephase werden von der Konsole nicht unterstützt. Verwenden Sie den AWS CLI , um die Abkühlwerte zu ändern.
11. Geben Sie für Mindestkapazität die Mindestanzahl von Replikaten ein, für deren Verwaltung die ElastiCache Auto Scaling Scaling-Richtlinie erforderlich ist.
12. Geben Sie für Maximale Kapazität die maximale Anzahl von Replikaten ein, die die ElastiCache Auto Scaling Scaling-Richtlinie verwalten muss. Der Wert muss größer als 5 sein.
13. Wählen Sie Erstellen aus.

Registrieren eines skalierbaren Ziels

Sie können eine Skalierungsrichtlinie anwenden, die entweder auf einer vordefinierten oder einer benutzerdefinierten Metrik basiert. Dazu können Sie die AWS CLI oder die Application Auto Scaling API verwenden. Der erste Schritt besteht darin, Ihre OSS-Replikationsgruppe ElastiCache für Valkey und Redis bei Auto Scaling zu registrieren.

Bevor Sie ElastiCache Auto Scaling mit einem Cluster verwenden können, müssen Sie Ihren Cluster bei ElastiCache Auto Scaling registrieren. Sie tun dies, um die Skalierungsdimension und die Grenzwerte zu definieren, die auf diesen Cluster angewendet werden sollen. ElastiCache Auto Scaling skaliert den Cluster dynamisch entlang der `elasticache:replication-group:Replicas` skalierbaren Dimension, die die Anzahl der Cluster-Replikate pro Shard darstellt.

Verwenden der CLI

Verwenden Sie den [register-scalable-target](#) Befehl mit den folgenden Parametern, um Ihren ElastiCache Cluster zu registrieren:

- `—service-namespace` — Legen Sie diesen Wert auf `elasticache` fest.
- `--resource-id` — Die Ressourcen-ID für den Cluster. ElastiCache Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des Clusters. `replication-group/myscalablecluster`
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.
- `--min-capacity` — Die Mindestanzahl von Replikaten, die mit Auto Scaling verwaltet werden sollen. ElastiCache Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

- `--max-capacity` — Die maximale Anzahl von Replikaten, die mit Auto Scaling verwaltet werden können. ElastiCache Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

Example

Im folgenden Beispiel registrieren Sie einen Cluster mit dem Namen `myscalablecluster`. Die Registrierung zeigt an, dass der Cluster dynamisch skaliert werden soll, um zwischen einem und 5 Replikaten zu haben.

Für Linux, macOS oder Unix:

```
aws application-autoscaling register-scalable-target \
  --service-namespace elasticache \
  --resource-id replication-group/myscalablecluster \
  --scalable-dimension elasticache:replication-group:Replicas \
  --min-capacity 1 \
  --max-capacity 5 \
```

Für Windows:

```
aws application-autoscaling register-scalable-target ^
  --service-namespace elasticache ^
  --resource-id replication-group/myscalablecluster ^
  --scalable-dimension elasticache:replication-group:Replicas ^
  --min-capacity 1 ^
  --max-capacity 5 ^
```

Verwenden der API

Verwenden Sie den [register-scalable-target](#) Befehl mit den folgenden Parametern, um Ihren ElastiCache Cluster zu registrieren:

- `ServiceNamespace` — Setzen Sie diesen Wert auf `elasticache`.
- `ResourceID` — Die Ressourcen-ID für den ElastiCache Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise `replication-group/myscalablecluster` der Name des Clusters.
- `ScalableDimension` — Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.

- **MinCapacity** — Die Mindestanzahl von Replikaten, die mit ElastiCache Auto Scaling verwaltet werden müssen. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).
- **MaxCapacity** — Die maximale Anzahl von Replikaten, die mit ElastiCache Auto Scaling verwaltet werden können. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

Example

Im folgenden Beispiel registrieren Sie einen Cluster, der `myscalablecluster` mit der Application Auto Scaling API benannt ist. Diese Registrierung zeigt an, dass der Cluster dynamisch skaliert werden soll, um zwischen 1 und 5 Replikate zu haben.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definieren einer Skalierungsrichtlinie

Eine Konfiguration der Skalierungsrichtlinie für die Zielverfolgung wird durch einen JSON-Block repräsentiert, in dem die Metriken und Zielwerte definiert sind. Sie können die Konfiguration einer Skalierungsrichtlinie als JSON-Block in einer Textdatei speichern. Sie verwenden diese Textdatei, wenn Sie die AWS CLI oder die Application Auto Scaling Scaling-API aufrufen. Weitere Informationen zur Syntax der Richtlinienkonfiguration finden Sie unter [TargetTrackingScalingPolicyConfiguration](#) in der API-Referenz für Application Auto Scaling.

Die folgenden Optionen stehen zur Verfügung, um eine Konfiguration der Skalierungsrichtlinien für die Zielverfolgung zu definieren:

Themen

- [Verwenden einer vordefinierten Metrik](#)
- [Bearbeiten einer Skalierungsrichtlinie](#)
- [Löschen einer Skalierungsrichtlinie](#)
- [CloudFormation Für Auto Scaling Scaling-Richtlinien verwenden](#)
- [Geplante Skalierung](#)

Verwenden einer vordefinierten Metrik

Eine Konfiguration der Skalierungsrichtlinie für die Zielverfolgung wird durch einen JSON-Block repräsentiert, in dem die Metriken und Zielwerte definiert sind. Sie können die Konfiguration einer Skalierungsrichtlinie als JSON-Block in einer Textdatei speichern. Sie verwenden diese Textdatei, wenn Sie die AWS CLI oder die Application Auto Scaling Scaling-API aufrufen. Weitere Informationen zur Syntax der Richtlinienkonfiguration finden Sie unter [TargetTrackingScalingPolicyConfiguration](#) in der API-Referenz für Application Auto Scaling.

Die folgenden Optionen stehen zur Verfügung, um eine Konfiguration der Skalierungsrichtlinien für die Zielverfolgung zu definieren:

Themen

- [Verwenden einer vordefinierten Metrik](#)
- [Verwenden einer benutzerdefinierten Metrik](#)
- [Verwenden von Ruhephasen](#)
- [Deaktivieren der Scale-In-Aktivität](#)
- [Anwendung einer Skalierungsrichtlinie auf einen ElastiCache OSS-Cluster für Valkey und Redis](#)

Verwenden einer vordefinierten Metrik

Mithilfe vordefinierter Metriken können Sie schnell eine Skalierungsrichtlinie für die Zielverfolgung für einen OSS-Cluster ElastiCache für Valkey und Redis definieren, der mit der Zielverfolgung in Auto Scaling funktioniert. ElastiCache ElastiCache Unterstützt derzeit die folgende vordefinierte Metrik in ElastiCache Replicas Auto Scaling:

ElastiCacheReplicaEngineCPUUtilization— Der Durchschnittswert der CPUUtilization Engine-Metrik CloudWatch für alle Replikate im Cluster. Den aggregierten Metrikwert finden Sie CloudWatch unter „Erforderlich“ ReplicationGroupId und „ElastiCache ReplicationGroupId, RoleRollenreplikat“.

Um eine vordefinierte Metrik in Ihrer Skalierungsrichtlinie zu verwenden, erstellen Sie eine Zielverfolgungskonfiguration für Ihre Skalierungsrichtlinie. Diese Konfiguration muss eine **PredefinedMetricSpecification** für die vordefinierte Metrik und einen **TargetValue** für den Zielwert dieser Metrik enthalten.

Verwenden einer benutzerdefinierten Metrik

Durch die Verwendung von benutzerdefinierten Metriken können Sie eine Skalierungsrichtlinie für die Zielverfolgung definieren, die Ihren individuellen Anforderungen entspricht. Sie können eine benutzerdefinierte Metrik definieren, die auf einer beliebigen OSS-Metrik ElastiCache für Valkey und Redis basiert und sich proportional zur Skalierung ändert. Nicht alle ElastiCache Metriken eignen sich für die Zielverfolgung. Die Metrik muss eine gültige Auslastungsmetrik sein und beschreiben, wie ausgelastet eine Instance ist. Der Wert der Metrik muss sich proportional zur Anzahl der -Replicas im -DB-Cluster erhöhen oder verringern. Diese proportionale Erhöhung oder Verminderung ist notwendig, um die metrischen Daten zur proportionalen Skalierung oder in der Anzahl der -Repliken zu verwenden.

Example

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt eine benutzerdefinierte Metrik einen Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 50 Prozent für alle Replikate in einem Cluster mit dem Namen `an.my-db-cluster`

```
{
  "TargetValue": 50,
  "CustomizedMetricSpecification": {
    "MetricName": "EngineCPUUtilization",
    "Namespace": "AWS/ElastiCache",
    "Dimensions": [
      { "Name": "ReplicationGroup", "Value": "my-db-cluster" },
      { "Name": "Role", "Value": "REPLICA" }
    ],
    "Statistic": "Average",
    "Unit": "Percent"
  }
}
```

```
}
```

Verwenden von Ruhephasen

Sie können einen Wert in Sekunden für `ScaleOutCooldown` angeben, um eine Ruhephase für die Aufskalierung Ihres Clusters hinzuzufügen. Ähnlich können Sie einen Wert in Sekunden für `ScaleInCooldown` angeben, um eine Ruhephase für die Abskalierung Ihres Clusters hinzuzufügen. Weitere Informationen über `ScaleInCooldown` und `ScaleOutCooldown` finden Sie unter [TargetTrackingScalingPolicyConfiguration](#) in der API-Referenz für Application Auto Scaling. Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration wird die `ElastiCacheReplicaEngineCPUUtilization` vordefinierte Metrik verwendet, um einen Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 40 Prozent für alle Replikate in diesem Cluster anzupassen. Die Konfiguration sieht eine Scale-In-Ruhephase von 10 Minuten und eine Scale-Out-Ruhephase von 5 Minuten vor.

```
{"TargetValue": 40.0,
  "PredefinedMetricSpecification":
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"},
  "ScaleInCooldown": 600,
  "ScaleOutCooldown": 300
}
```

Deaktivieren der Scale-In-Aktivität

Sie können verhindern, dass die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung in Ihrem OSS-Cluster ElastiCache für Valkey und Redis skaliert, indem Sie die Scale-In-Aktivität deaktivieren. Das Deaktivieren der Scale-In-Aktivität verhindert das Löschen von -Repliken durch die Skalierungsrichtlinie, und erlaubt der Skalierungsrichtlinie dennoch, Repliken nach Bedarf zu erstellen.

Sie können einen booleschen Wert für `DisableScaleIn` angeben, um die Scale-In-Aktivität für Ihren Cluster zu aktivieren oder zu deaktivieren. Weitere Informationen über `DisableScaleIn` finden Sie unter [TargetTrackingScalingPolicyConfiguration](#) in der API-Referenz für Application Auto Scaling.

Example

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt die `ElastiCacheReplicaEngineCPUUtilization` vordefinierte

Metrik einen Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 40 Prozent für alle Replikate in diesem Cluster an. Die Konfiguration deaktiviert die Scale-In-Aktivität für die Skalierungsrichtlinie.

```
{"TargetValue": 40.0,
  "PredefinedMetricSpecification":
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    },
  "DisableScaleIn": true
}
```

Anwendung einer Skalierungsrichtlinie auf einen ElastiCache OSS-Cluster für Valkey und Redis

Nachdem Sie Ihren Cluster bei ElastiCache for Valkey und Redis OSS Auto Scaling registriert und eine Skalierungsrichtlinie definiert haben, wenden Sie die Skalierungsrichtlinie auf den registrierten Cluster an. Um eine Skalierungsrichtlinie auf einen OSS-Cluster ElastiCache für Valkey und Redis anzuwenden, können Sie die AWS CLI oder die Application Auto Scaling Scaling-API verwenden.

Unter Verwendung der AWS CLI

Verwenden Sie den [put-scaling-policy](#) Befehl mit den folgenden Parametern, um eine Skalierungsrichtlinie auf Ihren OSS-Cluster ElastiCache für Valkey und Redis anzuwenden:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--policy-type` – Setzen Sie diesen Wert auf `TargetTrackingScaling`.
- `--resource-id` — Die Ressourcen-ID für den Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des Clusters.
`replication-group/myscalablecluster`
- `—service-namespace` — Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.
- `--target-tracking-scaling-policy-configuration` — Die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung, die für den Cluster verwendet werden soll.

Example

Im folgenden Beispiel wenden Sie eine Zielverfolgungs-Skalierungsrichtlinie mit dem Namen auf einen Cluster `myscalablepolicy` an, der `myscalablecluster` mit ElastiCache Auto Scaling

benannt ist. Dazu verwenden Sie eine Richtlinienkonfiguration, die in einer Datei mit dem Namen `config.json` gespeichert ist.

Für Linux, macOS oder Unix:

```
aws application-autoscaling put-scaling-policy \  
  --policy-name myscalablepolicy \  
  --policy-type TargetTrackingScaling \  
  --resource-id replication-group/myscalablecluster \  
  --service-namespace elasticache \  
  --scalable-dimension elasticache:replication-group:Replicas \  
  --target-tracking-scaling-policy-configuration file://config.json
```

```
{"TargetValue": 40.0,  
  "PredefinedMetricSpecification":  
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"  
    },  
  "DisableScaleIn": true  
}
```

Für Windows:

```
aws application-autoscaling put-scaling-policy ^  
  --policy-name myscalablepolicy ^  
  --policy-type TargetTrackingScaling ^  
  --resource-id replication-group/myscalablecluster ^  
  --service-namespace elasticache ^  
  --scalable-dimension elasticache:replication-group:Replicas ^  
  --target-tracking-scaling-policy-configuration file://config.json
```

Verwenden der API

Um mit der Application Auto Scaling API eine Skalierungsrichtlinie auf Ihren ElastiCache Cluster anzuwenden, verwenden Sie den [PutScalingPolicy](#) Application Auto Scaling API-Vorgang mit den folgenden Parametern:

- **PolicyName** — Der Name der Skalierungsrichtlinie.
- **PolicyType** — Setzen Sie diesen Wert auf `TargetTrackingScaling`.

- **ResourceID** — Die Ressourcen-ID für den Cluster. Für diesen Parameter ist der Ressourcentyp **ReplicationGroup** und der eindeutige Bezeichner beispielsweise der Name des ElastiCache für Redis OSS-Clusters. **replication-group/myscalablecluster**
- **ServiceNamespace** — Setzen Sie diesen Wert auf **Elasticache**.
- **ScalableDimension** — Setze diesen Wert auf **elasticache:replication-group:Replicas**.
- **TargetTrackingScalingPolicyConfiguration** — Die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung, die für den Cluster verwendet werden soll.

Example

Im folgenden Beispiel wenden Sie eine Zielverfolgungs-Skalierungsrichtlinie mit dem Namen auf einen Cluster **scalablepolicy** an, der **myscalablecluster** mit ElastiCache Auto Scaling benannt ist. Sie verwenden die Richtlinienkonfiguration **ElastiCacheReplicaEngineCPUUtilization** basierend auf einer vordefinierten Metrik.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {
      "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    }
  }
}
```

Bearbeiten einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie mit der AWS-Managementkonsole, der oder der AWS CLI Application Auto Scaling API bearbeiten.

Bearbeiten einer Skalierungsrichtlinie mit dem AWS-Managementkonsole

Sie können Richtlinien mit dem Typ Vordefinierte Metriken nur bearbeiten, indem Sie die AWS-Managementkonsole

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS
3. Wählen Sie den Cluster aus, dem Sie eine Richtlinie hinzufügen möchten (wählen Sie den Cluster-Namen und nicht die Schaltfläche links davon aus).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie unter Scaling policies (Skalierungsrichtlinien) die Schaltfläche links neben der Auto-Scaling-Richtlinie aus, die Sie ändern möchten, und klicken Sie dann auf Modify (Ändern).
6. Nehmen Sie die erforderlichen Änderungen an der Richtlinie vor.
7. Wählen Sie Ändern aus.
8. Nehmen Sie die Änderungen an der Richtlinie vor.
9. Wählen Sie Ändern aus.

Bearbeiten einer Skalierungsrichtlinie mit der AWS CLI oder der Application Auto Scaling API

Sie können die AWS CLI oder die Application Auto Scaling Scaling-API verwenden, um eine Skalierungsrichtlinie auf die gleiche Weise zu bearbeiten, wie Sie eine Skalierungsrichtlinie anwenden:

- Wenn Sie die Application Auto Scaling API verwenden, geben Sie den Namen der zu bearbeitenden Richtlinie im Parameter `PolicyName` an. Legen Sie neue Werte für die Parameter fest, die Sie ändern möchten.

Weitere Informationen finden Sie unter [Anwendung einer Skalierungsrichtlinie auf einen ElastiCache OSS-Cluster für Valkey und Redis](#).

Löschen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie mit der AWS-Managementkonsole, der AWS CLI oder der Application Auto Scaling API löschen

Löschen einer Skalierungsrichtlinie mithilfe der AWS-Managementkonsole

Sie können Richtlinien mit dem Typ Vordefinierte Metriken nur bearbeiten, indem Sie die AWS-Managementkonsole

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS
3. Wählen Sie den Cluster aus, dessen Auto-Scaling-Richtlinie Sie löschen möchten.
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie im Abschnitt Scaling policies (Skalierungsrichtlinien) die Auto-Scaling-Richtlinie und dann Delete (Löschen) aus.

Löschen einer Skalierungsrichtlinie mithilfe der AWS CLI oder der Application Auto Scaling API

Sie können die AWS CLI oder die Application Auto Scaling API verwenden, um eine Skalierungsrichtlinie aus einem ElastiCache Cluster zu löschen.

CLI

Verwenden Sie den [delete-scaling-policy](#)Befehl mit den folgenden Parametern, um eine Skalierungsrichtlinie aus Ihrem OSS-Cluster ElastiCache für Valkey und Redis zu löschen:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--resource-id` — Die Ressourcen-ID für den Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des Clusters.
`replication-group/myscalablecluster`
- `—service-namespace` — Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.

Example

Im folgenden Beispiel löschen Sie eine Zielverfolgungs-Skalierungsrichtlinie namens `myscalablepolicy` aus einem ElastiCache-Cluster namens `myscalablecluster`.

Für Linux, macOS oder Unix:

```
aws application-autoscaling delete-scaling-policy \
  --policy-name myscalablepolicy \
  --resource-id replication-group/myscalablecluster \
  --service-namespace elasticache \
  --scalable-dimension elasticache:replication-group:Replicas \
```

Für Windows:

```
aws application-autoscaling delete-scaling-policy ^
  --policy-name myscalablepolicy ^
  --resource-id replication-group/myscalablecluster ^
  --service-namespace elasticache ^
  --scalable-dimension elasticache:replication-group:Replicas ^
```

API

Um eine Skalierungsrichtlinie aus Ihrem ElastiCache für Valkey und Redis OSS-Cluster zu löschen, verwenden Sie den [DeleteScalingPolicy](#) Application Auto Scaling API-Vorgang mit den folgenden Parametern:

- **PolicyName** — Der Name der Skalierungsrichtlinie.
- **ResourceId** — Die ResourceID für den Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise `replication-group/myscalablecluster` der Name des Clusters.
- **ServiceNamespace** — Setzen Sie diesen Wert auf `Elasticache`.
- **ScalableDimension** — Setze diesen Wert auf `elasticache:replication-group:Replicas`.

Im folgenden Beispiel löschen Sie eine Skalierungsrichtlinie für die Zielverfolgung, die `myscalablepolicy` aus einem Cluster benannt ist, der `myscalablecluster` mit der Application Auto Scaling Scaling-API benannt ist.

```

POST / HTTP/1.1
>>>>>> mainline
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas"
}

```

CloudFormation Für Auto Scaling Scaling-Richtlinien verwenden

Dieser Ausschnitt zeigt, wie Sie eine geplante Aktion erstellen und sie auf eine Ressource anwenden, die die [AWS::ElastiCache::ReplicationGroup](#) [AWS::ApplicationAutoScaling::ScalableTarget](#) Ressource verwendet. Sie verwendet die intrinsischen Funktionen [Fn::Join](#) und [Ref](#), um die `ResourceId`-Eigenschaft mit dem logischen Namen der `AWS::ElastiCache::ReplicationGroup`-Ressource zu erstellen, die in derselben Vorlage angegeben ist.

```

ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 0
    MinCapacity: 0
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:Replicas'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
  Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
  Properties:
    ScalingTargetId: !Ref ScalingTarget
    ServiceNamespace: elasticache

```

```
PolicyName: testpolicy
PolicyType: TargetTrackingScaling
ScalableDimension: 'elasticache:replication-group:Replicas'
TargetTrackingScalingPolicyConfiguration:
  PredefinedMetricSpecification:
    PredefinedMetricType: ElastiCacheReplicaEngineCPUUtilization
  TargetValue: 40
```

Geplante Skalierung

Eine Skalierung anhand eines Zeitplans ermöglicht es Ihnen, Ihre Anwendung entsprechend vorhersagbarer Anforderungsänderungen zu skalieren. Um die geplante Skalierung zu verwenden, erstellen Sie geplante Aktionen, die Valkey und Redis OSS anweisen ElastiCache, Skalierungsaktivitäten zu bestimmten Zeiten durchzuführen. Wenn Sie eine geplante Aktion erstellen, geben Sie einen vorhandenen ElastiCache Cluster an, wann die Skalierungsaktivität stattfinden soll, sowie die Mindestkapazität und die maximale Kapazität. Sie können geplante Aktionen erstellen, die nur einmal skalieren oder wiederholt geplant ausgeführt werden.

Sie können eine geplante Aktion nur für ElastiCache Cluster erstellen, die bereits vorhanden sind. Sie können eine geplante Aktion und einen Cluster nicht gleichzeitig erstellen.

Weitere Informationen zur Terminologie für die Erstellung, Verwaltung und Löschung geplanter Aktionen finden Sie unter [Häufig verwendete Befehle für geplante Aktionen erstellen, verwalten und löschen](#)

Erstellen Sie eine einmalige geplante Aktion wie folgt

Ähnlich wie Shard Dimension. Siehe [Geplante Skalierung](#).

Löschen einer geplanten Aktion

Ähnlich wie Shard Dimension. Siehe [Geplante Skalierung](#).

Verwalten einer geplanten Skalierung mit AWS CLI

Verwenden Sie die folgende automatische Anwendungsskalierung APIs:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Wird verwendet CloudFormation , um Auto Scaling Scaling-Richtlinien zu erstellen

Dieser Ausschnitt zeigt, wie Sie eine geplante Aktion erstellen und sie auf eine Ressource anwenden, die die [AWS::ElastiCache::ReplicationGroup](#)[AWS::ApplicationAutoScaling::ScalableTarget](#) Ressource verwendet. Sie verwendet die intrinsischen Funktionen [Fn::Join](#) und [Ref](#), um die ResourceId-Eigenschaft mit dem logischen Namen der `AWS::ElastiCache::ReplicationGroup`-Ressource zu erstellen, die in derselben Vorlage angegeben ist.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 0
    MinCapacity: 0
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:Replicas'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
    ScheduledActions:
      - EndTime: '2020-12-31T12:00:00.000Z'
        ScalableTargetAction:
          MaxCapacity: '5'
          MinCapacity: '2'
          ScheduledActionName: First
          Schedule: 'cron(0 18 * * ? *)'
```

Ändern des Clustermodus

Valkey und Redis OSS sind verteilte In-Memory-Datenbanken, die Sharding und Replikation unterstützen. ElastiCache Valkey- und Redis OSS-Cluster sind die verteilte Implementierung, mit der Daten auf mehrere Knoten partitioniert werden können. Ein OSS-Cluster ElastiCache für Redis hat zwei Betriebsmodi: den Clustermodus aktiviert (CME) und den Clustermodus deaktiviert (CMD). In CME funktioniert eine Valkey- und Redis-OSS-Engine als verteilte Datenbank mit mehreren Shards und Knoten, während Valkey und Redis OSS in CMD als ein einziger Knoten arbeiten.

Vor der Migration von CMD zu CME müssen die folgenden Bedingungen erfüllt sein:

 Important

Die Konfiguration des Clustermodus kann nur vom deaktivierten in den aktivierten Clustermodus geändert werden. Das Zurücksetzen dieser Konfiguration ist nicht möglich.

- Der Cluster darf nur Schlüssel in der Datenbank 0 haben.
- Anwendungen müssen einen Valkey- oder Redis-OSS-Client verwenden, der das Cluster-Protokoll verwenden kann, und einen Konfigurationsendpunkt verwenden.
- Das automatische Failover muss auf dem Cluster mit mindestens einem Replikat aktiviert sein.
- Die für die Migration erforderliche Engine-Mindestversion ist Valkey 7.2 und höher oder Redis OSS 7.0 und höher.

Um von CMD zu CME zu migrieren, muss die Konfiguration des Clustermodus vom deaktivierten in den aktivierten Clustermodus geändert werden. Dies ist ein zweistufiges Verfahren, das die Clusterverfügbarkeit während des Migrationsprozesses sicherstellt.

 Note

Sie müssen eine Parametergruppe mit clusterfähiger Konfiguration angeben, d. h., der clusterfähige Parameter ist auf yes festgelegt. Wenn Sie eine Standardparametergruppe verwenden, wählt OSS ElastiCache für Redis automatisch die entsprechende Standardparametergruppe mit einer clusterfähigen Konfiguration aus. Der clusterfähige Parameterwert ist für einen CMD-Cluster auf no festgelegt. Wenn der Cluster in den kompatiblen Modus wechselt, wird der clusterfähige Parameterwert im Rahmen der Änderungsaktion auf yes aktualisiert.

Weitere Informationen finden Sie unter [Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen](#).

1. Vorbereitung — Erstellen Sie einen CME-Test-CME-Cluster und stellen Sie sicher, dass Ihr Stack bereit ist, damit zu arbeiten. ElastiCache für Redis OSS gibt es keine Möglichkeit, Ihre Bereitschaft zu überprüfen. Weitere Informationen finden Sie unter [Einen Cluster für Valkey oder Redis OSS erstellen](#).
2. Ändern Sie die bestehende CMD-Clusterkonfiguration so, dass sie mit dem Clustermodus kompatibel ist — In diesem Modus wird ein einziger Shard bereitgestellt, und ElastiCache für

Redis funktioniert OSS als einzelner Knoten, aber auch als einzelner Shard-Cluster. Kompatibler Modus bedeutet, dass die Client-Anwendung eines der beiden Protokolle verwenden kann, um mit dem Cluster zu kommunizieren. In diesem Modus müssen Anwendungen neu konfiguriert werden, damit sie das Valkey- oder Redis OSS-Cluster-Protokoll und den Konfigurationsendpunkt verwenden können. Gehen Sie wie folgt vor, um den Valkey- oder Redis OSS-Clustermodus auf Clusterkompatibilität umzustellen:

 Note

Im kompatiblen Modus sind andere Änderungsvorgänge, wie Änderung der Skalierung und der Engine-Version, für den Cluster nicht zulässig. Darüber hinaus können Parameter (ausgenommen `cacheParameterGroupName`) nicht geändert werden, wenn der Clustermodus-Parameter in der Anfrage definiert wird. [ModifyReplicationGroup](#)

- a. Verwenden Sie den AWS-Managementkonsole, sehen Sie sich den Clustermodus an [Ändern einer Replikationsgruppe](#) und setzen Sie ihn auf Compatible
- b. Sehen Sie sich den `ClusterMode` Parameter mithilfe der API an [ModifyReplicationGroup](#) und aktualisieren Sie ihn auf `compatible`.
- c. Verwenden Sie den AWS CLI, sehen Sie sich den `cluster-mode` Parameter an [modify-replication-group](#) und aktualisieren Sie ihn auf `compatible`.

Nach dem Ändern des Valkey- oder Redis OSS-Clustermodus in den Clustermodus kompatibel, gibt die [DescribeReplicationGroups](#) API den Endpunkt ElastiCache für die Redis OSS-Clusterkonfiguration zurück. Der Cluster-Konfigurationsendpunkt ist ein einzelner Endpunkt, der von Anwendungen verwendet werden kann, um eine Verbindung mit dem Cluster herzustellen. Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).

3. Clusterkonfiguration in den aktivierten Clustermodus ändern – Sobald der Clustermodus auf den kompatiblen Clustermodus eingestellt ist, besteht der zweite Schritt darin, die Clusterkonfiguration so zu ändern, dass der Clustermodus aktiviert ist. In diesem Modus wird ein einzelner Shard ausgeführt und Kunden können jetzt ihre Cluster skalieren oder andere Clusterkonfigurationen ändern.

Gehen Sie wie folgt vor, um den Clustermodus in aktiviert zu ändern:

Bevor Sie beginnen, stellen Sie sicher, dass Ihre Valkey- oder Redis OSS-Clients auf das Cluster-Protokoll umgestellt wurden und dass der Konfigurationsendpunkt des Clusters nicht verwendet wird.

- a. Sehen Sie sich mit dem den AWS-Managementkonsole Clustermodus an [Ändern einer Replikationsgruppe](#) und setzen Sie ihn auf Aktiviert.
- b. Sehen Sie sich mithilfe der API den `ClusterMode` Parameter an [ModifyReplicationGroup](#) und aktualisieren Sie ihn auf `enabled`.
- c. Verwenden Sie den AWS CLI, sehen Sie sich den `cluster-mode` Parameter an [modify-replication-group](#) und aktualisieren Sie ihn auf `enabled`.

Nachdem der Clustermodus auf aktiviert geändert wurde, werden die Endpoints gemäß der Valkey- oder Redis OSS-Clusterspezifikation konfiguriert. Die [DescribeReplicationGroups](#) API gibt den Clustermodus-Parameter als `enabled` und die Cluster-Endpunkte zurück, die jetzt für Anwendungen zur Verbindung mit dem Cluster verfügbar sind.

Beachten Sie, dass sich die Cluster-Endpunkte ändern, sobald der Clustermodus auf aktiviert geändert wird. Achten Sie darauf, Ihre Anwendungen mit den neuen Endpunkten zu aktualisieren.

Sie können sich auch dafür entscheiden, vom kompatiblen Clustermodus (CMD) zum deaktivierten Clustermodus (CMD) zurückzukehren und die ursprünglichen Konfigurationen beizubehalten.

Ändern der Clusterkonfiguration vom kompatiblen in den deaktivierten Clustermodus

1. Sehen Sie sich den AWS-Managementkonsole Clustermodus mithilfe von an [Ändern einer Replikationsgruppe](#) und setzen Sie ihn auf Deaktiviert
2. Sehen Sie sich den `ClusterMode` Parameter mithilfe der API an [ModifyReplicationGroup](#) und aktualisieren Sie ihn auf `disabled`.
3. Verwenden Sie den AWS CLI, sehen Sie sich den `cluster-mode` Parameter an [modify-replication-group](#) und aktualisieren Sie ihn auf `disabled`.

Nachdem der Clustermodus auf deaktiviert geändert wurde, gibt die [DescribeReplicationGroups](#) API den Clustermodus-Parameter als `disabled` zurück.

AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher

Note

Global Datastore ist derzeit nur für knotenbasierte Cluster verfügbar.

Mithilfe der Global Datastore-Funktion können Sie mit einer vollständig verwalteten, schnellen, zuverlässigen und sicheren Valkey- oder Redis OSS-Clusterreplikation über Regionen hinweg arbeiten. AWS Mit dieser Funktion können Sie regionsübergreifende Read Replica-Cluster erstellen, um Lesevorgänge mit geringer Latenz und regionsübergreifende Disaster Recovery zu ermöglichen. AWS

In den folgenden Abschnitten finden Sie eine Beschreibung der Arbeit mit globalen Datenspeichern.

Themen

- [-Übersicht](#)
- [Voraussetzungen und Einschränkungen](#)
- [Verwendung globaler Datenspeicher \(Konsole\)](#)
- [Verwenden globaler Datenspeicher \(CLI\)](#)

-Übersicht

Jeder globale Datenspeicher ist eine Sammlung von einem oder mehreren Clustern, die zueinander repliziert werden.

Ein globaler Datenspeicher besteht aus folgenden Komponenten:

- Primärer (aktiver) Cluster – Ein primärer Cluster nimmt Schreibvorgänge entgegen, die auf alle Cluster innerhalb des globalen Datenspeichers repliziert werden. Ein primärer Cluster akzeptiert auch Leseanfragen.
- Sekundärer (passiver) Cluster – Ein sekundärer Cluster nimmt nur Leseanforderungen entgegen und repliziert Datenaktualisierungen von einem primären Cluster. Ein sekundärer Cluster muss sich in einer anderen AWS Region befinden als der primäre Cluster.

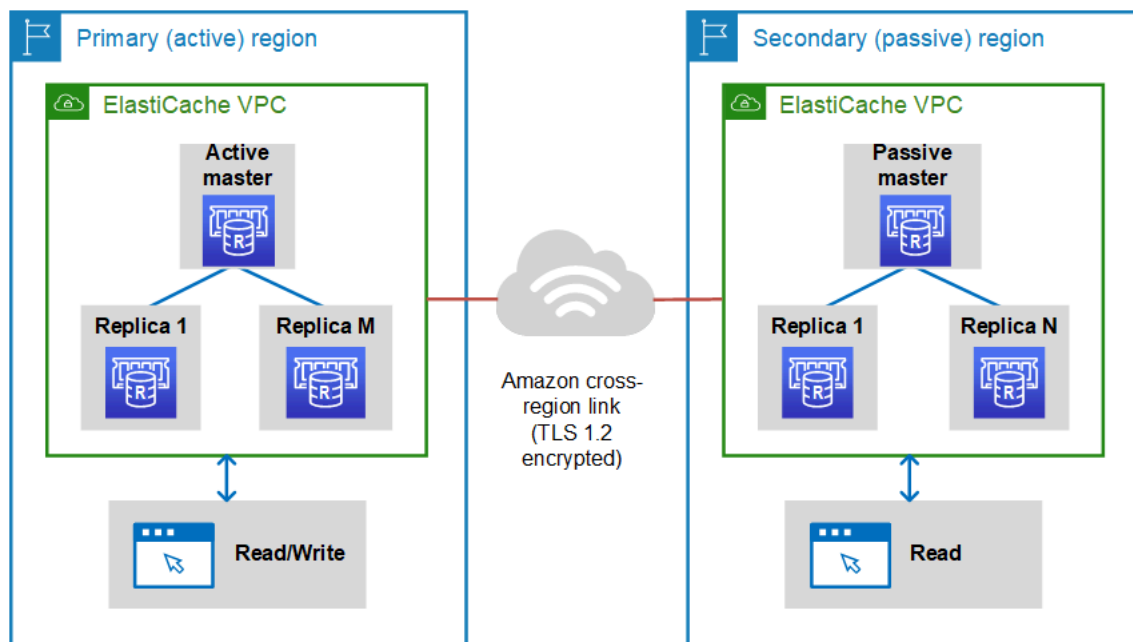
Wenn Sie einen globalen Datenspeicher ElastiCache für Valkey oder Redis OSS erstellen, repliziert dieser Ihre Daten automatisch vom primären Cluster auf den sekundären Cluster. Sie wählen die

AWS Region aus, in der die Valkey- oder Redis OSS-Daten repliziert werden sollen, und erstellen dann einen sekundären Cluster in dieser Region. AWS ElastiCache richtet dann die automatische, asynchrone Replikation von Daten zwischen den beiden Clustern ein und verwaltet sie.

Die Verwendung eines globalen Datenspeichers für Valkey oder Redis OSS bietet die folgenden Vorteile:

- **Geolokale Leistung** — Indem Sie Remote-Replikat-Cluster in zusätzlichen AWS Regionen einrichten und Ihre Daten zwischen diesen Regionen synchronisieren, können Sie die Latenz beim Datenzugriff in dieser Region reduzieren. AWS Ein globaler Datenspeicher kann dazu beitragen, die Reaktionsfähigkeit Ihrer Anwendung zu erhöhen, indem er geolokale Lesevorgänge mit niedriger Latenz in allen Regionen bereitstellt. AWS
- **Notfallwiederherstellung** – Wenn Ihr primärer Cluster in einem globalen Datenspeicher eine Verschlechterung aufweist, können Sie einen sekundären Cluster als Ihren neuen primären Cluster heraufstufen. Sie können dies tun, indem Sie eine Verbindung zu einer beliebigen Region herstellen, die einen sekundären Cluster enthält. AWS.

Das folgende Diagramm zeigt, wie globale Datenspeicher funktionieren können.



Voraussetzungen und Einschränkungen

Bevor Sie die Arbeit mit globalen Datenspeichern beginnen, beachten Sie Folgendes:

- Globale Datenspeicher werden in den folgenden AWS Regionen unterstützt:

- Afrika — Kapstadt
- Asien-Pazifik — Hongkong, Hyderabad, Jakarta, Malaysia, Melbourne, Mumbai, Osaka, Seoul, Singapur, Sydney, Thailand und Tokio
- Kanada — Kanada Zentral und Kanada West (Calgary)
- China - Peking und Ningxia
- Europa — Frankfurt, London, Irland, Mailand, Paris, Spanien, Stockholm und Zürich
- AWS GovCloud-US-West und US-Ost
- Israel - Tel Aviv
- Naher Osten — Bahrain und Vereinigte Arabische Emirate
- USA — Ost (Nord-Virginia und Ohio) und USA West (Nordkalifornien und Oregon)
- Südamerika — Mexiko (Zentral) und São Paulo
- Alle Cluster – primär und sekundär – in Ihrem globalen Datenspeicher sollten die gleiche Anzahl von primären Knoten, Knotentyp, Engine-Version und Anzahl von Shards (bei aktiviertem Cluster-Modus) aufweisen. Jeder Cluster in Ihrem globalen Datenspeicher kann über eine andere Anzahl von Read Replicas verfügen, um mit dem lokalen Lesedatenverkehr in diesem Cluster umgehen zu können.

Die Replikation muss aktiviert sein, wenn Sie einen vorhandenen Cluster mit einem einzelnen Knoten verwenden möchten.

- Globale Datenspeicher werden auf Instances mit einer Größe von mindestens einer Größe unterstützt.
- Sie können die Replikation für einen primären Cluster von einer AWS Region zu einem sekundären Cluster in bis zu zwei anderen AWS Regionen einrichten.

 Note

Eine Ausnahme sind die Regionen China (Peking) und China (Ningxia), wo eine Replikation nur zwischen den beiden Regionen erfolgen kann.

- Mit globalen Datenspeichern können Sie nur in VPC-Clustern arbeiten. Weitere Informationen finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#). Globale Datenspeicher werden nicht unterstützt, wenn Sie -Classic verwenden EC2. Weitere Informationen finden Sie unter [EC2-Classic](#) im EC2 Amazon-Benutzerhandbuch.

 Note

Derzeit können Sie keine globalen Datenspeicher in [Verwenden von lokalen Zonen mit ElastiCache](#) verwenden.

- ElastiCache unterstützt kein automatisches Failover von einer AWS Region in eine andere. Bei Bedarf können Sie einen sekundären Cluster manuell heraufstufen. Ein Beispiel finden Sie unter [Heraufstufen des sekundären Clusters zum primären Cluster](#).
- Um eine Bootstrap-Aktion von vorhandenen Daten durchzuführen, verwenden Sie einen vorhandenen Cluster als primären Cluster, um einen globalen Datenspeicher zu erstellen. Das Hinzufügen eines vorhandenen Clusters als sekundärer Cluster wird nicht unterstützt. Beim Hinzufügen des Clusters als sekundärer Cluster werden Daten gelöscht, was zu Datenverlusten führen kann.
- Parameteraktualisierungen werden auf alle Cluster angewendet, wenn Sie eine lokale Parametergruppe eines Clusters ändern, der zu einem globalen Datenspeicher gehört.
- Sie können regionale Cluster sowohl vertikal (Skalierung nach oben und unten) als auch horizontal (Ein- und Ausskalierung) skalieren. Sie können die Cluster skalieren, indem Sie den globalen Datenspeicher ändern. Alle regionalen Cluster in dem globalen Datenspeicher werden dann ohne Unterbrechung skaliert. Weitere Informationen finden Sie unter [Skalierung ElastiCache](#).
- [Globale Datenspeicher unterstützen Verschlüsselung im Ruhezustand, Verschlüsselung bei der Übertragung und AUTH.](#)
- Globale Datenspeicher unterstützen das Internetprotokoll Version 6 () nicht. IPv6
- Globale Datenspeicher unterstützen Schlüssel. AWS KMS Weitere Informationen finden Sie unter [AWS Schlüsselverwaltungsdienstkonzpte](#) im AWS Key Management Service Entwicklerhandbuch.

 Note

Globale Datenspeicher unterstützen [Pub/Sub-Messaging](#) mit den folgenden Vorgaben:

- Für den Cluster-Modus deaktiviert, pub/sub wird vollständig unterstützt. Ereignisse, die auf dem primären Cluster der primären AWS Region veröffentlicht wurden, werden an sekundäre Regionen weitergegeben. AWS
- Bei aktiviertem Cluster-Modus gilt Folgendes:

- Bei veröffentlichten Ereignissen, die sich nicht in einem Schlüsselraum befinden, erhalten nur Abonnenten in derselben AWS Region die Ereignisse.
- Bei veröffentlichten Keyspace-Veranstaltungen erhalten Abonnenten in allen AWS Regionen die Ereignisse.

Verwendung globaler Datenspeicher (Konsole)

Führen Sie den folgenden aus zwei Schritten bestehenden Vorgang durch, um einen globalen Datenspeicher mit der Konsole zu erstellen:

1. Erstellen Sie einen primären Cluster, entweder mithilfe eines vorhandenen Clusters oder durch Erstellen eines neuen Clusters. Die Engine muss Valkey 7.2 oder höher oder Redis OSS 5.0.6 oder höher sein.
2. Fügen Sie bis zu zwei sekundäre Cluster in verschiedenen AWS Regionen hinzu, wiederum mit Valkey 7.2 oder höher oder Redis OSS 5.0.6 oder höher.

In den folgenden Verfahren erfahren Sie, wie Sie einen globalen Datenspeicher für Valkey oder Redis OSS erstellen und andere Vorgänge mithilfe der Konsole ausführen. ElastiCache

Themen

- [Erstellen eines globalen Datenspeichers mit einem vorhandenen Cluster](#)
- [Erstellen eines neuen globalen Datenspeichers mit einem neuen primären Cluster](#)
- [Anzeigen von Details zu einem globalen Datenspeicher](#)
- [Hinzufügen einer Region zu einem globalen Datenspeicher](#)
- [Ändern eines globalen Datenspeichers](#)
- [Heraufstufen des sekundären Clusters zum primären Cluster](#)
- [Entfernen einer Region aus einem globalen Datenspeicher](#)
- [Löschen eines globalen Datenspeichers](#)

Erstellen eines globalen Datenspeichers mit einem vorhandenen Cluster

In diesem Szenario verwenden Sie einen vorhandenen Cluster, der als primärer Cluster des neuen globalen Datenspeichers dient. Anschließend erstellen Sie einen sekundären, schreibgeschützten

Cluster in einer separaten AWS -Region. Dieser sekundäre Cluster erhält automatische und asynchrone Updates vom primären Cluster.

 Important

Der vorhandene Cluster muss eine Engine verwenden, die Valkey 7.2 oder höher oder Redis OSS 5.0.6 oder höher ist.

So erstellen Sie einen globalen Datenspeicher unter Verwendung eines vorhandenen Clusters:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Global Datastores und dann Create Global Datastore aus.
3. Gehen Sie auf der Seite mit den Einstellungen für den primären Cluster wie folgt vor:
 - Geben Sie im Feld Informationen zum globalen Datenspeicher einen Namen für den neuen globalen Datenspeicher ein.
 - (Optional) Geben Sie einen Wert für Description (Beschreibung) ein.
4. Wählen Sie unter Regionaler Cluster die Option Bestehenden regionalen Cluster verwenden aus.
5. Wählen Sie unter Existierender Cluster den vorhandenen Cluster aus, den Sie verwenden möchten.
6. Behalten Sie die folgenden Optionen unverändert bei. Sie sind so vorausgefüllt, dass sie der primären Clusterkonfiguration entsprechen. Sie können sie nicht ändern.
 - Engine-Version
 - Knotentyp
 - Parametergruppe

 Note

ElastiCache generiert automatisch eine neue Parametergruppe aus Werten der angegebenen Parametergruppe und wendet die neue Parametergruppe auf den Cluster an. Verwenden Sie diese neue Parametergruppe, um Parameter in einem globalen Datenspeicher zu ändern. Jede automatisch generierte Parametergruppe ist einem Cluster und daher nur einem globalen Datenspeicher zugeordnet.

- Anzahl der Shards
- Verschlüsselung im Ruhezustand – Aktiviert die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

 Note

Sie können einen anderen Verschlüsselungsschlüssel angeben, indem Sie Customer Managed AWS KMS key und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwenden von vom Kunden verwalteten AWS KMS-Schlüsseln](#).

- Verschlüsselung während der Übertragung – Aktiviert die Verschlüsselung von Daten bei der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey 7.2 und höher und ab Redis OSS 6.0 die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:
 - Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies weist auf keine Einschränkungen hin.
 - Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einem definierten Satz von Benutzern und Berechtigungen für verfügbare Vorgänge aus. Weitere Informationen finden Sie unter [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
 - AUTH Default User — Ein Authentifizierungsmechanismus für einen Valkey- oder Redis OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).
7. (Optional) Aktualisieren Sie bei Bedarf die verbleibenden Einstellungen für den sekundären Cluster. Diese werden mit den gleichen Werten wie beim primären Cluster vorausgefüllt, Sie können sie jedoch aktualisieren, um bestimmte Anforderungen für diesen Cluster zu erfüllen.
- Port
 - Anzahl der Replikate
 - Subnetzgruppe
 - Bevorzugte Availability Zone(s)
 - Sicherheitsgruppen
 - Vom Kunden verwaltet (AWS KMS-Schlüssel)
 - AUTH-Token

- Aktivieren automatischer Sicherungen.
 - Aufbewahrungszeitraum für Backups
 - Backup-Fenster
 - Wartungsfenster
 - Thema für SNS-Benachrichtigung
8. Wählen Sie Erstellen aus. Dadurch wird der Status des globalen Datenspeichers auf Creating (Erstellen) festgelegt. Der Status wechselt zu Modifying (Ändern), nachdem der primäre Cluster dem globalen Datenspeicher zugeordnet wurde und der sekundäre Cluster den Status Associating (Zuordnen) hat.

Nachdem der primäre Cluster und die sekundären Cluster mit dem globalen Datenspeicher verknüpft wurden, ändert sich der Status zu Available (Verfügbar). An dieser Stelle verfügen Sie über einen primären Cluster, der Lese- und Schreibvorgänge akzeptiert, sowie sekundäre Cluster, die vom primären Cluster replizierte Lesevorgänge akzeptieren.

Die Seite wird aktualisiert und gibt nun an, ob ein Cluster Teil eines globalen Datenspeichers ist. Dazu gehören:

- Globaler Datenspeicher – Der Name des globalen Datenspeichers, zu dem der Cluster gehört.
- Globale Datenspeicherrolle – Die Rolle des Clusters, entweder primär oder sekundär.

Sie können bis zu einem zusätzlichen sekundären Cluster in einer anderen AWS Region hinzufügen. Weitere Informationen finden Sie unter [Hinzufügen einer Region zu einem globalen Datenspeicher](#).

Erstellen eines neuen globalen Datenspeichers mit einem neuen primären Cluster

Wenn Sie einen globalen Datenspeicher mit einem neuen Cluster erstellen möchten, gehen Sie folgendermaßen vor.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Global Datastores und dann Create Global Datastore aus.
3. Führen Sie unter Primary cluster settings (Primäre Cluster-Einstellungen) die folgenden Schritte aus:
 - a. Wählen Sie für Cluster mode (Cluster-Modus) Enabled (Aktiviert) oder Disabled (Deaktiviert) aus.

- b. Geben Sie für Informationen zum globalen Datenspeicher einen Wert für Name ein. ElastiCache verwendet das Suffix, um einen eindeutigen Namen für den globalen Datenspeicher zu generieren. Sie können nach dem globalen Datenspeicher suchen, indem Sie das hier angegebene Suffix verwenden.
 - c. (Optional) Geben Sie einen Wert für die Global Datastore description (Beschreibung des globalen Datenspeichers) ein.
4. Unter Regional cluster (Regionaler Cluster):
 - a. Wählen Sie für Region eine verfügbare Region aus. AWS
 - b. Wählen Sie Create new regional cluster (Neuen regionalen Cluster erstellen) oder Use existing regional cluster (Bestehenden regionalen Cluster verwenden) aus.
 - c. Wenn Sie Create new regional cluster (Neuen regionalen Cluster erstellen), auswählen, geben Sie unter Cluster infos (Cluster-Infos) einen Namen und eine optionale Beschreibung des Clusters ein.
 - d. Wir empfehlen, unter Location (Speicherort) die Voreinstellungen für Multi-AZ und Auto-failover (Automatisches Failover) zu akzeptieren.
5. Unter Cluster settings (Cluster-Einstellungen):
 - a. Wählen Sie für Engine version (Engine-Version) eine verfügbare Version (5.0.6 oder höher).
 - b. Verwenden Sie für Port den Standardport 6379. Wenn es einen Grund gibt, einen anderen Port zu verwenden, geben Sie die betreffende Portnummer ein.
 - c. Wählen Sie für die Parameter group (Parametergruppe) eine Parametergruppe aus oder erstellen Sie eine neue Parametergruppe. Parametergruppen steuern die Laufzeitparameter Ihres Clusters. Weitere Informationen zu Parametergruppen finden Sie unter [Valkey- und Redis OSS-Parameter](#) und [Eine ElastiCache Parametergruppe erstellen](#).

 Note

Wenn Sie eine Parametergruppe zum Festlegen der Engine-Konfigurationswerte auswählen, wird diese Parametergruppe auf alle Cluster im globalen Datenspeicher angewendet. Auf der Seite Parameter Groups (Parametergruppen) gibt das Ja/nein-Attribut Global an, ob eine Parametergruppe Teil eines globalen Datenspeichers ist.

- d. Klicken Sie für Node type (Knotentyp) auf den Abwärtspfeil (▼). Wählen Sie im Dialogfeld Change node type (Knotentyp ändern) einen Wert für Instance

family (Instance-Familie) für den gewünschten Knotentyp aus. Wählen Sie dann den Knotentyp aus, den Sie für diesen Cluster verwenden möchten, und wählen Sie dann Save (Speichern).

Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Wenn Sie einen r6gd-Knotentyp wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

- e. Wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) erstellen:

Wählen Sie unter Number of replicas (Anzahl der Replikate) die für diesen Cluster gewünschte Anzahl von Replikaten aus.

- f. Wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) erstellen:

- i. Wählen Sie unter Anzahl der Shards die Anzahl der Shards (Partitionen/ Knotengruppen) aus, die Sie für diesen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) benötigen.

Bei einigen Versionen von Valkey oder Redis OSS (Clustermodus aktiviert) können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern:

- Redis OSS 3.2.10 und höher — Wenn auf Ihrem Cluster Redis OSS 3.2.10 oder spätere Versionen ausgeführt werden, können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern. Weitere Informationen finden Sie unter [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#).
 - Andere Redis OSS-Versionen — Wenn auf Ihrem Cluster eine Version von Redis OSS vor Version 3.2.10 ausgeführt wird, gibt es einen anderen Ansatz. Um die Anzahl der Shards in Ihrem Cluster in diesem Fall zu ändern, erstellen Sie einen neuen Cluster mit der neuen Anzahl von Shards. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).
- ii. Wählen Sie für Replicas per shard (Replikate pro Shard) die Anzahl der Read Replica-Knoten aus, die sich in jedem Shard befinden sollen.

Die folgenden Einschränkungen gelten für Valkey oder Redis OSS (Clustermodus aktiviert).

- Wenn Sie Multi-AZ aktiviert haben, stellen Sie sicher, dass mindestens ein Replikat pro Shard vorhanden ist.

- Die Anzahl der Replikate ist für jeden Shard gleich, wenn der Cluster mithilfe der Konsole erstellt wird.
- Die Anzahl der Lesereplikate pro Shard ist ein fester Wert, der nicht geändert werden kann. Wenn Sie feststellen, dass Sie mehr oder weniger Replikate pro Shard (API/CLI: Knotengruppe) benötigen, müssen Sie einen neuen Cluster mit der neuen Anzahl von Replikaten erstellen. Weitere Informationen finden Sie unter [Tutorial: Seeding eines neuen knotenbasierten Clusters mit einem extern erstellten Backup](#).

6. Wählen Sie unter Einstellungen für Subnetzgruppen das Subnetz aus, das Sie auf diesen Cluster anwenden möchten. ElastiCache stellt eine IPv4 Standard-Subnetzgruppe bereit, oder Sie können wählen, ob Sie eine neue erstellen möchten. Denn IPv6 Sie müssen eine Subnetzgruppe mit einem IPv6 CIDR-Block erstellen. Wenn Sie sich für Dual-Stack entscheiden, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder oder IPv6 . IPv4

Weitere Informationen finden Sie unter [Erstellen eines Subnetzes in Ihrer VPC](#).

7. Für Availability zone placements (Availability-Zone-Platzierungen) haben Sie zwei Optionen:
- Keine Präferenz — ElastiCache wählt die Availability Zone.
 - Availability Zones angeben – Sie geben die Availability Zone für jeden Cluster an.

Wenn Sie die Availability Zones angeben, wählen Sie für jeden Cluster in jedem Shard die Availability Zone aus der Liste aus.

Weitere Informationen finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

Slots and keyspaces Custom distribution ⓘ

Availability zone(s) Specify availability zones ⓘ

	Slots/Keyspaces	Primary	Replica 1
NodeGroup 1	0-1234	us-east-1a	us-east-1a
NodeGroup 2		us-east-1b	us-east-1a
NodeGroup 3		us-east-1a	us-east-1a

Angeben von Keyspaces und Availability Zones

8. Wählen Sie Next (Weiter).
9. Unter Advanced Valkey und Redis OSS-Einstellungen
 - Für Security (Sicherheit):

i. Zur Verschlüsselung Ihrer Daten haben Sie die folgenden Optionen:

- Verschlüsselung im Ruhezustand – Ermöglicht die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

Note

Sie haben die Möglichkeit, einen anderen Verschlüsselungsschlüssel anzugeben, indem Sie Customer Managed AWS KMS Key und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwendung von kundenverwalteten Schlüsseln aus AWS -KMS](#).

- Verschlüsselung während der Übertragung – Ermöglicht die Verschlüsselung von Daten während der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey 7.2 und höher und

Redis OSS 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:

- Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
- Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern aus, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
- AUTH Default User — Ein Authentifizierungsmechanismus für einen Valkey- oder Redis OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).
- AUTH — Ein Authentifizierungsmechanismus für einen Valkey- oder Redis OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).

 Note

Für Redis OSS-Versionen zwischen 3.2.6 und höher, mit Ausnahme von Version 3.2.10, ist AUTH die einzige Option.

- ii. Wählen Sie für Security groups (Sicherheitsgruppen) die gewünschten Sicherheitsgruppen für diesen Cluster aus. Eine security group (Sicherheitsgruppe) fungiert als Firewall, um den Netzwerkzugriff auf Ihren Cluster zu steuern. Sie können die Standardsicherheitsgruppe für Ihre VPC verwenden oder eine neue erstellen.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Sicherheitsgruppen für Ihre VPC](#) im Benutzerhandbuch zu Amazon VPC.

10. Wenn Sie regelmäßig geplante automatische Sicherungen möchten, aktivieren Sie Enable automatic backups und geben Sie ein, wie viele Tage lang jede automatische Sicherung beibehalten werden soll, bevor sie automatisch gelöscht wird. Wenn Sie keine regelmäßig geplanten automatischen Sicherungen möchten, deaktivieren Sie das Kontrollkästchen Enable automatic backups. In beiden Fällen haben Sie jederzeit die Option, manuelle Sicherungen zu erstellen.

Weitere Informationen zur Sicherung und Wiederherstellung finden Sie unter [Snapshot und Wiederherstellung](#)

11. (Optional) Geben Sie ein Wartungsfenster an. Das Maintenance window (Wartungsfenster) ist der (in der Regel eine Stunde lange) allwöchentliche Zeitraum, für den ElastiCache die

Systemwartung Ihres Clusters plant. Sie können ElastiCache den Tag und die Uhrzeit für Ihr Wartungsfenster wählen (keine Präferenz), oder Sie können Tag, Uhrzeit und Dauer selbst wählen (Wartungsfenster angeben). Treffen Sie bei Wahl von Specify maintenance window eine Auswahl in den Listen Start day, Start time und Duration (in Stunden) für Ihr Wartungsfenster. Alle Uhrzeiten sind in UCT angegeben.

Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

12. (Optional) Für Protokolle:

- Unter Protokollformat wählen Sie entweder Text oder JSON aus.
- Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
- Wählen Sie unter Protokollziel entweder Neu erstellen aus und geben Sie entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und wählen Sie dann entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Streamnamen aus.

13. Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen unter Tags zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).

14. Überprüfen Sie alle Ihre Einträge und ausgewählten Optionen und machen Sie dann evtl. erforderliche Korrekturen. Wählen Sie danach Next (Weiter) aus.

15. Nachdem Sie den Cluster in den vorherigen Schritten konfiguriert haben, konfigurieren Sie nun die Details des sekundären Clusters.

16. Wählen Sie unter Regionaler Cluster die AWS Region aus, in der sich der Cluster befindet.

17. Geben Sie unter Cluster info (Cluster-Info) einen Namen und eine optionale Beschreibung des Clusters ein.

18. Die folgenden Optionen sind für die Konfiguration des primären Clusters vorausgefüllt und können nicht geändert werden:

- Ort
- Engine-Version
- Instance-Typ
- Knotentyp
- Anzahl der Shards
- Parametergruppe

 Note

ElastiCache generiert automatisch eine neue Parametergruppe aus Werten der angegebenen Parametergruppe und wendet die neue Parametergruppe auf den Cluster an. Verwenden Sie diese neue Parametergruppe, um Parameter in einem globalen Datenspeicher zu ändern. Jede automatisch generierte Parametergruppe ist einem Cluster und daher nur einem globalen Datenspeicher zugeordnet.

- Verschlüsselung im Ruhezustand – Aktiviert die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

 Note

Sie können einen anderen Verschlüsselungsschlüssel angeben, indem Sie Customer Managed AWS KMS key und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwenden von vom Kunden verwalteten AWS KMS-Schlüsseln](#).

- Verschlüsselung während der Übertragung – Aktiviert die Verschlüsselung von Daten bei der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey 7.2 und höher und Redis OSS 6.4 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:
 - Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
 - Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
 - AUTH Default User — Ein Authentifizierungsmechanismus für einen Valkey- oder Redis OSS-Server. [Weitere Informationen finden Sie unter AUTH](#).

 Note

Für Redis OSS-Versionen zwischen 4.0.2, als Verschlüsselung bei der Übertragung erstmals unterstützt wurde, und 6.0.4 ist AUTH die einzige Option.

Die verbleibenden Einstellungen für den sekundären Cluster werden mit den gleichen Werten wie der primäre Cluster ausgefüllt, aber Folgendes kann aktualisiert werden, um bestimmte Anforderungen für diesen Cluster zu erfüllen.

- Port
 - Anzahl der Replikate
 - Subnetzgruppe
 - Bevorzugte Availability Zone(s)
 - Sicherheitsgruppen
 - Vom Kunden verwaltet (KMS-Schlüssel)AWS
 - AUTH-Token
 - Aktivieren automatischer Sicherungen.
 - Aufbewahrungszeitraum für Backups
 - Backup-Fenster
 - Wartungsfenster
 - Thema für SNS-Benachrichtigung
19. Wählen Sie Erstellen aus. Dadurch wird der Status des globalen Datenspeichers auf Creating (Erstellen) gesetzt. Nachdem der primäre Cluster und die sekundären Cluster mit dem globalen Datenspeicher verknüpft wurden, ändert sich der Status zu Available (Verfügbar). Sie verfügen über einen primären Cluster, der Lese- und Schreibvorgänge akzeptiert, und einen sekundären Cluster, der Lesevorgänge akzeptiert, die vom primären Cluster repliziert werden.

Die Seite wird außerdem aktualisiert, um anzugeben, ob ein Cluster Teil eines globalen Datenspeichers ist, einschließlich der folgenden Informationen:

- Globaler Datenspeicher – Der Name des globalen Datenspeichers, zu dem der Cluster gehört.
- Globale Datenspeicherrolle – Die Rolle des Clusters, entweder primär oder sekundär.

Sie können bis zu einem zusätzlichen sekundären Cluster in einer anderen AWS Region hinzufügen. Weitere Informationen finden Sie unter [Hinzufügen einer Region zu einem globalen Datenspeicher](#).

Anzeigen von Details zu einem globalen Datenspeicher

Sie können die Details vorhandener globaler Datenspeicher anzeigen und sie auch auf der Seite Globale Datenspeicher ändern.

So zeigen Sie Details zu einem globalen Datenspeicher an:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Globale Datenspeicher und dann einen verfügbaren globalen Datenspeicher aus.

Anschließend können Sie die folgenden Eigenschaften des globalen Datenspeichers untersuchen:

- Global Datastore Name (Name des globalen Datenspeichers): Der Name des globalen Datenspeichers
- Description (Beschreibung): Eine Beschreibung des globalen Datenspeichers
- Status: Folgende Optionen stehen zur Verfügung:
 - Erstellen
 - Ändern
 - Verfügbar
 - Löschen
 - Nur Primär - Dieser Status gibt an, dass der globale Datenspeicher nur einen primären Cluster enthält. Entweder wurden alle sekundären Cluster gelöscht oder nicht erfolgreich erstellt.
- Cluster Mode (Cluster-Modus): Aktiviert oder deaktiviert.
- Engine-Version: Die Valkey- oder Redis OSS-Engine-Version, auf der der globale Datenspeicher ausgeführt wird
- Instance Node Type (Instance-Knotentyp): Der für den globalen Datenspeicher verwendete Knotentyp
- Encryption at-rest (Verschlüsselung im Ruhezustand): Aktiviert oder deaktiviert
- Encryption in-transit (Verschlüsselung während der Übertragung): Aktiviert oder deaktiviert
- AUTH: Entweder aktiviert oder deaktiviert

Sie können die folgenden Änderungen an dem globalen Datenspeicher vornehmen:

- [Hinzufügen einer Region zu einem globalen Datenspeicher](#)
- [Entfernen einer Region aus einem globalen Datenspeicher](#)
- [Heraufstufen des sekundären Clusters zum primären Cluster](#)
- [Ändern eines globalen Datenspeichers](#)

Auf der Seite „Global Datastore (Globaler Datenspeicher)“ werden auch die einzelnen Cluster aufgelistet, aus denen der globale Datenspeicher besteht, sowie die folgenden Eigenschaften für jeden davon:

- Region — Die AWS Region, in der der Cluster gespeichert ist
- Role (Rolle) - Entweder primär oder sekundär
- Cluster name (Cluster-Name) - Der Name des Clusters
- Status - Zu den Optionen gehören:
 - Associating (Zuordnen) - Der Cluster wird gerade dem globalen Datenspeicher zugeordnet
 - Associated (Zugeordnet) - Der Cluster ist dem globalen Datenspeicher zugeordnet
 - Disassociating (Trennen) - Der Vorgang des Entfernens eines sekundären Clusters aus dem globalen Datenspeicher unter Verwendung des Namens des globalen Datenspeichers. Danach erhält der sekundäre Cluster keine Updates mehr vom primären Cluster, sondern er verbleibt als eigenständiger Cluster in dieser AWS Region.
 - Disassociated (Getrennt) - Der sekundäre Cluster wurde aus dem globalen Datenspeicher entfernt und ist jetzt ein eigenständiger Cluster in seiner AWS -Region.
- Global Datastore Replica Lag — Zeigt einen Wert pro sekundärer AWS Region im globalen Datenspeicher an. Dies ist die Verzögerung zwischen dem Primärknoten der sekundären Region und dem Primärknoten der primären Region. Bei aktiviertem Valkey oder Redis OSS im Clustermodus gibt die Verzögerung die maximale Verzögerung in Sekunden zwischen den Shards an.

Hinzufügen einer Region zu einem globalen Datenspeicher

Sie können einem vorhandenen globalen Datenspeicher bis zu eine zusätzliche AWS Region hinzufügen. In diesem Szenario erstellen Sie einen schreibgeschützten Cluster in einer separaten AWS Region, der automatische und asynchrone Updates vom primären Cluster empfängt.

Um eine AWS Region zu einem globalen Datenspeicher hinzuzufügen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Globale Datenspeicher und dann einen vorhandenen globalen Datenspeicher aus.
3. Wählen Sie Regionalen Cluster hinzufügen und wählen Sie die AWS Region aus, in der sich der sekundäre Cluster befinden soll.
4. Geben Sie unter Clusterinformationen einen Wert für Name und optional für Beschreibung für den Cluster ein.
5. Behalten Sie die folgenden Optionen unverändert bei. Sie sind bereits so ausgefüllt, dass sie der primären Clusterkonfiguration entsprechen, und Sie können sie nicht ändern.

- Engine-Version
- Instance-Typ
- Knotentyp
- Anzahl der Shards
- Parametergruppe

Note

ElastiCache generiert automatisch eine neue Parametergruppe aus Werten der angegebenen Parametergruppe und wendet die neue Parametergruppe auf den Cluster an. Verwenden Sie diese neue Parametergruppe, um Parameter in einem globalen Datenspeicher zu ändern. Jede automatisch generierte Parametergruppe ist einem Cluster und daher nur einem globalen Datenspeicher zugeordnet.

- Verschlüsselung im Ruhezustand

Note

Sie können einen anderen Verschlüsselungsschlüssel angeben, indem Sie Customer Managed AWS KMS key und dann den Schlüssel auswählen.

- Verschlüsselung während der Übertragung
- AUTHENTIFIZIEREN

6. (Optional) Aktualisieren Sie die verbleibenden Einstellungen für den sekundären Cluster. Diese werden mit den gleichen Werten wie beim primären Cluster vorausgefüllt, Sie können sie jedoch aktualisieren, um bestimmte Anforderungen für diesen Cluster zu erfüllen.
 - Port
 - Anzahl der Replikate
 - Subnetzgruppe
 - Bevorzugte Availability Zone(s)
 - Sicherheitsgruppen
 - Vom Kunden verwalteter AWS KMS-Schlüssel)
 - AUTH-Token
 - Aktivieren automatischer Sicherungen.
 - Aufbewahrungszeitraum für Backups
 - Backup-Fenster
 - Wartungsfenster
 - Thema für SNS-Benachrichtigung
7. Wählen Sie Hinzufügen aus.

Ändern eines globalen Datenspeichers

Sie können die Eigenschaften regionaler Cluster ändern. In einem globalen Datenspeicher kann nur jeweils eine Änderungsoperation ausgeführt werden, mit Ausnahme der Heraufstufung eines sekundären Clusters zu einem primären Cluster. Weitere Informationen finden Sie unter [Heraufstufen des sekundären Clusters zum primären Cluster](#).

So ändern Sie einen globalen Datenspeicher:

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Global Datastores und dann für Global Datastore Name einen globalen Datenspeicher aus.
3. Wählen Sie Modify (Ändern), und wählen Sie unter den folgenden Optionen:
 - Modify description (Beschreibung ändern) – Aktualisieren der Beschreibung des globalen **Datenspeichers**

- Engine-Version ändern — Nur Valkey 7.2 und höher oder Redis OSS 5.0.6 und höher sind verfügbar.
- Modify node type (Knotentyp ändern) – Vertikale (Aufwärts- und Abwärts-) und horizontale (Ein- und Aus-) Skalierung regionaler Cluster. Zu den Optionen gehören die R5- und M5-Knotenfamilien. Weitere Informationen zu Knotentypen finden Sie unter [Unterstützte Knotentypen](#).
- Modify Automatic Failover (Ändern des automatischen Failovers) – Aktivieren oder deaktivieren Sie das automatische Failover. Wenn Sie Failover aktivieren und Primärknoten in regionalen Clustern unerwartet heruntergefahren werden, erfolgt ein Failover ElastiCache zu einem der regionalen Replikate. Weitere Informationen finden Sie im Abschnitt [Automatisches Failover](#).

Für Valkey- oder Redis-OSS-Cluster mit aktiviertem Clustermodus:

- Add shards (Shards hinzufügen) – Geben Sie die Anzahl der hinzuzufügenden Shards ein, und geben Sie eine oder mehrere Availability Zones an.
- Shards löschen — Wählen Sie Shards aus, die in jeder Region gelöscht werden sollen. AWS
- Rebalance shards (Shards neu ausbalancieren) – Balancieren Sie die Slot-Verteilung neu aus, um eine gleichmäßige Verteilung über die in dem Cluster vorhandenen Shards zu gewährleisten.

Um die Parameter eines globalen Datenspeichers zu ändern, ändern Sie die Parametergruppe eines beliebigen Mitgliedsclusters für den globalen Datenspeicher. ElastiCache wendet diese Änderung automatisch auf alle Cluster innerhalb dieses globalen Datenspeichers an. Verwenden Sie die Valkey- oder Redis-OSS-Konsole oder den API-Vorgang, um die Parametergruppe dieses Clusters zu ändern. [ModifyCacheCluster](#) Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#). Wenn Sie die Parametergruppe eines Clusters ändern, der in einem globalen Datenspeicher enthalten ist, wird sie auf alle Cluster innerhalb dieses globalen Datenspeichers angewendet.

Verwenden Sie die [ResetCacheParameterGroup](#) API-Operation, um eine gesamte Parametergruppe oder bestimmte Parameter zurückzusetzen.

Heraufstufen des sekundären Clusters zum primären Cluster

Wenn der primäre Cluster oder die AWS Region nicht mehr verfügbar ist oder Leistungsprobleme auftreten, können Sie einen sekundären Cluster zum primären Cluster heraufstufen. Die

Heraufstufung ist jederzeit erlaubt, auch wenn gleichzeitig andere Änderungen vorgenommen werden. Sie können auch mehrere Heraufstufungen gleichzeitig durchführen, und der globale Datenspeicher wird schließlich zu einem primären Cluster aufgelöst. Wenn Sie mehrere sekundäre Cluster gleichzeitig heraufstufen, kann ElastiCache nicht garantiert werden, welcher letztendlich zum primären Cluster wird.

So stufen Sie einen sekundären Cluster zu einem primären Cluster herauf:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Global Datastores aus.
3. Wählen Sie den Namen des globalen Datenspeichers aus, um die Details anzuzeigen.
4. Wählen Sie den Secondary (sekundären) Cluster.
5. Wählen Sie Auf Primär heraufstufen.

Sie werden dann mit der folgenden Warnung aufgefordert, Ihre Entscheidung zu bestätigen:
Promoting a region to primary will make the cluster in this region as read/writable. Are you sure you want to promote the *secondary* cluster to primary?

The current primary cluster in *primary region* will become secondary and will stop accepting writes after this operation completes. Please ensure you update your application stack to direct traffic to the new primary region.

6. Wählen Sie Confirm (Bestätigen), wenn Sie die Heraufstufung fortsetzen möchten, oder Cancel (Abbrechen), wenn Sie dies nicht möchten.

Wenn Sie sich für die Bestätigung entscheiden, wechselt Ihr globaler Datenspeicher in den Status Modifying (Ändern) und ist erst verfügbar, wenn die Heraufstufung abgeschlossen ist.

Entfernen einer Region aus einem globalen Datenspeicher

Sie können eine AWS Region aus einem globalen Datenspeicher entfernen, indem Sie das folgende Verfahren verwenden.

Um eine AWS Region aus einem globalen Datenspeicher zu entfernen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Global Datastores aus.
3. Wählen Sie einen globalen Datenspeicher aus.
4. Wählen Sie die Region aus, die Sie entfernen möchten.
5. Wählen Sie Remove region (Region entfernen).

Note

Diese Option ist nur für sekundäre Cluster verfügbar.

Sie werden dann mit der folgenden Warnung aufgefordert, Ihre Entscheidung zu bestätigen:

Removing the region will remove your only available cross region replica for the primary cluster. Your primary cluster will no longer be set up for disaster recovery and improved read latency in remote region. Are you sure you want to remove the selected region from the global datastore?

6. Wählen Sie Confirm (Bestätigen), wenn Sie die Heraufstufung fortsetzen möchten, oder Cancel (Abbrechen), wenn Sie dies nicht möchten.

Wenn Sie „Bestätigen“ wählen, wird die AWS Region entfernt und der sekundäre Cluster erhält keine Replikationsupdates mehr.

Löschen eines globalen Datenspeichers

Um einen globalen Datenspeicher zu löschen, entfernen Sie zuerst alle sekundären Cluster. Weitere Informationen finden Sie unter [Entfernen einer Region aus einem globalen Datenspeicher](#). Dadurch bleibt der globale Datenspeicher im Status primary-only.

So löschen Sie einen globalen Datenspeicher:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Global Datastores aus.

3. Wählen Sie unter Global Datastore Name (Name des globalen Datenspeichers) den globalen Datenspeicher, den Sie löschen möchten, und dann Delete (Löschen).

Sie werden dann mit der folgenden Warnung aufgefordert, Ihre Entscheidung zu bestätigen: Are you sure you want to delete this Global Datastore?

4. Wählen Sie Löschen aus.

Der globale Datenspeicher wechselt zum Status Deleting (Löschen).

Verwenden globaler Datenspeicher (CLI)

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um mehrere AWS Dienste von der Befehlszeile aus zu steuern und sie mithilfe von Skripten zu automatisieren. Sie können die AWS CLI für (einmalige) Ad-hoc-Operationen verwenden.

Herunterladen und Konfigurieren der AWS CLI

Das AWS CLI läuft unter Windows, MacOS oder Linux. Gehen Sie folgendermaßen vor, um sie herunterzuladen und zu konfigurieren.

So laden Sie den CLI herunter, installieren und konfigurieren ihn

1. Laden Sie die AWS CLI von der Webseite mit der [AWS Befehlszeilenschnittstelle](#) herunter.
2. Folgen Sie den Anweisungen zur Installation der AWS CLI und zur Konfiguration der AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Verwenden der AWS CLI mit globalen Datenspeichern

Verwenden Sie die folgenden CLI-Operationen, um mit globalen Datenspeichern zu arbeiten:

- [create-global-replication-group](#)

```
aws elasticache create-global-replication-group \
  --global-replication-group-id-suffix my global datastore \
  --primary-replication-group-id sample-repl-group \
  --global-replication-group-description an optional description of the global
datastore
```

Amazon wendet ElastiCache automatisch ein Präfix auf die globale Datenspeicher-ID an, wenn sie erstellt wird. Jede AWS Region hat ihr eigenes Präfix. Beispielsweise beginnt eine globale

Datenspeicher-ID, die in der Region USA West (Nordkalifornien) erstellt wurde, mit „virxk“ zusammen mit dem von Ihnen angegebenen Suffixnamen. Das Suffix in Kombination mit dem automatisch generierten Präfix garantiert die Eindeutigkeit des globalen Datenspeichernamens über mehrere Regionen hinweg.

In der folgenden Tabelle sind jede AWS Region und ihr globales Datenspeicher-ID-Präfix aufgeführt.

Regionsname/Region	Präfix
Region USA Ost (Ohio) us-east-2	fpkhr
Region USA Ost (Nord-Virginia) us-east-1	ldgnf
Region USA West (Nordkalifornien) us-west-1	virxk
Region USA West (Oregon) us-west-2	sgau
Region Kanada (Zentral) ca-central-1	bxodz
Region Asien-Pazifik (Mumbai) ap-south-1	erpgt
Region Asien-Pazifik (Tokio) ap-northeast-1	qusw

Regionsname/Region	Präfix
Region Asien-Pazifik (Seoul) ap-northeast-2	lfqnh
Region Asien-Pazifik (Osaka) ap-northeast-3	nlapn
Region Asien-Pazifik (Singapur) ap-southeast-1	vlqxn
Region Asien-Pazifik (Sydney) ap-southeast-2	vbgxd
Region Europa (Frankfurt) eu-central-1	iudkw
Region Europa (Irland) eu-west-1	gxeiz
Region Europa (London) eu-west-2	okuqm
Region Europa (Paris) eu-west-3	fgjhi
Region Südamerika (São Paulo) sa-east-1	juxlw
Region China (Peking) cn-north-1	emvgo

Regionsname/Region	Präfix
Region China (Ningxia) cn-northwest-1	ckbem
Region Asien-Pazifik (Hongkong) ap-east-1	knjmp
AWS GovCloud (USA West) us-gov-west-1	sgwui

- [create-replication-group](#)— Verwenden Sie diesen Vorgang, um sekundäre Cluster für einen globalen Datenspeicher zu erstellen, indem Sie den Namen des globalen Datenspeichers für den Parameter angeben. `--global-replication-group-id`

```
aws elasticache create-replication-group \
  --replication-group-id secondary replication group name \
  --replication-group-description "Replication group description" \
  --global-replication-group-id global datastore name
```

Wenn Sie diesen Vorgang aufrufen und einen `--global-replication-group-id` Wert übergeben, ElastiCache werden die Werte für die folgenden Parameter aus der primären Replikationsgruppe der globalen Replikationsgruppe abgeleitet. Übergeben Sie keine Werte für diese Parameter:

"PrimaryClusterId",

"AutomaticFailoverEnabled",

"NumNodeGroups",

"CacheParameterGroupName",

"CacheNodeType",

"Engine",

"EngineVersion",
"CacheSecurityGroupNames",
"EnableTransitEncryption",
"AtRestEncryptionEnabled",
"SnapshotArns",
"SnapshotName"

- [describe-global-replication-groups](#)

```
aws elasticache describe-global-replication-groups \  
  --global-replication-group-id my global datastore \  
  --show-member-info an optional parameter that returns a list of the primary and  
  secondary clusters that make up the global datastore
```

- [modify-global-replication-group](#)

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --automatic-failover-enabled \  
  --cache-node-type node type \  
  --cache-parameter-group-name parameter group name \  
  --engine-version engine version \  
  --apply-immediately \  
  --global-replication-group-description description
```

Engine-übergreifendes Upgrade von Redis auf OSS Valkey für ElastiCache GlobalDataStore

Sie können eine bestehende globale Redis OSS-Replikationsgruppe mithilfe der Konsole, API oder CLI auf Valkey aktualisieren.

Wenn Sie bereits über eine globale Redis OSS-Replikationsgruppe verfügen, können Sie ein Upgrade auf Valkey durchführen, indem Sie die neue Engine und die Engine-Version mit der API angeben. `modify-global-replication-group`

Für Linux, macOS oder Unix:


```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id myGlobalReplGroup \  
  --engine valkey \  
  --apply-immediately \  
  --engine-version 8.0
```

Für Windows:

```
aws elasticache modify-global-replication-group ^  
  --global-replication-group-id myGlobalReplGroup ^  
  --engine valkey ^  
  --apply-immediately ^  
  --engine-version 8.0
```

Wenn Sie eine benutzerdefinierte Cache-Parametergruppe auf die bestehende globale Redis OSS-Replikationsgruppe angewendet haben, die Sie aktualisieren möchten, müssen Sie in der Anfrage auch eine benutzerdefinierte Valkey-Cache-Parametergruppe übergeben. Die benutzerdefinierte Valkey-Eingabeparametergruppe muss dieselben statischen Redis OSS-Parameterwerte haben wie die vorhandene benutzerdefinierte Redis OSS-Parametergruppe.

Für Linux, macOS oder Unix:

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id myGlobalReplGroup \  
  --engine valkey \  
  --engine-version 8.0 \  
  --apply-immediately \  
  --cache-parameter-group-name myParamGroup
```

Für Windows:

```
aws elasticache modify-global-replication-group ^  
  --global-replication-group-id myGlobalReplGroup ^  
  --engine valkey ^  
  --engine-version 8.0 ^  
  --apply-immediately ^  
  --cache-parameter-group-name myParamGroup
```

- [delete-global-replication-group](#)

```
aws elasticache delete-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --retain-primary-replication-group defaults to true
```

- [disassociate-global-replication-group](#)

```
aws elasticache disassociate-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --replication-group-id my secondary cluster \  
  --replication-group-region the AWS Region in which the secondary cluster resides
```

- [failover-global-replication-group](#)

```
aws elasticache failover-replication-group \  
  --global-replication-group-id my global datastore \  
  --primary-region The AWS Region of the primary cluster \  
  --primary-replication-group-id The name of the global datastore, including the suffix.
```

- [increase-node-groups-in-global-replication-group](#)

```
aws elasticache increase-node-groups-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name \  
  --node-group-count 3
```

- [decrease-node-groups-in-global-replication-group](#)

```
aws elasticache decrease-node-groups-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name \  
  --node-group-count 3
```

- [rebalance-shards-in-global-Replikationsgruppe](#)

```
aws elasticache rebalance-shards-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name
```

Verwenden Sie die Hilfe, um alle verfügbaren Befehle ElastiCache für Valkey oder Redis OSS aufzulisten.

```
aws elasticache help
```

Sie können die Hilfe auch verwenden, um einen bestimmten Befehl zu beschreiben und mehr über seine Verwendung zu erfahren:

```
aws elasticache create-global-replication-group help
```

Hohe Verfügbarkeit mit Replikationsgruppen

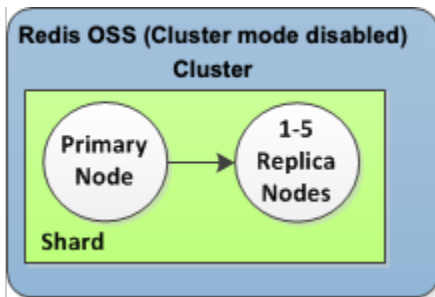
Amazon ElastiCache Valkey- und Redis OSS-Cluster mit einem Knoten sind In-Memory-Einheiten mit eingeschränkten Datenschutzdiensten (AOF). Sollte Ihr Cluster aus irgendeinem Grund ausfallen, verlieren Sie alle Daten des Clusters. Wenn Sie jedoch eine Valkey- oder Redis OSS-Engine ausführen, können Sie 2 bis 6 Knoten zu einem Cluster mit Replikaten gruppieren, wobei 1 bis 5 schreibgeschützte Knoten replizierte Daten des einzelnen primären Knotens der Gruppe enthalten. read/write Wenn aus irgendeinem Grund ein Knoten in diesem Szenario ausfällt, verlieren Sie nicht alle Daten, da sie auf einem oder mehreren Knoten repliziert sind. Aufgrund der Replikationslatenz können einige Daten verloren gehen, wenn der primäre Knoten ausfällt. read/write

Wie in der folgenden Grafik zu sehen ist, ist die Replikationsstruktur in einem Shard (in der API/CLI als Knotengruppe bezeichnet) enthalten, der in einem Valkey- oder Redis-OSS-Cluster enthalten ist. Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) haben immer einen Shard. Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) können bis zu 500 Shards haben, wobei die Daten des Clusters auf die Shards verteilt sind. Sie können einen Cluster mit einer höheren Anzahl an Shards und einer geringeren Anzahl an Replikaten mit bis zu 90 Knoten pro Cluster erstellen. Diese Clusterkonfiguration reicht von 90 Shards und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die Anzahl erlaubter Replikate entspricht.

Das Knoten- oder Shard-Limit kann ElastiCache für Valkey auf maximal 500 pro Cluster und mit ElastiCache Version 5.0.6 oder höher für Redis OSS erhöht werden. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.



Der Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) hat einen Shard und 0 bis 5 Replikatknoten

Wenn für den Cluster mit Replikaten Multi-AZ aktiviert ist, wird bei einem Ausfall des primären Knotens ein automatisches Failover auf eine Read Replica durchgeführt. Da die Daten auf den Replikatknoten asynchron aktualisiert werden, kann die Latenz bei der Aktualisierung der Replikatknoten zu geringfügigem Datenverlust führen. Weitere Informationen finden Sie unter [Minimierung von Fehlern beim Ausführen von Valkey oder Redis OSS](#).

Themen

- [Die OSS-Replikation von Valkey und Redis verstehen](#)
- [Replikation: Valkey- und Redis OSS-Clustermodus deaktiviert oder aktiviert](#)
- [Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS](#)
- [So werden Synchronisation und Backup implementiert](#)
- [Erstellen einer Valkey- oder Redis OSS-Replikationsgruppe](#)
- [Anzeigen der Details einer Replikationsgruppe](#)
- [Suchen von Endpunkten von Replikationsgruppen](#)
- [Ändern einer Replikationsgruppe](#)
- [Löschen einer Replikationsgruppe](#)
- [Ändern der Anzahl von Replikaten](#)
- [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#)

Die OSS-Replikation von Valkey und Redis verstehen

Redis OSS implementiert die Replikation auf zwei Arten:

- Mit einem einzigen Shard, der alle Clusterdaten in jedem Knoten enthält — Valkey oder Redis OSS (Clustermodus deaktiviert)
- Mit Daten, die auf bis zu 500 Shards partitioniert sind — Valkey oder Redis OSS (Clustermodus aktiviert)

Jeder Shard in einer Replikationsgruppe hat einen einzelnen read/write primären Knoten und bis zu 5 schreibgeschützte Replikatknoten. Sie können einen Cluster mit einer höheren Anzahl an Shards und einer geringeren Anzahl an Replikaten mit bis zu 90 Knoten pro Cluster erstellen. Diese Clusterkonfiguration reicht von 90 Shards und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die Anzahl erlaubter Replikate entspricht.

Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Redis OSS-Engine-Version 5.0.6 oder höher ist. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

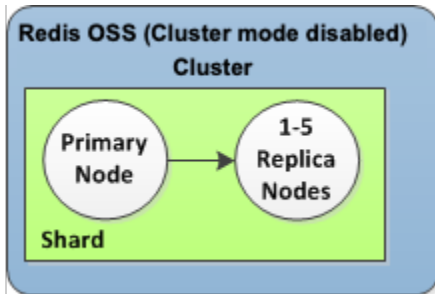
Themen

- [Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#)
- [Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)

Valkey oder Redis OSS (Clustermodus deaktiviert)

Ein Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) hat einen einzelnen Shard, in dem sich eine Sammlung von Knoten befindet: ein primärer read/write Knoten und bis zu fünf sekundäre,

schreibgeschützte Replikatknoten. Jede Read Replica speichert eine Kopie der Daten aus dem primären Knoten des Clusters. Mittels asynchroner Replikationsmechanismen werden die Read Replicas mit dem primären Knoten synchron gehalten. Anwendungen können aus jedem Knoten im Cluster lesen. Anwendungen können nur in den primären Knoten schreiben. Read Replicas verbessern den Lesedurchsatz und schützen vor Datenverlust im Falle eines Knotenausfalls.



Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) mit einem einzigen Shard und Replikatknoten

Sie können Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) mit Replikatknoten verwenden, um Ihre Lösung für leseintensive Anwendungen ElastiCache zu skalieren oder um eine große Anzahl von Clients zu unterstützen, die gleichzeitig aus demselben Cluster lesen.

Alle Knoten in einem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) müssen sich in derselben Region befinden.

Wenn Sie eine Read Replica zu einem Cluster hinzufügen, werden alle Daten vom primären Knoten auf den neuen Knoten kopiert. Ab diesem Punkt werden Änderung jedes Mal, wenn Daten in den primären Knoten geschrieben werden, synchron auf alle Read Replicas übertragen.

Um die Fehlertoleranz zu verbessern und Ausfallzeiten beim Schreiben zu reduzieren, aktivieren Sie Multi-AZ mit automatischem Failover für Ihren Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) mit Replikaten. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS](#).

Sie können die Rollen der Knoten innerhalb des Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) ändern, wobei der primäre Knoten und eines der Replikate die Rollen austauschen. Sie könnten sich aus Gründen der Leistungsoptimierung hierfür entscheiden. Bei einer Webanwendung mit der intensivsten Schreibaktivität könnten Sie z. B. den Knoten mit der niedrigsten Netzwerklatenz wählen. Weitere Informationen finden Sie unter [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#).

Valkey oder Redis OSS (Clustermodus aktiviert)

Ein Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) besteht aus 1 bis 500 Shards (API/CLI: Knotengruppen). Jeder Shard verfügt über einen primären Knoten und bis zu fünf schreibgeschützte Replikatknoten. Die Konfiguration reicht von 90 Shards und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die Anzahl erlaubter Replikate entspricht.

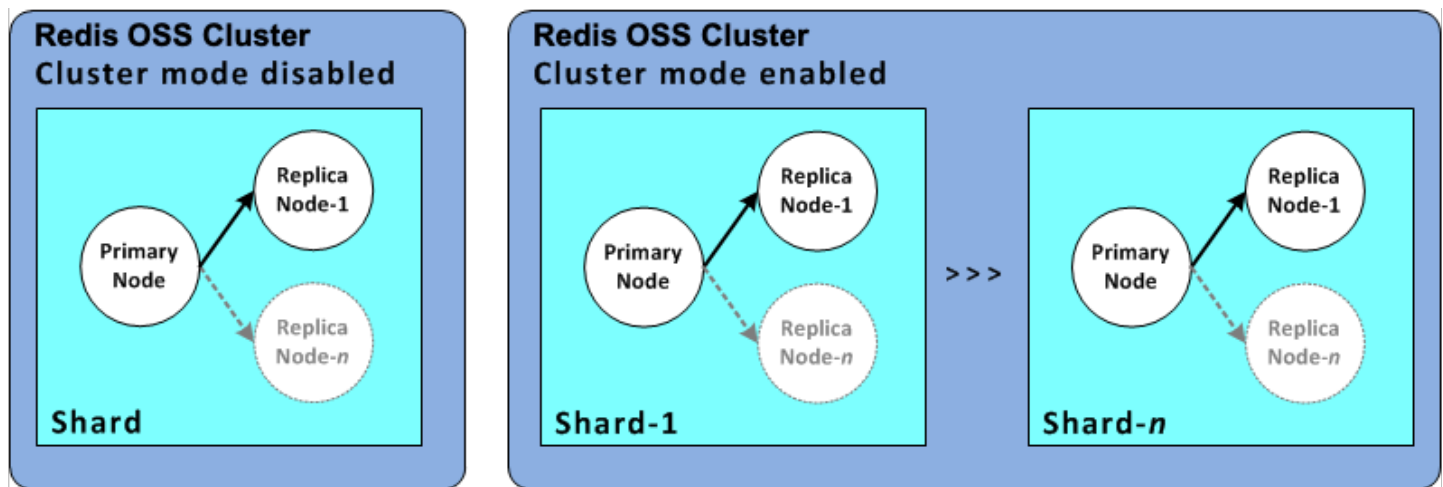
Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Engine-Version Valkey 7.2 und höher oder Redis OSS 5.0.6 und höher ist. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

Jede Read Replica in einem Shard speichert eine Kopie der Daten aus dem primären Knoten des Shards. Mittels asynchroner Replikationsmechanismen werden die Read Replicas mit dem primären Knoten synchron gehalten. Anwendungen können aus jedem Knoten im Cluster lesen. Anwendungen können nur in die primären Knoten schreiben. Read Replicas verbessern die Leseskalierbarkeit und schützen vor Datenverlust. Die Daten werden auf die Shards in einem Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) aufgeteilt.

Anwendungen verwenden den Konfigurationsendpunkt des Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert), um eine Verbindung mit den Knoten im Cluster herzustellen. Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).



Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mit mehreren Shards und Replikatknoten

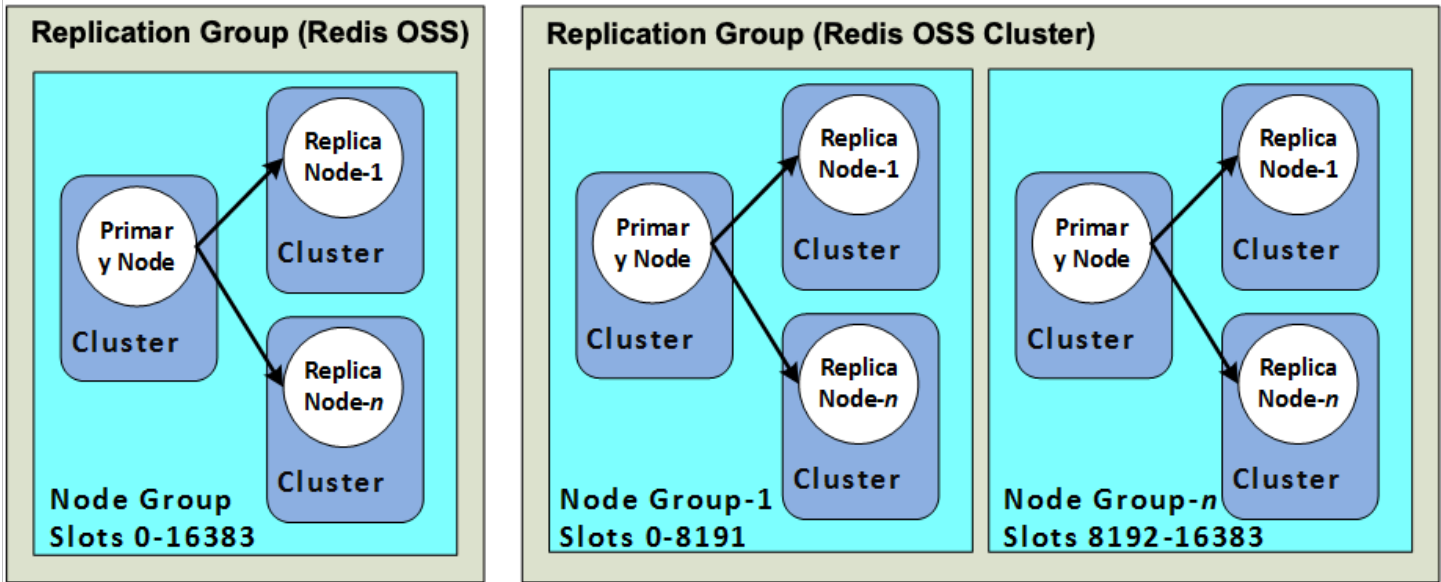
Alle Knoten in einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) müssen sich in derselben Region befinden. Zur Verbesserung der Fehlertoleranz können Sie sowohl primären Knoten als auch Read Replicas in mehreren Availability Zones innerhalb dieser Region bereitstellen.

Derzeit weisen die Funktionen von Valkey oder Redis OSS (Clustermodus aktiviert) einige Einschränkungen auf.

- Keine der Replikatknoten können zum primären Knoten manuell heraufgestuft werden.

Replikation: Valkey- und Redis OSS-Clustermodus deaktiviert oder aktiviert

Ab Valkey 7.2 und Redis OSS Version 3.2 haben Sie die Möglichkeit, einen von zwei unterschiedlichen Clustertypen (API/CLI: replication groups). A Valkey or Redis OSS (cluster mode disabled) cluster always has a single shard (API/CLI: Knotengruppe) mit bis zu 5 Read Replica-Knoten zu erstellen. Ein Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) hat bis zu 500 Shards mit jeweils 1 bis 5 Read Replica-Knoten.



Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) und Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert)

In der folgenden Tabelle werden wichtige Unterschiede zwischen Valkey- oder Redis OSS-Clustern (Clustermodus deaktiviert) und Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert) zusammengefasst.

Vergleich von Valkey- oder Redis OSS-Clustern (Clustermodus deaktiviert) und Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert)

Funktion	Valkey oder Redis OSS (Clustermodus deaktiviert)	Valkey oder Redis OSS (Clustermodus aktiviert)
Anpassbar	Ja. Unterstützt das Hinzufügen und Löschen von Replikaten und das Hochskalieren des Knotentyps.	Begrenzt. Weitere Informationen erhalten Sie unter Versionsverwaltung für ElastiCache und Skalierung von Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert) .
Datenpartitionierung	Nein	Ja
Shards	1	1 bis 500
Read Replicas	0 bis 5	0 bis 5 pro Shard.

Funktion	Valkey oder Redis OSS (Clustermodus deaktiviert)	Valkey oder Redis OSS (Clustermodus aktiviert)
	 Important Wenn Sie bei einem Knotenausfall über keine Replikate verfügen, gehen alle Ihre Daten verloren.	 Important Wenn Sie bei einem Knotenausfall über keine Replikate verfügen, gehen alle Ihre Daten in diesem Shard verloren.
Multi-AZ	Ja, mit mindestens 1 Replikat. Optional. Standardmäßig aktiviert	Ja Optional. Standardmäßig aktiviert
Snapshots (Backups)	Ja, es wird eine einzelne .rdb-Datei erstellt.	Ja, es wird für jeden Shard eine eindeutige .rdb-Datei erstellt.
Wiederherstellung	Ja, mit einer einzigen .rdb-Datei aus einem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert).	Ja, mit .rdb-Dateien aus einem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) oder einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert).
Unterstützt von	Alle Valkey- und Redis OSS-Versionen	Alle Valkey-Versionen und Redis OSS 3.2 und folgende
Engine aktualisierbar	Ja, mit einigen Einschränkungen. Weitere Informationen finden Sie unter Versionsverwaltung für ElastiCache .	Ja, mit einigen Einschränkungen. Weitere Informationen finden Sie unter Versionsverwaltung für ElastiCache .

Funktion	Valkey oder Redis OSS (Clustermodus deaktiviert)	Valkey oder Redis OSS (Clustermodus aktiviert)
Verschlüsselung	Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis OSS-Versionen) und 4.0.10 und höher.	Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis OSS-Versionen) und 4.0.10 und höher.
HIPAA-konform	Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis OSS-Versionen) und 4.0.10 und höher.	Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis OSS-Versionen) und 4.0.10 und höher.
Zu PCI DSS konform	Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis OSS-Versionen) und 4.0.10 und höher.	Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis OSS-Versionen) und 4.0.10 und höher.
Online-Resharding	N/A	Version 3.2.10 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis OSS-Versionen) und höher.

Welchen Cluster soll ich wählen?

Wenn Sie zwischen Valkey oder Redis OSS (Cluster-Modus deaktiviert) oder Valkey oder Redis OSS (Cluster-Modus aktiviert) wählen, sollten Sie die folgenden Faktoren berücksichtigen:

- Skalierung im Vergleich zu Partitionierung – Geschäftliche Anforderungen ändern sich. Sie müssen entweder für Spitzenbedarf vorsorgen oder bei Bedarfsänderungen skalieren. Valkey oder Redis OSS (Clustermodus deaktiviert) unterstützt die Skalierung. Sie können die Lesekapazität durch Hinzufügen oder Löschen von Replikatknoten skalieren. Sie können die Kapazität auch nach oben skalieren, indem Sie auf einen größeren Knotentyp skalieren. Beide dieser Operationen sind

zeitaufwendig. Weitere Informationen finden Sie unter [Skalierung von Replikatknoten für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#).

Valkey oder Redis OSS (Clustermodus aktiviert) unterstützt die Partitionierung Ihrer Daten auf bis zu 500 Knotengruppen. Sie können die Anzahl der Shards dynamisch ändern, wenn sich Ihre Geschäftsanforderungen ändern. Ein Vorteil der Partitionierung liegt darin, dass Ihre Arbeitslast über eine größere Anzahl von Endpunkten verteilt wird. Dadurch lassen sich Engpässe in Zeiten mit höchster Auslastung reduzieren. Zusätzlich dazu können Sie einen größeren Datensatz bewältigen, da die Daten über mehrere Server verteilt werden können. Informationen zur Skalierung Ihrer Partitionen finden Sie unter [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#).

- Knotengröße vs. Anzahl der Knoten — Da ein Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) nur einen Shard hat, muss der Knotentyp groß genug sein, um alle Daten des Clusters sowie den erforderlichen Overhead aufzunehmen. Da Sie Ihre Daten andererseits auf mehrere Shards partitionieren können, wenn Sie einen Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) verwenden, können die Knotentypen kleiner sein, obwohl Sie mehr davon benötigen. Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).
- Lesevorgänge versus Schreibvorgänge — Wenn die Hauptlast auf Ihrem Cluster darin besteht, dass Anwendungen Daten lesen, können Sie einen Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) skalieren, indem Sie Read Replicas hinzufügen und löschen. Es gibt jedoch maximal 5 Read Replicas. Wenn die Last auf Ihrem Cluster viele Schreibvorgänge erfordert, können Sie von den zusätzlichen Schreibendpunkten eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) mit mehreren Shards profitieren.

Unabhängig von dem ausgewählten, zu implementierenden Clustertyp ist es wichtig, einen Knotentyp zu wählen, der für Ihre derzeitigen und zukünftigen Anforderungen angemessen ist.

Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS

Es gibt eine Reihe von Fällen, in denen OSS ElastiCache für Valkey und Redis möglicherweise einen Primärknoten austauschen muss. Dazu gehören bestimmte Arten von geplanten Wartungsarbeiten und der unwahrscheinliche Fall eines Ausfalls eines Primärknotens oder einer Availability Zone.

Dieser Austausch führt zu einer gewissen Ausfallzeit für den Cluster, aber wenn Multi-AZ aktiviert ist, wird die Ausfallzeit minimiert. Die Rolle des primären Knotens wird automatisch auf eines der Read Replicas übertragen. Es ist nicht erforderlich, einen neuen Primärknoten zu erstellen und bereitzustellen, da dies transparent ElastiCache gehandhabt wird. Dieser Failover und die Replikatheraufstufung stellen sicher, dass Sie weiter in den neuen primären Knoten schreiben können, sobald die Heraufstufung abgeschlossen wurde.

ElastiCache verbreitet auch den DNS-Namen (Domain Name Service) des beworbenen Replikats. Auf diese Weise ist in Ihrer Anwendung, falls sie in den primären Endpunkt schreibt, keine Endpunktänderung erforderlich. Wenn Sie aus individuellen Endpunkten lesen, müssen Sie den Leseendpunkt des zum primären Knoten heraufgestuften Replikats in den Endpunkt des neuen Replikats ändern.

Im Falle eines geplanten Knotenaustauschs, der aufgrund von Wartungs- oder Self-Service-Aktualisierungen eingeleitet wird, beachten Sie Folgendes:

- Bei Valkey- und Redis OSS-Clustern werden die geplanten Knotenersetzungen abgeschlossen, während der Cluster eingehende Schreib Anforderungen bearbeitet.
- Bei deaktivierten Clustern im Valkey- und Redis OSS-Clustermodus mit aktiviertem Multi-AZ, die auf der Engine 5.0.6 oder höher ausgeführt werden, werden die geplanten Knotenersetzungen abgeschlossen, während der Cluster eingehende Schreib Anforderungen bearbeitet.
- Bei deaktivierten Clustern im Valkey- und Redis OSS-Clustermodus mit aktiviertem Multi-AZ, die auf der Engine 4.0.10 oder früher ausgeführt werden, stellen Sie möglicherweise eine kurze Schreibunterbrechung im Zusammenhang mit DNS-Updates fest. Diese Unterbrechung kann bis zu einigen Sekunden dauern. Dieser Vorgang ist viel schneller als die Neuerstellung und Bereitstellung einer neuen Primärdatenbank, was geschieht, wenn Sie Multi-AZ nicht aktivieren.

Sie können Multi-AZ mithilfe der ElastiCache Management Console, der oder der API aktivieren.
AWS CLI ElastiCache

Die Aktivierung von ElastiCache Multi-AZ auf Ihrem Valkey- oder Redis OSS-Cluster (in der API und CLI, Replikationsgruppe) verbessert Ihre Fehlertoleranz. Dies gilt insbesondere in Fällen, in denen der read/write primäre Cluster Ihres Clusters nicht mehr erreichbar ist oder aus irgendeinem Grund ausfällt. Multi-AZ wird nur auf Valkey- und Redis OSS-Clustern mit mehr als einem Knoten in jedem Shard unterstützt.

Themen

- [Aktivieren von Multi-AZ](#)
- [Fehlerszenarien mit Multi-AZ-Antworten](#)
- [Testen des automatischen Failovers](#)
- [Einschränkungen bei Multi-AZ](#)

Aktivieren von Multi-AZ

Sie können Multi-AZ aktivieren, wenn Sie einen Cluster (API oder CLI, Replikationsgruppe) mithilfe der ElastiCache Konsole oder der ElastiCache API erstellen oder ändern. AWS CLI

Sie können Multi-AZ nur auf Valkey- oder Redis OSS-Clustern (Clustermodus deaktiviert) aktivieren, für die mindestens eine Read Replica verfügbar ist. Cluster ohne Read Replicas bieten keine hohe Verfügbarkeit oder Fehlertoleranz. Weitere Informationen zum Erstellen eines Clusters mit Replikation finden Sie unter [Erstellen einer Valkey- oder Redis OSS-Replikationsgruppe](#). Weitere Informationen zum Hinzufügen einer Read Replica zu einem Cluster mit Replikation finden Sie unter [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

Themen

- [Aktivieren von Multi-AZ \(Konsole\)](#)
- [Aktivieren von Multi-AZ \(AWS CLI\)](#)
- [Multi-AZ \(ElastiCache API\) aktivieren](#)

Aktivieren von Multi-AZ (Konsole)

Sie können Multi-AZ mithilfe der ElastiCache Konsole aktivieren, wenn Sie einen neuen Valkey- oder Redis OSS-Cluster erstellen oder indem Sie einen vorhandenen Cluster mit Replikation ändern.

Multi-AZ ist standardmäßig auf Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert) aktiviert.

 Important

ElastiCache aktiviert Multi-AZ automatisch nur dann, wenn der Cluster in allen Shards mindestens ein Replikat in einer anderen Availability Zone als der primären enthält.

Multi-AZ wird aktiviert, wenn ein Cluster mithilfe der Konsole erstellt wird ElastiCache

Weitere Informationen zu diesem Vorgang finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#). Achten Sie darauf, dass mehr als ein Replikat vorhanden und Multi-AZ aktiviert ist.

Aktivieren von Multi-AZ auf einem vorhandenen Cluster (Konsole)

Weitere Informationen zu diesem Prozess finden Sie unter „Ändern eines Clusters“ [Unter Verwendung der ElastiCache AWS-Managementkonsole](#).

Aktivieren von Multi-AZ (AWS CLI)

Im folgenden Codebeispiel wird Multi-AZ für die AWS CLI Replikationsgruppe aktiviert. `redis12`

 Important

Die Replikationsgruppe `redis12` muss bereits vorhanden sein und mindestens eine verfügbare Read Replica besitzen.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id redis12 \  
  --automatic-failover-enabled \  
  --multi-az-enabled \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id redis12 ^  
  --automatic-failover-enabled ^
```

```
--multi-az-enabled ^  
--apply-immediately
```

Die JSON-Ausgabe dieses Befehls sollte in etwa folgendermaßen aus.

```
{  
  "ReplicationGroup": {  
    "Status": "modifying",  
    "Description": "One shard, two nodes",  
    "NodeGroups": [  
      {  
        "Status": "modifying",  
        "NodeGroupMembers": [  
          {  
            "CurrentRole": "primary",  
            "PreferredAvailabilityZone": "us-west-2b",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis12-001.v5r9dc.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis12-001"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis12-002.v5r9dc.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis12-002"  
          }  
        ],  
        "NodeGroupId": "0001",  
        "PrimaryEndpoint": {  
          "Port": 6379,  
          "Address": "redis12.v5r9dc.ng.0001.usw2.cache.amazonaws.com"  
        }  
      }  
    ],  
  },  
}
```



```
    "ReplicationGroupId": "redis12",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabling",
    "MultiAZ": "enabled",
    "SnapshotWindow": "07:00-08:00",
    "SnapshottingClusterId": "redis12-002",
    "MemberClusters": [
        "redis12-001",
        "redis12-002"
    ],
    "PendingModifiedValues": {}
}
```

Weitere Informationen zu diesen Themen finden Sie in der AWS CLI -Befehlsreferenz:

- [create-cache-cluster](#)
- [create-replication-group](#)
- [modify-replication-group](#) in der AWS CLI Befehlsreferenz.

Multi-AZ (ElastiCache API) aktivieren

Das folgende Codebeispiel verwendet die ElastiCache API, um Multi-AZ für die Replikationsgruppe zu aktivieren. `redis12`

Note

Zur Verwendung dieses Beispiels muss die Replikationsgruppe `redis12` bereits vorhanden sein und mindestens ein verfügbares Lesereplikat besitzen.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&ApplyImmediately=true
&AutoFailover=true
&MultiAZEnabled=true
&ReplicationGroupId=redis12
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
```

```
&Timestamp=20140401T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen zu diesen Themen finden Sie in der ElastiCache -API-Befehlsreferenz:

- [CreateCacheCluster](#)
- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

Fehlerszenarien mit Multi-AZ-Antworten

Vor der Einführung von Multi-AZ wurden die ausgefallenen Knoten eines Clusters ElastiCache erkannt und ersetzt, indem der ausgefallene Knoten neu erstellt und erneut bereitgestellt wurde. Wenn Sie Multi-AZ aktivieren, wird bei einem ausgefallenen primärer Knoten ein Failover auf das Replikat mit der geringsten Replikationsverzögerung durchgeführt. Das ausgewählte Replikat wird automatisch zum primären Knoten heraufgestuft. Dies ist sehr viel schneller, als einen neuen primären Knoten zu erstellen und neu bereitzustellen. Bei diesem Vorgang dauert gewöhnlich nur wenige Sekunden, bis Sie wieder in den Cluster schreiben können.

Wenn Multi-AZ aktiviert ist, wird der Status des ElastiCache Primärknotens kontinuierlich überwacht. Sollte der primäre Knoten ausfallen, wird abhängig von der Art des Ausfalls eine der folgenden Aktionen durchgeführt.

Themen

- [Fehlerszenarien, wenn nur der Primärknoten ausfällt](#)
- [Fehlerszenarien, wenn der Primärknoten und einige Read Replicas \(Lesereplikate\) fehlschlagen](#)
- [Fehlerszenarien, wenn der gesamte Cluster ausfällt](#)

Fehlerszenarien, wenn nur der Primärknoten ausfällt

Wenn nur der primäre Knoten ausfällt, wird das Read Replica mit der geringsten Replikationsverzögerung zum primären Cluster heraufgestuft. Ein Ersatz-Read Replica wird dann in derselben Availability Zone wie der ausgefallene primäre Knoten erstellt und bereitgestellt .

Wenn nur der Primärknoten ausfällt, geht ElastiCache Multi-AZ wie folgt vor:

1. Der ausgefallene primäre Knoten wird in den Offline-Zustand versetzt.
2. Die Read Replica mit der geringsten Replikationsverzögerung wird zum primären Knoten heraufgestuft.

Die Schreibvorgänge können fortgesetzt werden, sobald der Vorgang der Heraufstufung abgeschlossen ist. Dies dauert in der Regel nur einige Sekunden. Wenn Ihre Anwendung auf den primären Endpunkt schreibt, ist keine Endpunktänderung für Schreib- oder Lesevorgänge erforderlich, da ElastiCache den DNS-Namen des heraufgestuften Replikats überträgt.

3. Ein Ersatz-Read Replica wird gestartet und bereitgestellt.

Die Ersatz-Read Replica wird in der Availability Zone gestartet, in der sich der ausgefallene Knoten befand. Die Verteilung der Knoten bleibt daher erhalten.

4. Die anderen Replikate werden mit dem neuen primären Knoten synchronisiert.

Nachdem das neue Replikat verfügbar ist, beachten Sie die folgenden Effekte:

- Primärer Endpunkt – Sie müssen keine Änderungen an Ihrer Anwendung vornehmen, da der DNS-Name des neuen primären Knotens an den primären Endpunkt weitergegeben wird.
- Lese-Endpunkt – Der Lese-Endpunkt wird automatisch aktualisiert, sodass er auf die neuen Replikatknoten verweist.

Weitere Informationen zum Suchen der Endpunkte eines Clusters finden Sie in den folgenden Themen:

- [Finden Sie die Endpunkte eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden der Endpunkte für Valkey- oder Redis OSS-Replikationsgruppen \(\)AWS CLI](#)
- [Suche nach Endpunkten für Valkey- oder Redis OSS-Replikationsgruppen \(API\) ElastiCache](#)

Fehlerszenarien, wenn der Primärknoten und einige Read Replicas (Lesereplikate) fehlschlagen

Bei Ausfall des primären Clusters und mindestens einer Read Replica wird das verfügbare Replikat mit der geringsten Replikationsverzögerung zum primären Cluster heraufgestuft. Zudem werden in denselben Availability Zones, in der sich die ausgefallenen Knoten und das zum primären Cluster heraufgestufte Replikat befanden, neue Read Replicas erstellt und bereitgestellt.

Wenn der Primärknoten und einige Read Replicas ausfallen, geht ElastiCache Multi-AZ wie folgt vor:

1. Der ausgefallene primäre Knoten und ausgefallenen die Read Replicas werden in den Offline-Zustand versetzt.
2. Das verfügbare Replikat mit der geringsten Replikationsverzögerung wird zum primären Knoten heraufgestuft.

Die Schreibvorgänge können fortgesetzt werden, sobald der Vorgang der Heraufstufung abgeschlossen ist. Dies dauert in der Regel nur einige Sekunden. Wenn Ihre Anwendung auf

den primären Endpunkt schreibt, müssen Sie den Endpunkt für Schreibvorgänge nicht ändern. ElastiCache verbreitet den DNS-Namen des beworbenen Replikats.

3. Ersatzreplikate werden erstellt und bereitgestellt.

Die Ersatzreplikate werden in den Availability Zones der ausgefallenen Knoten erstellt, sodass die Verteilung der Knoten erhalten bleibt.

4. Alle Cluster werden mit dem neuen primären Knoten synchronisiert.

Nehmen Sie die folgenden Änderungen an Ihrer Anwendung vor, wenn die neuen Knoten verfügbar sind:

- Primärer Endpunkt – Nehmen Sie keine Änderungen an Ihrer Anwendung vor. Der DNS-Name des neuen Primärknotens wird an den primären Endpunkt weitergegeben.
- Lese-Endpunkt – Der Lese-Endpunkt wird automatisch so aktualisiert, dass er auf die neuen Replikatknoten verweist.

Weitere Informationen zum Suchen der Endpunkte einer Replikationsgruppe finden Sie in den folgenden Themen:

- [Finden Sie die Endpunkte eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden der Endpunkte für Valkey- oder Redis OSS-Replikationsgruppen \(\)AWS CLI](#)
- [Suche nach Endpunkten für Valkey- oder Redis OSS-Replikationsgruppen \(API\) ElastiCache](#)

Fehlerszenarien, wenn der gesamte Cluster ausfällt

Bei einem umfassenden Ausfall werden in denselben Availability Zones, der sich die Originalknoten befanden, alle Knoten neu erstellt und bereitgestellt.

In diesem Szenario gehen alle Daten im Cluster aufgrund des Ausfalls eines jeden Knotens im Cluster verloren. Ein solches Ereignis ist selten.

Wenn der gesamte Cluster ausfällt, geht ElastiCache Multi-AZ wie folgt vor:

1. Der ausgefallene primäre Knoten und die Read Replicas werden in den Offline-Zustand versetzt.

2. Es wird ein primäre Ersatzknoten erstellt und bereitgestellt.
3. Ersatzreplikate werden erstellt und bereitgestellt.

Die Ersetzungen werden in den Availability Zones der ausgefallenen Knoten erstellt, sodass die Verteilung der Knoten erhalten bleibt.

Da der gesamte Cluster ausgefallen ist, kam es zu Datenverlust. Alle neuen Knoten werden kalt gestartet.

Da jeder Ersatzknoten denselben Endpunkt wie der Knoten hat, der durch ihn ersetzt wird, müssen in Ihrer Anwendung keine Endpunktänderungen vorgenommen werden.

Weitere Informationen zum Suchen der Endpunkte einer Replikationsgruppe finden Sie in den folgenden Themen:

- [Finden Sie die Endpunkte eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden der Endpunkte für Valkey- oder Redis OSS-Replikationsgruppen \(\)AWS CLI](#)
- [Suche nach Endpunkten für Valkey- oder Redis OSS-Replikationsgruppen \(API\) ElastiCache](#)

Es wird empfohlen, den primären Knoten und die Read Replicas in verschiedenen Availability Zones zu erstellen. Dadurch wird der Grad der Fehlertoleranz erhöht.

Testen des automatischen Failovers

Nachdem Sie den automatischen Failover aktiviert haben, können Sie ihn mithilfe der ElastiCache Konsole AWS CLI, der und der ElastiCache API testen.

Beim Testen ist Folgendes zu beachten:

- Sie können diesen Vorgang verwenden, um das automatische Failover auf bis zu 15 Shards (in der ElastiCache API als Knotengruppen bezeichnet AWS CLI) in einem beliebigen Zeitraum von 24 Stunden zu testen.
- Wenn Sie diese Operation für Shards in verschiedenen Clustern (in der API und CLI als Replikationsgruppen bezeichnet) aufrufen, können diese Aufrufe gleichzeitig erfolgen.
- In einigen Fällen können Sie diesen Vorgang mehrmals auf verschiedenen Shards in derselben Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) aufrufen. In solchen Fällen muss die erste Knotenersetzung abgeschlossen werden, bevor ein nachfolgender Aufruf ausgeführt werden kann.
- Um festzustellen, ob der Knotenaustausch abgeschlossen ist, überprüfen Sie die Ereignisse mithilfe der ElastiCache Amazon-Konsole AWS CLI, der oder der ElastiCache API. Suchen Sie nach den folgenden Ereignissen im Zusammenhang mit automatischem Failover, die hier nach der Wahrscheinlichkeit ihres Auftretens aufgelistet werden:
 1. Meldung der Replikationsgruppe: `Test Failover API called for node group <node-group-id>`
 2. Meldung des Cache-Clusters: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
 3. Meldung der Replikationsgruppe: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
 4. Meldung des Cache-Clusters: `Recovering cache nodes <node-id>`
 5. Meldung des Cache-Clusters: `Finished recovery for cache nodes <node-id>`

Weitere Informationen finden Sie hier:

- [ElastiCache Ereignisse anzeigen](#) im ElastiCache -Benutzerhandbuch
- [DescribeEvents](#) in der ElastiCache -API-Referenz
- [describe-events](#) in der AWS CLI -Befehlsreferenz.
- Diese API wurde entwickelt, um das Verhalten Ihrer Anwendung im Falle eines ElastiCache Failovers zu testen. Sie wurde nicht als Betriebstool zum Einleiten eines Failovers konzipiert, um

ein Problem mit dem Cluster zu beheben. Darüber hinaus AWS kann diese API unter bestimmten Bedingungen, z. B. bei großen Betriebsereignissen, blockiert werden.

Themen

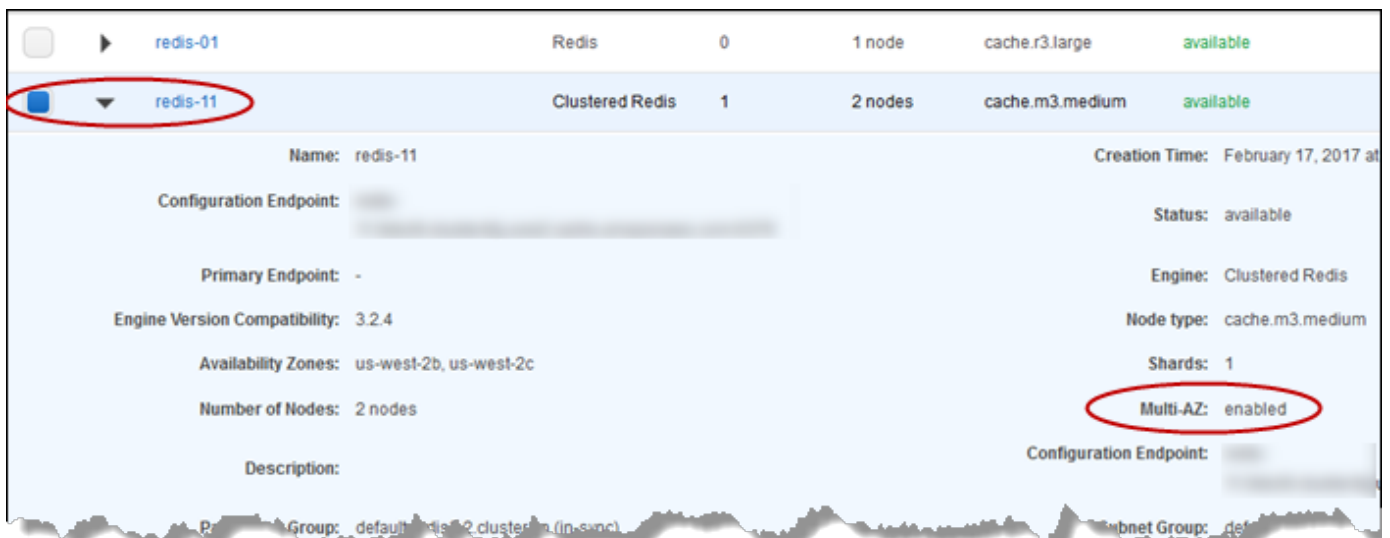
- [Testen des automatischen Failovers mit dem AWS-Managementkonsole](#)
- [Testen Sie den automatischen Failover mit dem AWS CLI](#)
- [Testen Sie den automatischen Failover mithilfe der API ElastiCache](#)

Testen des automatischen Failovers mit dem AWS-Managementkonsole

Verwenden Sie das folgende Verfahren, um das automatische Failover mit der Konsole zu testen.

So testen Sie das automatische Failover

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS aus.
3. Wählen Sie in der Clusterliste das Feld links neben dem Cluster aus, den Sie testen möchten. Dieser Cluster muss mindestens einen Read Replica-Knoten enthalten.
4. Bestätigen Sie im Bereich Details, dass dieser Cluster Multi-AZ-fähig ist. Wenn der Cluster nicht Multi-AZ-fähig ist, wählen Sie einen anderen Cluster aus oder bearbeiten Sie diesen Cluster so, dass Multi-AZ aktiviert wird. Weitere Informationen finden Sie unter [Unter Verwendung der ElastiCache AWS-Managementkonsole](#).



5. Wählen Sie für Valkey oder Redis OSS (Clustermodus deaktiviert) den Namen des Clusters.

Gehen Sie für Valkey oder Redis OSS (Clustermodus aktiviert) wie folgt vor:

- a. Wählen Sie den Cluster-Namen aus.
 - b. Wählen Sie auf der Seite Shards für den Shard (in API und CLI als Knotengruppe bezeichnet), auf dem das Failover getestet werden soll, den Namen des Shards aus.
6. Wählen Sie auf der Seite „Nodes“ Failover Primary.
 7. Wählen Sie Continue, um ein Failover des primären Knotens auszuführen, oder wählen Sie Cancel, um die Operation ohne ein Failover des primären Knotens abubrechen.

Während des Failover-Vorgangs zeigt die Konsole den Status des Knotens weiterhin als available an. Um den Status des Failover-Tests zu verfolgen, wählen Sie im Navigationsbereich der Konsole Events aus. Suchen Sie auf der Registerkarte Events nach Ereignissen, für die angegeben wird, dass Ihr Failover gestartet (Test Failover API called) und abgeschlossen (Recovery completed) wurde.

Testen Sie den automatischen Failover mit dem AWS CLI

Mithilfe dieses Vorgangs können Sie das automatische Failover auf jedem Multi-AZ-fähigen Cluster testen. `AWS CLI test-failover`

Parameter

- `--replication-group-id` – Erforderlich. Die Replikationsgruppe (auf der Konsole als Cluster bezeichnet), die getestet werden soll.
- `--node-group-id` – Erforderlich. Der Name der Knotengruppe, auf der das automatische Failover getestet werden soll. Sie können maximal 15 Knotengruppen in einem fortlaufenden Zeitraum von 24 Stunden testen.

Im folgenden Beispiel wird der verwendet AWS CLI , um das automatische Failover für die Knotengruppe `redis00-0003` im Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) zu testen. `redis00`

Example Testen des automatischen Failovers

Für Linux, macOS oder Unix:

```
aws elasticache test-failover \  
--replication-group-id redis00 \  
--node-group-id redis00-0003
```

Für Windows:

```
aws elasticache test-failover ^  
--replication-group-id redis00 ^  
--node-group-id redis00-0003
```

Die Ausgabe des vorhergehenden Befehls sieht in etwa wie folgt aus.

```
{  
  "ReplicationGroup": {  
    "Status": "available",  
    "Description": "1 shard, 3 nodes (1 + 2 replicas)",  
    "NodeGroups": [  
      {  
        "Status": "available",  
        "NodeGroupMembers": [  
          {  
            "CurrentRole": "primary",  
            "PreferredAvailabilityZone": "us-west-2c",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis1x3-001.7ekv3t.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis1x3-001"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis1x3-002.7ekv3t.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis1x3-002"  
          }  
        ]  
      }  
    ]  
  }  
}
```

```

        {
            "CurrentRole": "replica",
            "PreferredAvailabilityZone": "us-west-2b",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Port": 6379,
                "Address":
"redis1x3-003.7ekv3t.0001.usw2.cache.amazonaws.com"
            },
            "CacheClusterId": "redis1x3-003"
        },
        {
            "NodeGroupId": "0001",
            "PrimaryEndpoint": {
                "Port": 6379,
                "Address": "redis1x3.7ekv3t.ng.0001.usw2.cache.amazonaws.com"
            }
        }
    ],
    "ClusterEnabled": false,
    "ReplicationGroupId": "redis1x3",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "MultiAZ": "enabled",
    "SnapshotWindow": "11:30-12:30",
    "SnapshottingClusterId": "redis1x3-002",
    "MemberClusters": [
        "redis1x3-001",
        "redis1x3-002",
        "redis1x3-003"
    ],
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled",
    "PendingModifiedValues": {}
}
}

```

Verwenden Sie den Vorgang, um den Fortschritt Ihres Failovers zu verfolgen. AWS CLI `describe-events`

Weitere Informationen finden Sie hier:

- [test-failover](#) in der AWS CLI -Befehlsreferenz.

- [describe-events](#) in der AWS CLI -Befehlsreferenz.

Testen Sie den automatischen Failover mithilfe der API ElastiCache

Mithilfe der ElastiCache API-Operation können Sie das automatische Failover auf jedem Cluster testen, für den Multi-AZ aktiviert ist. `TestFailover`

Parameter

- `ReplicationGroupId` – Erforderlich. Die Replikationsgruppe (auf der Konsole als Cluster bezeichnet), die getestet werden soll.
- `NodeGroupId` – Erforderlich. Der Name der Knotengruppe, auf der das automatische Failover getestet werden soll. Sie können maximal 15 Knotengruppen in einem fortlaufenden Zeitraum von 24 Stunden testen.

Das folgende Beispiel testet das automatische Failover auf der Knotengruppe `redis00-0003` in der Replikationsgruppe (auf der Konsole im Cluster) `redis00`.

Example Testen des automatischen Failovers

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=TestFailover  
&NodeGroupId=redis00-0003  
&ReplicationGroupId=redis00  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20140401T192317Z  
&X-Amz-Credential=<credential>
```

Verwenden Sie den ElastiCache `DescribeEvents` API-Vorgang, um den Fortschritt Ihres Failovers zu verfolgen.

Weitere Informationen finden Sie hier:

- [TestFailover](#) in der ElastiCache API-Referenz
- [DescribeEvents](#) in der ElastiCache API-Referenz

Einschränkungen bei Multi-AZ

Beachten Sie die folgenden Einschränkungen für Multi-AZ:

- Multi-AZ wird auf Valkey und auf Redis OSS Version 2.8.6 und höher unterstützt.
- Multi-AZ wird auf T1-Knotentypen nicht unterstützt.
- Die OSS-Replikation von Valkey und Redis erfolgt asynchron. Daher kann es passieren, dass bei dem Failover eines primären Clusters auf ein Replikat aufgrund der Replikationsverzögerung eine kleine Datenmenge verloren geht.

Bei der Auswahl des Replikats, das zum primären Replikat heraufgestuft werden soll, wird das Replikat mit ElastiCache der geringsten Verzögerung bei der Replikation ausgewählt. Es wird also das neueste Replikat ausgewählt. Hierdurch kann die Menge verlorener Daten reduziert werden. Das Replikat mit der geringsten Replikationsverzögerung kann sich in der gleichen oder in einer anderen Availability Zone als der ausgefallene primäre Knoten befinden.

- Wenn Sie Read Replicas auf Valkey- oder Redis OSS-Clustern mit deaktiviertem Clustermodus manuell zu primär heraufstufen, können Sie dies nur tun, wenn Multi-AZ und automatisches Failover deaktiviert sind. So stufen Sie eine Read Replica zum primären Knoten herauf:
 1. Deaktivieren Sie Multi-AZ auf dem Cluster.
 2. Deaktivieren Sie das automatische Failover auf dem Cluster. Sie können dies über die Konsole tun, indem Sie das Kontrollkästchen Auto Failover für die Replikationsgruppe deaktivieren. Sie können dies auch tun, AWS CLI indem Sie die `AutomaticFailoverEnabled` Eigenschaft `false` beim Aufrufen des `ModifyReplicationGroup` Vorgangs auf setzen.
 3. Stufen Sie das Lesereplikat zum primären Knoten herauf.
 4. Multi-AZ wieder aktivieren.
- ElastiCache für Redis OSS schließen sich Multi-AZ und Append-Only File (AOF) gegenseitig aus. Wenn Sie eine dieser Funktionen aktivieren, kann die andere nicht aktiviert werden.
- Der Ausfall eines Knotens kann durch den Ausfall einer gesamten Availability Zone verursacht werden. In diesem Fall wird das Replikat, das den ausgefallenen primären Knoten ersetzt, nur bei Verfügbarkeit der Availability Zone erstellt. Betrachten Sie beispielsweise eine Replikationsgruppe mit den primären Knoten in AZ-a und Replikaten in AZ-b und AZ-c. Wenn der primäre Knoten ausfällt, wird das Replikat mit der geringsten Replikationsverzögerung zum primären Cluster heraufgestuft. ElastiCache Erstellt dann nur dann ein neues Replikat in AZ-a (wo sich die ausgefallene Primärreplikation befand), wenn AZ-a gesichert und verfügbar ist.

- Ein vom Kunden initiiert Neustart des primären Clusters löst kein automatisches Failover aus. Andere Neustarts und Ausfälle hingegen lösen ein automatisches Failover aus.
- Wenn der primäre Cluster neu gestartet wird, werden seine Daten gelöscht, wenn er wieder online ist. Wenn die Read Replicas den gelöschten primären Cluster sehen, löschen sie ihre Kopie der Daten, was zu Datenverlust führt.
- Nachdem die Read Replica heraufgestuft wurde, werden die anderen Replikate mit dem neuen primären Cluster synchronisiert. Nach der ersten Synchronisierung wird der Inhalt der Replikate gelöscht, und sie synchronisieren die Daten aus dem neuen primären Cluster. Dieser Synchronisierungsvorgang führt zu einer kurzen Unterbrechung, bei der auf die Replikate nicht zugegriffen werden kann. Dieser Synchronisierungsvorgang bewirkt eine temporäre Zunahme der Arbeitslast auf dem primären Cluster, während er mit den Replikaten synchronisiert wird. Dieses Verhalten ist typisch für Valkey und Redis OSS und gilt nicht nur für Multi-AZ. ElastiCache Einzelheiten zu diesem Verhalten finden Sie unter [Replikation](#) auf der Valkey-Website.

Important

Für Valkey 7.2.6 und höher oder Redis OSS Version 2.8.22 und höher können Sie keine externen Replikate erstellen.

Für Redis OSS-Versionen vor 2.8.22 empfehlen wir, kein externes Replikat mit einem Cluster zu verbinden, der Multi-AZ-fähig ist. ElastiCache Diese nicht unterstützte Konfiguration kann zu Problemen führen, die eine ordnungsgemäße Durchführung von Failover und Wiederherstellung ElastiCache verhindern. Um ein externes Replikat mit einem ElastiCache Cluster zu verbinden, stellen Sie sicher, dass Multi-AZ nicht aktiviert ist, bevor Sie die Verbindung herstellen.

So werden Synchronisation und Backup implementiert

Alle unterstützten Versionen von Valkey und Redis OSS unterstützen Backup und Synchronisation zwischen dem Primär- und dem Replikatknoten. Die Art und Weise, wie Sicherung und Synchronisation implementiert werden, variiert jedoch je nach Version.

Redis OSS Version 2.8.22 und höher

Bei der Redis OSS-Replikation in den Versionen 2.8.22 und höher können Sie zwischen zwei Methoden wählen. Weitere Informationen erhalten Sie unter [Redis OSS-Versionen vor 2.8.22](#) und [Snapshot und Wiederherstellung](#).

Wenn die Schreiblasten während des unvergabelten Vorgangs zu intensiv werden, werden Schreibvorgänge in den Cluster verzögert, damit sich nicht zu viele Änderungen anstauen und somit einen erfolgreichen Snapshot verhindern würden.

Redis OSS-Versionen vor 2.8.22

Die Sicherung und Synchronisation von Redis OSS in Versionen vor 2.8.22 erfolgt in drei Schritten.

1. Vergabeln und serialisieren Sie die Clusterdaten als Hintergrundprozess auf dem Datenträger. Dadurch wird ein Snapshot erstellt. point-in-time
2. Akkumulieren Sie im Vordergrund ein Änderungsprotokoll im Client-Ausgabepuffer.

Important

Wenn das Änderungsprotokoll die Größe des Client-Ausgabepuffers überschreitet, schlägt die Sicherung oder Synchronisierung fehl. Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#).

3. Übertragen Sie schließlich die Cachedaten und anschließend das Änderungsprotokoll auf den Replikatknoten.

Erstellen einer Valkey- oder Redis OSS-Replikationsgruppe

Es bieten sich Ihnen die folgenden Optionen für das Erstellen eines Clusters mit Replikatknoten. Eine Option gilt, wenn Sie bereits über einen verfügbaren Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) verfügen, der keinem Cluster zugeordnet ist, der über Replikate verfügt, die als primärer Knoten verwendet werden können. Die andere trifft zu, wenn Sie einen Primärknoten mit dem Cluster und Lesereplikaten erstellen müssen. Derzeit muss ein Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) von Grund auf neu erstellt werden.

Option 1: [Erstellen einer Replikationsgruppe unter Verwendung eines vorhandenen Clusters](#)

Verwenden Sie diese Option, um einen vorhandenen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) mit einem Knoten zu nutzen. Sie geben diesen vorhandenen Knoten als Primärknoten im neuen Cluster an und fügen dem Cluster dann einzeln 1 bis 5 Lesereplikate hinzu. Wenn der vorhandene Cluster aktiv ist, werden die Read Replicas bei ihrer Erstellung mit ihm synchronisiert. Siehe [Erstellen einer Replikationsgruppe unter Verwendung eines vorhandenen Clusters](#).

Important

Sie können keinen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mithilfe eines vorhandenen Clusters erstellen. Informationen zum Erstellen eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) mithilfe der ElastiCache Konsole finden Sie unter [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Option 2: [Eine Valkey- oder Redis OSS-Replikationsgruppe von Grund auf neu erstellen](#)

Verwenden Sie diese Option, wenn Sie noch keinen verfügbaren Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) haben, den Sie als primären Knoten des Clusters verwenden können, oder wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) erstellen möchten. Siehe [Eine Valkey- oder Redis OSS-Replikationsgruppe von Grund auf neu erstellen](#).

Erstellen einer Replikationsgruppe unter Verwendung eines vorhandenen Clusters

Das folgende Verfahren fügt Ihrem Valkey- oder Redis OSS-Single-Node-Cluster (Clustermodus deaktiviert) eine Replikationsgruppe hinzu, die erforderlich ist, um Ihren Cluster auf die neueste Version von Valkey zu aktualisieren. Dabei handelt es sich um ein direktes Verfahren, das keine Ausfallzeiten und keinen Datenverlust beinhaltet. Wenn Sie eine Replikationsgruppe für Ihren Cluster mit einem Knoten erstellen, wird der Knoten des Clusters zum primären Knoten im neuen Cluster. Wenn Sie keinen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) haben, den Sie als primären Cluster verwenden können, finden Sie weitere Informationen unter. [Eine Valkey- oder Redis OSS-Replikationsgruppe von Grund auf neu erstellen](#)

Ein verfügbarer Cluster ist ein vorhandener Valkey- oder Redis OSS-Cluster mit einem Knoten. Derzeit unterstützt Valkey oder Redis OSS (Clustermodus aktiviert) die Erstellung eines Clusters mit Replikaten unter Verwendung eines verfügbaren Clusters mit einem einzelnen Knoten nicht. Wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) erstellen möchten, finden Sie weitere Informationen unter. [Erstellen eines Valkey- oder Redis OSS-Clusters \(Clustermodus aktiviert\) \(Konsole\)](#)

Erstellen einer Replikationsgruppe mithilfe eines vorhandenen Clusters (Konsole)

Weitere Informationen finden Sie im Thema [Mit dem ElastiCache AWS-Managementkonsole](#).

Erstellen einer Replikationsgruppe mithilfe eines verfügbaren Valkey- oder Redis OSS-Clusters (AWS CLI)

Es gibt zwei Schritte zum Erstellen einer Replikationsgruppe mit Read Replicas, wenn Sie einen verfügbaren Valkey- oder Redis OSS-Cache-Cluster als Primär-Cache-Cluster verwenden, wenn Sie den verwenden. AWS CLI

Wenn AWS CLI Sie den verwenden, erstellen Sie eine Replikationsgruppe, indem Sie den verfügbaren Standalone-Knoten als primären Knoten des Clusters `--primary-cluster-id` und die Anzahl der Knoten, die Sie im Cluster haben möchten, mit dem CLI-Befehl `angeben create-replication-group`. Schließen Sie die folgenden Parameter ein.

`--replication-group-id`

Der Name der zu erstellenden Replikationsgruppe. Der Wert dieses Parameters wird als Basis für die Namen der hinzugefügten Knoten mit einer fortlaufenden 3-stelligen Nummer am Ende der `--replication-group-id` verwendet. Beispiel, `sample-repl-group-001`.

Die Benennungsbeschränkungen für Valkey oder Redis OSS (Clustermodus deaktiviert) für Replikationsgruppen lauten wie folgt:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

`--replication-group-description`

Beschreibung der Replikationsgruppe.

`--num-node-groups`

Die Anzahl der Knoten, die Sie in diesem Cluster benötigen. Dieser Wert umfasst den primären Knoten. Dieser Parameter hat einen Maximalwert von sechs.

`--primary-cluster-id`

Der Name des Knotens des verfügbaren Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert), den Sie als primärer Knoten in dieser Replikationsgruppe verwenden möchten.

Mit dem folgenden Befehl wird die Replikationsgruppe erstellt, `sample-repl-group` wobei der verfügbare Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) `redis01` als primärer Knoten der Replikationsgruppe verwendet wird. Er erstellt 2 neue Knoten, bei denen es sich um Read Replicas handelt. Die Einstellungen von `redis01` (d. h. Parametergruppe, Sicherheitsgruppe, Knotentyp, Engine-Version usw.) werden auf alle Knoten in der Replikationsgruppe angewendet.

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --primary-cluster-id redis01
```

Für Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "demo cluster with replicas" ^
```

```
--num-cache-clusters 3 ^  
--primary-cluster-id redis01
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im AWS CLI Thema. [create-replication-group](#)

Als Nächstes Hinzufügen von Read Replicas zur Replikationsgruppe

Nachdem die Replikationsgruppe erstellt wurde, fügen Sie ihr mit dem Befehl `create-cache-cluster` eine bis fünf Read Replicas hinzu und schließen Sie unbedingt die folgenden Parameter ein.

`--cache-cluster-id`

Der Name des Clusters, den Sie zur Replikationsgruppe hinzufügen.

Für die Benennung von Clustern gelten die folgenden Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

`--replication-group-id`

Der Name der Replikationsgruppe, zu der Sie diesen Cluster hinzufügen.

Wiederholen Sie diesen Befehl für jede Read Replica, die Sie zur Replikationsgruppe hinzufügen möchten, und ändern Sie dabei nur den Wert des Parameters `--cache-cluster-id`.

Note

Denken Sie daran, dass eine Replikationsgruppe über nicht mehr als fünf Read Replicas verfügen kann. Bei dem Versuch, einer Replikationsgruppe mit bereits fünf Read Replicas eine weitere Read Replica hinzuzufügen, schlägt die Operation fehl.

Der folgende Code fügt die Read Replica `my-replica01` zur Replikationsgruppe `sample-repl-group` hinzu. Die Einstellungen des primären Clusters – Parametergruppe, Sicherheitsgruppe,

Knotentyp usw. – werden auf Knoten angewendet, sobald sie der Replikationsgruppe hinzugefügt werden.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id my-replica01 \  
  --replication-group-id sample-repl-group
```

Für Windows:

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id my-replica01 ^  
  --replication-group-id sample-repl-group
```

Die Ausgabe dieses Befehls sieht in etwa folgendermaßen aus.

```
{  
  "ReplicationGroup": {  
    "Status": "creating",  
    "Description": "demo cluster with replicas",  
    "ClusterEnabled": false,  
    "ReplicationGroupId": "sample-repl-group",  
    "SnapshotRetentionLimit": 1,  
    "AutomaticFailover": "disabled",  
    "SnapshotWindow": "00:00-01:00",  
    "SnapshottingClusterId": "redis01",  
    "MemberClusters": [  
      "sample-repl-group-001",  
      "sample-repl-group-002",  
      "redis01"  
    ],  
    "CacheNodeType": "cache.m4.large",  
    "DataTiering": "disabled",  
    "PendingModifiedValues": {}  
  }  
}
```

Weitere Informationen finden Sie in den folgenden AWS CLI Themen:

- [create-replication-group](#)

- [modify-replication-group](#)

Hinzufügen von Replikaten zu einem eigenständigen Valkey- oder Redis OSS-Cluster (Cluster Mode Disabled) (API) ElastiCache

Wenn Sie die ElastiCache API verwenden, erstellen Sie eine Replikationsgruppe, die den verfügbaren Standalone-Knoten als primären Knoten des Clusters `PrimaryClusterId` und die Anzahl der Knoten, die Sie im Cluster haben möchten, mit dem CLI-Befehl, angeben `CreateReplicationGroup`. Schließen Sie die folgenden Parameter ein.

`ReplicationGroupId`

Der Name der zu erstellenden Replikationsgruppe. Der Wert dieses Parameters wird als Basis für die Namen der hinzugefügten Knoten mit einer fortlaufenden 3-stelligen Nummer am Ende der `ReplicationGroupId` verwendet. Beispiel, `sample-repl-group-001`.

Die Benennungsbeschränkungen für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert) lauten wie folgt:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

`ReplicationGroupDescription`

Beschreibung des Clusters mit Replikaten.

`NumCacheClusters`

Die Anzahl der Knoten, die Sie in diesem Cluster benötigen. Dieser Wert umfasst den primären Knoten. Dieser Parameter hat einen Maximalwert von sechs.

`PrimaryClusterId`

Der Name des verfügbaren Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert), den Sie als primärer Knoten in diesem Cluster verwenden möchten.

Der folgende Befehl erstellt den Cluster mit Replikaten, `sample-repl-group` wobei der verfügbare Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) `redis01` als primärer Knoten der Replikationsgruppe verwendet wird. Er erstellt 2 neue Knoten, bei denen es sich um Read Replicas

handelt. Die Einstellungen von `redis01` (d. h. Parametergruppe, Sicherheitsgruppe, Knotentyp, Engine-Version usw.) werden auf alle Knoten in der Replikationsgruppe angewendet.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateReplicationGroup  
&Engine=redis  
&EngineVersion=6.0  
&ReplicationGroupDescription=Demo%20cluster%20with%20replicas  
&ReplicationGroupId=sample-repl-group  
&PrimaryClusterId=redis01  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie in den ElastiCache APL-Themen:

- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

Als Nächstes Hinzufügen von Read Replicas zur Replikationsgruppe

Nachdem die Replikationsgruppe erstellt wurde, fügen Sie ihr mit der Operation `CreateCacheCluster` eine bis fünf Read Replicas hinzu und schließen Sie unbedingt die folgenden Parameter ein.

CacheClusterId

Der Name des Clusters, den Sie zur Replikationsgruppe hinzufügen.

Für die Benennung von Clustern gelten die folgenden Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

ReplicationGroupId

Der Name der Replikationsgruppe, zu der Sie diesen Cluster hinzufügen.

Wiederholen Sie diese Operation für jede Read Replica, die Sie zur Replikationsgruppe hinzufügen möchten, und ändern Sie dabei nur den Wert des Parameters `CacheClusterId`.

Der folgende Code fügt die Lesereplikate `myReplica01` der Replikationsgruppe `myReplGroup` hinzu. Die Einstellungen des primären Clusters – Parametergruppe, Sicherheitsgruppe, Knotentyp usw. – werden auf Knoten angewendet, wenn sie der Replikationsgruppe hinzugefügt werden.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheCluster  
&CacheClusterId=myReplica01  
&ReplicationGroupId=myReplGroup  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2015-02-02  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Credential=[your-access-key-id]/20150202/us-west-2/elasticache/aws4_request  
&X-Amz-Date=20150202T170651Z  
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date  
&X-Amz-Signature=[signature-value]
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im ElastiCache API-Thema [CreateCacheCluster](#).

Eine Valkey- oder Redis OSS-Replikationsgruppe von Grund auf neu erstellen

Im Folgenden erfahren Sie, wie Sie eine Valkey- oder Redis OSS-Replikationsgruppe erstellen, ohne einen vorhandenen Valkey- oder Redis OSS-Cluster als primären Cluster zu verwenden. Sie können mithilfe der Konsole, der oder der API eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) oder Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) von Grund auf neu erstellen. [ElastiCache AWS CLI ElastiCache](#)

Bevor Sie fortfahren, entscheiden Sie, ob Sie eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) oder eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) erstellen möchten. Hilfreiche Informationen zur Entscheidung finden Sie unter [Replikation: Valkey- und Redis OSS-Clustermodus deaktiviert oder aktiviert](#).

Themen

- [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Cluster Mode Disabled\) von Grund auf neu erstellen](#)
- [Eine Replikationsgruppe in Valkey oder Redis OSS \(Cluster Mode Enabled\) von Grund auf neu erstellen](#)

Eine Valkey- oder Redis OSS-Replikationsgruppe (Cluster Mode Disabled) von Grund auf neu erstellen

Sie können mithilfe der ElastiCache Konsole, der oder der API eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu erstellen. AWS CLI ElastiCache Eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) besteht immer aus einer Knotengruppe, einem primären Cluster und bis zu fünf Read Replicas. Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert) unterstützen die Partitionierung Ihrer Daten nicht.

 Note

Das node/shard Limit kann auf maximal 500 pro Cluster erhöht werden. Informationen zum Anfordern einer Limiterhöhung finden Sie unter [AWS Service-Limits](#), wenn Sie den Instance-Typ in die Anfrage eingeben.

Um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu zu erstellen, wählen Sie einen der folgenden Ansätze:

Eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu erstellen (AWS CLI)

Das folgende Verfahren erstellt eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) mithilfe von AWS CLI

Wenn Sie eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und alle ihre Knoten mit einem einzigen Befehlsaufruf. AWS CLI `create-replication-group` Schließen Sie die folgenden Parameter ein.

`--replication-group-id`

Der Name der zu erstellenden Replikationsgruppe.

Die Benennungsbeschränkungen für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert) lauten wie folgt:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

`--replication-group-description`

Beschreibung der Replikationsgruppe.

`--num-cache-clusters`

Die Gesamtzahl von Knoten, die Sie mit dieser Replikationsgruppe erstellen möchten, primäre Knoten und Read Replicas kombiniert.

Wenn Sie Multi-AZ (`--automatic-failover-enabled`) aktivieren, muss der Wert für `--num-cache-clusters` mindestens 2 sein.

`--cache-node-type`

Der Knotentyp für jeden Knoten in der Replikationsgruppe.

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2 Amazon-Instanztypen](#).

`--data-tiering-enabled`

Legen Sie diesen Parameter fest, wenn Sie einen R6gd-Knotentyp verwenden. Wenn Sie kein Daten-Tiering verwenden möchten, legen Sie `--no-data-tiering-enabled` fest. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

`--cache-parameter-group`

Geben Sie eine Parametergruppe an, die Ihrer Engine-Version entspricht. Wenn Sie Redis OSS 3.2.4 oder höher ausführen, geben Sie die Parametergruppe oder eine davon abgeleitete `default.redis3.2` Parametergruppe an, um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) `default.redis3.2` zu erstellen. Weitere Informationen finden Sie unter [Valkey- und Redis OSS-Parameter](#).

`--Netzwerktyp`

Entweder `ipv4`, `ipv6` oder `dual-stack`. Wenn Sie Dual-Stack auswählen, müssen Sie den `--IpDiscovery`-Parameter entweder auf `ipv4` oder `ipv6` festlegen.

`--engine`

`redis`

--engine-version

Wählen Sie für den umfangreichsten Satz von Funktionen die neueste Engine-Version aus.

Die Namen der Knoten werden vom Namen der Replikationsgruppe abgeleitet, indem `-00#` an den Namen der Replikationsgruppe angehängt wird. Wenn Sie z. B. den Replikationsgruppennamen `myReplGroup` verwenden, lauten die Namen des primären Knotens `myReplGroup-001` und der Read Replicas `myReplGroup-002` bis `myReplGroup-006`.

Wenn Sie die Verschlüsselung während der Übertragung oder im Ruhezustand für diese Replikationsgruppe aktivieren möchten, fügen Sie einen oder beide der Parameter `--transit-encryption-enabled` oder `--at-rest-encryption-enabled` hinzu und erfüllen die folgenden Bedingungen.

- In Ihrer Replikationsgruppe muss Redis OSS Version 3.2.6 oder 4.0.10 ausgeführt werden.
- Die Replikationsgruppe muss in einer Amazon VPC erstellt werden.
- Sie müssen außerdem den Parameter `--cache-subnet-group` einbinden.
- Sie müssen auch den Parameter `--auth-token` einbinden, mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH-Token (Passwort), das für Operationen auf dieser Replikationsgruppe benötigt wird.

Der folgende Vorgang erstellt eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) `sample-repl-group` mit drei Knoten, einem primären und zwei Replikaten.

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "Demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --cache-node-type cache.m4.large \  
  --engine redis
```

Für Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "Demo cluster with replicas" ^  
  --num-cache-clusters 3 ^
```

```
--cache-node-type cache.m4.large ^  
--engine redis
```

Die Ausgabe dieses Befehls sieht in etwa wie folgt aus:

```
{  
  "ReplicationGroup": {  
    "Status": "creating",  
    "Description": "Demo cluster with replicas",  
    "ClusterEnabled": false,  
    "ReplicationGroupId": "sample-repl-group",  
    "SnapshotRetentionLimit": 0,  
    "AutomaticFailover": "disabled",  
    "SnapshotWindow": "01:30-02:30",  
    "MemberClusters": [  
      "sample-repl-group-001",  
      "sample-repl-group-002",  
      "sample-repl-group-003"  
    ],  
    "CacheNodeType": "cache.m4.large",  
    "DataTiering": "disabled",  
    "PendingModifiedValues": {}  
  }  
}
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im Thema. AWS CLI [create-replication-group](#)

Eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu erstellen (ElastiCache API)

Mit dem folgenden Verfahren wird mithilfe der API eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) erstellt. ElastiCache

Wenn Sie eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und alle ihre Knoten mit einem einzigen Aufruf des API-Vorgangs. ElastiCache CreateReplicationGroup Schließen Sie die folgenden Parameter ein.

ReplicationGroupId

Der Name der zu erstellenden Replikationsgruppe.

Die Benennungsbeschränkungen für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus aktiviert) lauten wie folgt:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

ReplicationGroupDescription

Ihre Beschreibung der Replikationsgruppe.

NumCacheClusters

Die Gesamtzahl der Knoten, die Sie mit dieser Replikationsgruppe erstellen möchten, primäre Knoten und Read Replicas kombiniert.

Wenn Sie Multi-AZ (`AutomaticFailoverEnabled=true`) aktivieren, muss der Wert für `NumCacheClusters` mindestens 2 sein.

CacheNodeType

Der Knotentyp für jeden Knoten in der Replikationsgruppe.

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2 Amazon-Instanztypen](#).

--data-tiering-enabled

Legen Sie diesen Parameter fest, wenn Sie einen R6gd-Knotentyp verwenden. Wenn Sie kein Daten-Tiering verwenden möchten, legen Sie `--no-data-tiering-enabled` fest. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

CacheParameterGroup

Geben Sie eine Parametergruppe an, die Ihrer Engine-Version entspricht. Wenn Sie Redis OSS 3.2.4 oder höher ausführen, geben Sie die Parametergruppe oder eine davon abgeleitete `default.redis3.2` Parametergruppe an, um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) `default.redis3.2` zu erstellen. Weitere Informationen finden Sie unter [Valkey- und Redis OSS-Parameter](#).

--Netzwerktyp

Entweder `ipv4`, `ipv` oder `dual-stack`. Wenn Sie Dual-Stack auswählen, müssen Sie den `--IpDiscovery`-Parameter entweder auf `ipv4` oder `ipv6` festlegen.

Engine

`redis`

EngineVersion

`6.0`

Die Namen der Knoten werden vom Namen der Replikationsgruppe abgeleitet, indem `-00#` an den Namen der Replikationsgruppe angehängt wird. Wenn Sie z. B. den Replikationsgruppennamen `myReplGroup` verwenden, lauten die Namen des primären Knotens `myReplGroup-001` und der Read Replicas `myReplGroup-002` bis `myReplGroup-006`.

Wenn Sie die Verschlüsselung während der Übertragung oder im Ruhezustand für diese Replikationsgruppe aktivieren möchten, fügen Sie einen oder beide der Parameter `TransitEncryptionEnabled=true` oder `AtRestEncryptionEnabled=true` hinzu und erfüllen die folgenden Bedingungen.

- In Ihrer Replikationsgruppe muss Redis OSS Version 3.2.6 oder 4.0.10 ausgeführt werden.
- Die Replikationsgruppe muss in einer Amazon VPC erstellt werden.
- Sie müssen außerdem den Parameter `CacheSubnetGroup` einbinden.
- Sie müssen auch den Parameter `AuthToken` einbinden, mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH-Token (Passwort), das für Operationen auf dieser Replikationsgruppe benötigt wird.

Der folgende Vorgang erstellt die Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) `myReplGroup` mit drei Knoten, einem primären und zwei Replikaten.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateReplicationGroup  
&CacheNodeType=cache.m4.large  
&CacheParameterGroup=default.redis6.x  
&Engine=redis  
&EngineVersion=6.0  
&NumCacheClusters=3
```

```
&ReplicationGroupDescription=test%20group
&ReplicationGroupId=myReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im ElastiCache API-Thema. [CreateReplicationGroup](#)

Eine Replikationsgruppe in Valkey oder Redis OSS (Cluster Mode Enabled) von Grund auf neu erstellen

Sie können einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) mithilfe der ElastiCache Konsole, der oder der AWS CLI API erstellen. ElastiCache Eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) hat 1 bis 500 Shards (API/CLI: Knotengruppen), einen primären Knoten in jedem Shard und bis zu 5 Read Replicas in jedem Shard. Sie können einen Cluster mit einer höheren Anzahl an Shards und einer geringeren Anzahl an Replikaten mit bis zu 90 Knoten pro Cluster erstellen. Diese Clusterkonfiguration reicht von 90 Shards und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die Anzahl erlaubter Replikate entspricht.

Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Valkey- oder Redis-OSS-Engine-Version 5.0.6 oder höher ist. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

Erstellen eines Clusters in Valkey oder Redis OSS (Clustermodus aktiviert)

- [Erstellen eines Valkey- oder Redis OSS-Clusters \(Clustermodus aktiviert\) \(Konsole\)](#)
- [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Cluster Mode Enabled\) von Grund auf neu erstellen \(\)AWS CLI](#)
- [Eine Replikationsgruppe in Valkey oder Redis OSS \(Cluster Mode Enabled\) von Grund auf neu erstellen \(ElastiCache API\)](#)

Erstellen eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) (Konsole)

Informationen zum Erstellen eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) finden Sie unter. [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Stellen Sie sicher, dass der Cluster-Modus aktiviert ist, Cluster-Modus aktiviert (aufskalieren) und dass mindestens zwei Shards und ein Replikationsknoten in jedem angegeben sind.

Eine Valkey- oder Redis OSS-Replikationsgruppe (Cluster Mode Enabled) von Grund auf neu erstellen ()AWS CLI

Das folgende Verfahren erstellt eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) mithilfe von AWS CLI

Wenn Sie eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und alle ihre Knoten mit einem einzigen Befehlsaufruf. AWS CLI `create-replication-group` Schließen Sie die folgenden Parameter ein.

`--replication-group-id`

Der Name der zu erstellenden Replikationsgruppe.

Die Benennungsbeschränkungen für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus aktiviert) lauten wie folgt:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

`--replication-group-description`

Beschreibung der Replikationsgruppe.

`--cache-node-type`

Der Knotentyp für jeden Knoten in der Replikationsgruppe.

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2 Amazon-Instanztypen](#).

--data-tiering-enabled

Legen Sie diesen Parameter fest, wenn Sie einen R6gd-Knotentyp verwenden. Wenn Sie kein Daten-Tiering verwenden möchten, legen Sie `--no-data-tiering-enabled` fest. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

--cache-parameter-group

Geben Sie die `default.redis6.x.cluster.on` Parametergruppe oder eine davon abgeleitete Parametergruppe an, um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) `default.redis6.x.cluster.on` zu erstellen. Weitere Informationen finden Sie unter [Änderungen der Parameter in Redis OSS 6.x](#).

--engine

redis

--engine-version

3.2.4

--num-node-groups

Die Anzahl der Knotengruppen in dieser Replikationsgruppe. Gültige Werte sind 1 bis 500.

 Note

Das node/shard Limit kann auf maximal 500 pro Cluster erhöht werden. Um eine Limiterhöhung anzufordern, lesen Sie [AWS Service Limits](#) und wählen Sie den Limittyp "Knoten per Cluster per Instance-Typ".

--replicas-per-node-group

Die Anzahl der Replikat-Knoten in jeder Knotengruppe. Gültige Werte sind 0 bis 5.

--Netzwerktyp

Entweder `ipv4`, `ipv` oder `dual-stack`. Wenn Sie Dual-Stack auswählen, müssen Sie den `--IpDiscovery`-Parameter entweder auf `ipv4` oder `ipv6` festlegen.

Wenn Sie die Verschlüsselung während der Übertragung oder im Ruhezustand für diese Replikationsgruppe aktivieren möchten, fügen Sie einen oder beide der Parameter `--transit-`

encryption-enabled oder --at-rest-encryption-enabled hinzu und erfüllen die folgenden Bedingungen.

- In Ihrer Replikationsgruppe muss Redis OSS Version 3.2.6 oder 4.0.10 ausgeführt werden.
- Die Replikationsgruppe muss in einer Amazon VPC erstellt werden.
- Sie müssen außerdem den Parameter --cache-subnet-group einbinden.
- Sie müssen auch den Parameter --auth-token einbinden, mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH-Token (Passwort), das für Operationen auf dieser Replikationsgruppe benötigt wird.

Der folgende Vorgang erstellt die Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) `sample-repl-group` mit drei Knoten groups/shards (--num-node-groups) mit jeweils drei Knoten, einem primären und zwei Read Replicas (--replicas-per-node-group).

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \
  --replication-group-id sample-repl-group \
  --replication-group-description "Demo cluster with replicas" \
  --num-node-groups 3 \
  --replicas-per-node-group 2 \
  --cache-node-type cache.m4.large \
  --engine redis \
  --security-group-ids SECURITY_GROUP_ID \
  --cache-subnet-group-name SUBNET_GROUP_NAME>
```

Für Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id sample-repl-group ^
  --replication-group-description "Demo cluster with replicas" ^
  --num-node-groups 3 ^
  --replicas-per-node-group 2 ^
  --cache-node-type cache.m4.large ^
  --engine redis ^
  --security-group-ids SECURITY_GROUP_ID ^
  --cache-subnet-group-name SUBNET_GROUP_NAME>
```

Der voranstehende Befehl erzeugt die folgende Ausgabe.

```
{
  "ReplicationGroup": {
    "Status": "creating",
    "Description": "Demo cluster with replicas",
    "ReplicationGroupId": "sample-repl-group",
    "SnapshotRetentionLimit": 0,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "05:30-06:30",
    "MemberClusters": [
      "sample-repl-group-0001-001",
      "sample-repl-group-0001-002",
      "sample-repl-group-0001-003",
      "sample-repl-group-0002-001",
      "sample-repl-group-0002-002",
      "sample-repl-group-0002-003",
      "sample-repl-group-0003-001",
      "sample-repl-group-0003-002",
      "sample-repl-group-0003-003"
    ],
    "PendingModifiedValues": {}
  }
}
```

Wenn Sie eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) von Grund auf neu erstellen, können Sie jeden Shard im Cluster mithilfe des `--node-group-configuration` Parameters konfigurieren, wie im folgenden Beispiel gezeigt, der zwei Knotengruppen konfiguriert (Konsole: Shards). Der erste Shard verfügt über zwei Knoten, einen primären Knoten und eine Read Replica. Der zweite Shard verfügt über drei Knoten, einen primären Knoten und zwei Read Replicas.

`--node-group-configuration`

Die Konfiguration jeder Knotengruppe. Der Parameter `--node-group-configuration` besteht aus den folgenden Feldern.

- **PrimaryAvailabilityZone** – Die Availability Zone, in der sich der Primärknoten dieser Knotengruppe befindet. Wenn dieser Parameter weggelassen wird, wird die Availability Zone ElastiCache für den primären Knoten ausgewählt.

Beispiel: `us-west-2a`.

- **ReplicaAvailabilityZones** – Eine durch Kommas getrennte Liste von Availability Zones, in denen sich die Lesereplikate befinden. Die Anzahl der Availability Zones in dieser Liste muss mit dem Wert von **ReplicaCount** übereinstimmen. Wenn dieser Parameter weggelassen wird, werden die ElastiCache Availability Zones für die Replikatknoten ausgewählt.

Beispiel: „us-west-2a,us-west-2b,us-west-2c“

- **ReplicaCount** – Die Anzahl der Replikatknoten in dieser Knotengruppe.
- **Slots** – Eine Zeichenfolge, die den Keyspace für die Knotengruppe angibt. Das Format für die Zeichenfolge ist **startKey-endKey**. Wenn dieser Parameter weggelassen wird, werden ElastiCache die Schlüssel gleichmäßig auf die Knotengruppen verteilt.

Beispiel: "0-4999"

Der folgende Vorgang erstellt die Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) **new-group** mit zwei Knotengruppen/Shards (). **--num-node-groups** Im Gegensatz zum voranstehenden Beispiel sind die einzelnen Knotengruppen unterschiedlich konfiguriert (**--node-group-configuration**).

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \
  --replication-group-id new-group \
  --replication-group-description "Sharded replication group" \
  --engine redis \
  --snapshot-retention-limit 8 \
  --cache-node-type cache.m4.medium \
  --num-node-groups 2 \
  --node-group-configuration \
    "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-east-1c',ReplicaAvailabilityZones='us-east-1b'" \
    "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a', 'us-east-1c'"
```

Für Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id new-group ^
  --replication-group-description "Sharded replication group" ^
```

```
--engine redis ^
--snapshot-retention-limit 8 ^
--cache-node-type cache.m4.medium ^
--num-node-groups 2 ^
--node-group-configuration \
    "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-east-1c',ReplicaAvailabilityZones='us-east-1b'" \
    "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a', 'us-east-1c'"
```

Die voranstehende Operation erzeugt die folgende Ausgabe.

```
{
  "ReplicationGroup": {
    "Status": "creating",
    "Description": "Sharded replication group",
    "ReplicationGroupId": "rc-rg",
    "SnapshotRetentionLimit": 8,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "10:00-11:00",
    "MemberClusters": [
      "rc-rg-0001-001",
      "rc-rg-0001-002",
      "rc-rg-0002-001",
      "rc-rg-0002-002",
      "rc-rg-0002-003"
    ],
    "PendingModifiedValues": {}
  }
}
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im Thema. AWS CLI [create-replication-group](#)

Eine Replikationsgruppe in Valkey oder Redis OSS (Cluster Mode Enabled) von Grund auf neu erstellen (ElastiCache API)

Das folgende Verfahren erstellt mithilfe der API eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert). ElastiCache

Wenn Sie eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und alle ihre Knoten mit einem einzigen Aufruf der

API-Operation. ElastiCache CreateReplicationGroup Schließen Sie die folgenden Parameter ein.

ReplicationGroupId

Der Name der zu erstellenden Replikationsgruppe.

Die Benennungsbeschränkungen für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus aktiviert) lauten wie folgt:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

ReplicationGroupDescription

Beschreibung der Replikationsgruppe.

NumNodeGroups

Die Anzahl der Knotengruppen, die Sie mit dieser Replikationsgruppe erstellen möchten. Gültige Werte sind 1 bis 500.

ReplicasPerNodeGroup

Die Anzahl der Replikat-Knoten in jeder Knotengruppe. Gültige Werte sind 1 bis 5.

NodeGroupConfiguration

Die Konfiguration jeder Knotengruppe. Der Parameter NodeGroupConfiguration besteht aus den folgenden Feldern.

- **PrimaryAvailabilityZone** – Die Availability Zone, in der sich der Primärknoten dieser Knotengruppe befindet. Wenn dieser Parameter weggelassen wird, wird die ElastiCache Availability Zone für den primären Knoten ausgewählt.

Beispiel: us-west-2a.

- **ReplicaAvailabilityZones** – Eine Liste der Availability Zones, in denen sich die Lesereplikate befinden. Die Anzahl der Availability Zones in dieser Liste muss mit dem Wert von ReplicaCount übereinstimmen. Wenn dieser Parameter weggelassen wird, werden die ElastiCache Availability Zones für die Replikatknoten ausgewählt.
- **ReplicaCount** – Die Anzahl der Replikatknoten in dieser Knotengruppe.

- **Slots** – Eine Zeichenfolge, die den Keyspace für die Knotengruppe angibt. Das Format für die Zeichenfolge ist `startKey-endKey`. Wenn dieser Parameter weggelassen wird, werden ElastiCache die Schlüssel gleichmäßig auf die Knotengruppen verteilt.

Beispiel: "0-4999"

CacheNodeType

Der Knotentyp für jeden Knoten in der Replikationsgruppe.

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2 Amazon-Instanztypen](#).

--data-tiering-enabled

Legen Sie diesen Parameter fest, wenn Sie einen R6gd-Knotentyp verwenden. Wenn Sie kein Daten-Tiering verwenden möchten, legen Sie `--no-data-tiering-enabled` fest. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

CacheParameterGroup

Geben Sie die `default.redis6.x.cluster.on` Parametergruppe oder eine davon abgeleitete Parametergruppe an, um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) `default.redis6.x.cluster.on` zu erstellen. Weitere Informationen finden Sie unter [Änderungen der Parameter in Redis OSS 6.x](#).

--Netzwerktyp

Entweder `ipv4`, `ipv6` oder `dual-stack`. Wenn Sie Dual-Stack auswählen, müssen Sie den `--IpDiscovery`-Parameter entweder auf `ipv4` oder `ipv6` festlegen.

Engine

redis

EngineVersion

6.0

Wenn Sie die Verschlüsselung während der Übertragung oder im Ruhezustand für diese Replikationsgruppe aktivieren möchten, fügen Sie einen oder beide der Parameter `TransitEncryptionEnabled=true` oder `AtRestEncryptionEnabled=true` hinzu und erfüllen die folgenden Bedingungen.

- Auf Ihrer Replikationsgruppe muss Redis OSS Version 3.2.6 oder 4.0.10 ausgeführt werden.
- Die Replikationsgruppe muss in einer Amazon VPC erstellt werden.
- Sie müssen außerdem den Parameter `CacheSubnetGroup` einbinden.
- Sie müssen auch den Parameter `AuthToken` einbinden, mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH-Token (Passwort), das für Operationen auf dieser Replikationsgruppe benötigt wird.

Die Zeilenumbrüche dienen der besseren Lesbarkeit.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=CreateReplicationGroup  
  &CacheNodeType=cache.m4.large  
  &CacheParameterGroup=default.redis6.xcluster.on  
  &Engine=redis  
  &EngineVersion=6.0  
  &NumNodeGroups=3  
  &ReplicasPerNodeGroup=2  
  &ReplicationGroupDescription=test%20group  
  &ReplicationGroupId=myReplGroup  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im API-Thema. ElastiCache [CreateReplicationGroup](#)

Anzeigen der Details einer Replikationsgruppe

Unter bestimmten Umständen möchten Sie vielleicht Details zu einer Replikationsgruppe einsehen. Sie können die ElastiCache Konsole, das AWS CLI for ElastiCache oder die ElastiCache API verwenden. Der Konsolenprozess unterscheidet sich für Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert).

Anzeige der Details einer Replikationsgruppe

- [Valkey- oder Redis-OSS \(Clustermodus deaktiviert\) mit Replikaten anzeigen](#)
 - [Anzeigen einer Valkey- oder Redis OSS-Replikationsgruppe \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
 - [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Clustermodus deaktiviert\) anzeigen \(\)AWS CLI](#)
 - [Eine Valkey- oder Redis OSS-Replikationsgruppe \(API\) anzeigen \(Cluster-Modus deaktiviert\) ElastiCache](#)
- [Eine Replikationsgruppe anzeigen: Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
 - [Anzeigen eines Valkey- oder Redis OSS-Clusters \(Clustermodus aktiviert\) \(Konsole\)](#)
 - [Einen Valkey- oder Redis OSS-Cluster \(Clustermodus aktiviert\) anzeigen \(\)AWS CLI](#)
 - [Einen Valkey- oder Redis OSS-Cluster \(Clustermodus aktiviert\) \(API\) anzeigen ElastiCache](#)
- [Anzeigen der Details einer Replikationsgruppe \(AWS CLI\)](#)
- [Details einer Replikationsgruppe anzeigen \(ElastiCache API\)](#)

Valkey- oder Redis-OSS (Clustermodus deaktiviert) mit Replikaten anzeigen

Sie können die Details eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) mit Replikaten (API/CLI: Replikationsgruppe) mithilfe der ElastiCache Konsole, des For oder der AWS CLI API anzeigen. ElastiCache ElastiCache

Details eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) anzeigen

- [Anzeigen einer Valkey- oder Redis OSS-Replikationsgruppe \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Clustermodus deaktiviert\) anzeigen \(\)AWS CLI](#)
- [Eine Valkey- oder Redis OSS-Replikationsgruppe \(API\) anzeigen \(Cluster-Modus deaktiviert\) ElastiCache](#)

Anzeigen einer Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus deaktiviert) (Konsole)

Informationen zum Anzeigen der Details eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) mit Replikaten mithilfe der ElastiCache Konsole finden Sie im Thema. [Details zu Valkey oder Redis OSS \(Clustermodus deaktiviert\) anzeigen \(Konsole\)](#)

Eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) anzeigen ()AWS CLI

Ein AWS CLI Beispiel, in dem die Details einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) angezeigt werden, finden Sie unter. [Anzeigen der Details einer Replikationsgruppe \(AWS CLI\)](#)

Eine Valkey- oder Redis OSS-Replikationsgruppe (API) anzeigen (Cluster-Modus deaktiviert)
ElastiCache

Ein ElastiCache API-Beispiel, in dem die Details einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) angezeigt werden, finden Sie unter. [Details einer Replikationsgruppe anzeigen \(ElastiCache API\)](#)

Eine Replikationsgruppe anzeigen: Valkey oder Redis OSS (Clustermodus aktiviert)

Anzeigen eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) (Konsole)

Informationen zum Anzeigen der Details eines Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert) mithilfe der ElastiCache Konsole finden Sie unter. [Details für einen Valkey- oder Redis OSS-Cluster \(Cluster Mode Enabled\) anzeigen \(Konsole\)](#)

Einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) anzeigen ()AWS CLI

Ein ElastiCache CLI-Beispiel, das die Details einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) anzeigt, finden Sie unter. [Anzeigen der Details einer Replikationsgruppe \(AWS CLI\)](#)

Einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) (API) anzeigen ElastiCache

Ein ElastiCache API-Beispiel, das die Details einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) anzeigt, finden Sie unter. [Details einer Replikationsgruppe anzeigen \(ElastiCache API\)](#)

Anzeigen der Details einer Replikationsgruppe (AWS CLI)

Sie können die Details für eine Replikationsgruppe mit dem AWS CLI `describe-replication-groups` Befehl anzeigen. Verfeinern Sie die Auflistung mit den folgenden optionalen Parametern. Wenn die Parameter weggelassen werden, werden Details für bis zu 100 Replikationsgruppen zurückgegeben.

Optionale Parameter

- `--replication-group-id` – Verwenden Sie diesen Parameter zur Auflistung der Details einer bestimmten Replikationsgruppe. Wenn die angegebene Replikationsgruppe mehr als eine Knotengruppe besitzt, werden die zurückgegebenen Ergebnisse nach Knotengruppe gruppiert.
- `--max-items` – Verwenden Sie diesen Parameter, um die Anzahl der aufgelisteten Replikationsgruppen einzuschränken. Der Wert für `--max-items` darf nicht kleiner als 20 oder größer als 100 sein.

Example

Der folgende Code listet die Details für bis zu 100 Replikationsgruppen auf.

```
aws elasticache describe-replication-groups
```

Der folgende Code listet die Details für `sample-repl-group` auf.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

Der folgende Code listet die Details für `sample-repl-group` auf.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

Der folgende Code listet die Details für bis zu 25 Replikationsgruppen auf.

```
aws elasticache describe-replication-groups --max-items 25
```

Die Ausgabe dieser Operation sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
      "Description": "test",
      "NodeGroups": [
        {
          "Status": "available",
          "NodeGroupMembers": [
            {
              "CurrentRole": "primary",
```

```

        "PreferredAvailabilityZone": "us-west-2a",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Port": 6379,
            "Address": "rg-name-001.1abc4d.0001.usw2.cache.amazonaws.com"
        },
        "CacheClusterId": "rg-name-001"
    },
    {
        "CurrentRole": "replica",
        "PreferredAvailabilityZone": "us-west-2b",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Port": 6379,
            "Address": "rg-name-002.1abc4d.0001.usw2.cache.amazonaws.com"
        },
        "CacheClusterId": "rg-name-002"
    },
    {
        "CurrentRole": "replica",
        "PreferredAvailabilityZone": "us-west-2c",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Port": 6379,
            "Address": "rg-name-003.1abc4d.0001.usw2.cache.amazonaws.com"
        },
        "CacheClusterId": "rg-name-003"
    }
],
"NodeGroupId": "0001",
"PrimaryEndpoint": {
    "Port": 6379,
    "Address": "rg-name.1abc4d.ng.0001.usw2.cache.amazonaws.com"
}
}
],
"ReplicationGroupId": "rg-name",
"AutomaticFailover": "enabled",
"SnapshottingClusterId": "rg-name-002",
"MemberClusters": [
    "rg-name-001",
    "rg-name-002",
    "rg-name-003"
],

```

```
    "PendingModifiedValues": {}  
  },  
  {  
    ... some output omitted for brevity  
  }  
]  
}
```

Weitere Informationen finden Sie im ElastiCache Thema AWS CLI für [describe-replication-groups](#).

Details einer Replikationsgruppe anzeigen (ElastiCache API)

Sie können die Details für eine Replikation mithilfe des AWS CLI `DescribeReplicationGroups` Vorgangs anzeigen. Verfeinern Sie die Auflistung mit den folgenden optionalen Parametern. Wenn die Parameter weggelassen werden, werden Details für bis zu 100 Replikationsgruppen zurückgegeben.

Optionale Parameter

- `ReplicationGroupId` – Verwenden Sie diesen Parameter zur Auflistung der Details einer bestimmten Replikationsgruppe. Wenn die angegebene Replikationsgruppe mehr als eine Knotengruppe besitzt, werden die zurückgegebenen Ergebnisse nach Knotengruppe gruppiert.
- `MaxRecords` – Verwenden Sie diesen Parameter, um die Anzahl der aufgelisteten Replikationsgruppen einzuschränken. Der Wert für `MaxRecords` darf nicht kleiner als 20 oder größer als 100 sein. Der Standardwert ist 100.

Example

Der folgende Code listet die Details für bis zu 100 Replikationsgruppen auf.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Der folgende Code listet die Details für `myReplGroup` auf.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=DescribeReplicationGroups
&ReplicationGroupId=myReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Der folgende Code listet die Details für bis zu 25 Cluster auf.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroups
&MaxRecords=25
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie im ElastiCache API-Referenzthema [DescribeReplicationGroups](#).

Suchen von Endpunkten von Replikationsgruppen

Eine Anwendung kann eine Verbindung zu jedem beliebigen Knoten in einer Replikationsgruppe herstellen, vorausgesetzt, dass es über den DNS-Endpunkt und die Portnummer für den betreffenden Knoten verfügt. Je nachdem, ob Sie eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) oder eine Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus aktiviert) ausführen, sind Sie an verschiedenen Endpunkten interessiert.

Valkey oder Redis OSS (Clustermodus deaktiviert)

Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) mit Replikaten haben drei Arten von Endpunkten: den primären Endpunkt, den Reader-Endpunkt und die Knotenendpunkte. Der primäre Endpunkt ist ein DNS-Name, der immer auf den primären Knoten im Cluster aufgelöst wird. Der primäre Endpunkt ist Änderungen an Ihrem Cluster, wie Heraufstufen einer Read Replica in die Rolle des primären Knotens, gegenüber immun. Für Schreibvorgänge empfehlen wir, dass Anwendungen sich ausschließlich mit dem primären Endpunkt verbinden.

Ein Reader-Endpunkt verteilt eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Read Replicas in einem Cluster. ElastiCache Zusätzliche Faktoren, z. B. wenn die Anwendung die Verbindungen erstellt oder wie die Anwendung die Verbindungen (erneut) verwendet, bestimmen die Verteilung des Datenverkehrs. Reader-Endpunkte bleiben hinsichtlich der Cluster-Änderungen beim Hinzufügen oder Entfernen von Replicas in Echtzeit auf dem aktuellen Stand. Sie können die mehreren Read Replicas Ihres ElastiCache Redis OSS-Clusters in verschiedenen AWS Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Leser-Endpunkte sicherzustellen.

Note

Ein Leser-Endpunkt ist kein Load Balancer. Es handelt sich um einen DNS-Eintrag, der nach dem Rotationsprinzip zu einer IP-Adresse eines der Replikationsknoten aufgelöst wird.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpunkt werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren.

Valkey oder Redis OSS (Clustermodus aktiviert)

Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) mit Replikaten, da sie mehrere Shards (API/CLI: Knotengruppen) haben, was bedeutet, dass sie auch mehrere Primärknoten haben, haben

eine andere Endpunktstruktur als Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert). Valkey oder Redis OSS (Clustermodus aktiviert) haben einen Konfigurationsendpunkt, der alle Primär- und Knotenendpunkte im Cluster „kennt“. Ihre Anwendung stellt eine Verbindung zum Konfigurationsendpunkt her. Immer wenn Ihre Anwendung auf den Konfigurationsendpunkt des Clusters schreibt oder von diesem liest, bestimmen Valkey und Redis OSS im Hintergrund, zu welchem Shard der Schlüssel gehört und welcher Endpunkt in diesem Shard verwendet werden soll. Dies läuft alles transparent für Ihre Anwendung ab.

Sie können die Endpunkte für einen Cluster mithilfe der ElastiCache Konsole, der oder der API finden. [AWS CLI ElastiCache](#)

Suchen von Replikationsgruppen-Endpunkten

Anweisungen zur Suche nach den Endpunkten für Ihre Replikationsgruppe finden Sie in einem der folgenden Themen:

- [Finden Sie die Endpunkte eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden von Endpunkten für einen Valkey- oder Redis OSS-Cluster \(Clustermodus aktiviert\) \(Konsole\)](#)
- [Finden der Endpunkte für Valkey- oder Redis OSS-Replikationsgruppen \(AWS CLI\)](#)
- [Suche nach Endpunkten für Valkey- oder Redis OSS-Replikationsgruppen \(API\) ElastiCache](#)

Ändern einer Replikationsgruppe

Wichtige Einschränkungen

- ElastiCache unterstützt derzeit begrenzte Änderungen einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert), z. B. das Ändern der Engine-Version mithilfe der API-Operation `ModifyReplicationGroup` (CLI: `modify-replication-group`). Sie können die Anzahl der Shards (Knotengruppen) in einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mit der API-Operation [ModifyReplicationGroupShardConfiguration](#) (CLI: `modify-replication-group-shard-configuration`) ändern. [Weitere Informationen finden Sie unter Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#).

Andere Änderungen an einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) erfordern, dass Sie einen Cluster mit dem neuen Cluster erstellen, der die Änderungen enthält.

- Sie können Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) und Valkey- oder Redis OSS-Cluster und Replikationsgruppen (Clustermodus aktiviert) auf neuere Engine-Versionen aktualisieren. Ein Downgrade auf vorhergehende Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters oder der vorhandenen Replikationsgruppe und der anschließenden Erstellung eines neuen Clusters oder einer neuen Replikationsgruppe möglich. Weitere Informationen finden Sie unter [Versionsverwaltung für ElastiCache](#).
- Sie können einen vorhandenen OSS-Cluster ElastiCache für Valkey oder Redis, der den Clustermodus deaktiviert verwendet, mithilfe der Konsole, der [ModifyReplicationGroup](#) API oder des `modify-replication-group` CLI-Befehls aktualisieren, um den Clustermodus aktiviert zu verwenden, wie im folgenden Beispiel gezeigt. Sie können auch den Schritten unter [Ändern des Clustermodus](#) folgen.

Sie können die Einstellungen eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) mithilfe der ElastiCache Konsole, der oder der AWS CLI API ändern. ElastiCache unterstützt derzeit eine begrenzte Anzahl von Änderungen an einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert). Andere Änderungen erfordern, dass Sie eine Sicherungskopie der aktuellen Replikationsgruppe erstellen und diese Sicherung dann verwenden, um eine neue Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) zu erstellen.

Themen

- [Unter Verwendung der AWS-Managementkonsole](#)
- [Verwenden des AWS CLI](#)
- [Verwenden der API ElastiCache](#)

Unter Verwendung der AWS-Managementkonsole

Informationen zum Ändern eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) finden Sie unter: [Einen ElastiCache Cluster ändern](#)

Verwenden des AWS CLI

Im Folgenden finden Sie AWS CLI Beispiele für den `modify-replication-group` Befehl. Sie können mit demselben Befehl andere Änderungen an einer Replikationsgruppe vornehmen.

Aktivieren Sie Multi-AZ für eine bestehende Valkey- oder Redis OSS-Replikationsgruppe:

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --multi-az-enabled = true
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --multi-az-enabled
```

Ändern des Clustermodus von deaktiviert auf aktiviert:

Wenn Sie den Clustermodus von deaktiviert auf aktiviert ändern möchten, müssen Sie zuerst den Clustermodus auf kompatibel festlegen. Im kompatiblen Modus können sich Ihre Valkey- oder Redis OSS-Clients sowohl mit aktiviertem als auch mit deaktiviertem Clustermodus verbinden. Nachdem Sie alle Valkey- oder Redis OSS-Clients zur Verwendung des aktivierten Clustermodus migriert haben, können Sie die Konfiguration des Clustermodus abschließen und den Clustermodus auf aktiviert setzen.

Für Linux, macOS oder Unix:

Legen Sie den Clustermodus auf kompatibel fest.

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --cache-parameter-group-name myParameterGroupName \  
  --cluster-mode compatible
```

Legen Sie den Clustermodus auf aktiviert fest.

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --cluster-mode enabled
```

Für Windows:

Legen Sie den Clustermodus auf kompatibel fest.

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --cache-parameter-group-name myParameterGroupName ^  
  --cluster-mode compatible
```

Legen Sie den Clustermodus auf aktiviert fest.

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --cluster-mode enabled
```

Weitere Informationen zu diesem AWS CLI `modify-replication-group` Befehl finden Sie unter [modify-replication-group](#) oder [Ändern des Clustermodus](#) im Benutzerhandbuch ElastiCache für Redis OSS.

Verwenden der API ElastiCache

Der folgende ElastiCache API-Vorgang aktiviert Multi-AZ auf einer vorhandenen Valkey- oder Redis OSS-Replikationsgruppe. Sie können mit derselben Operation andere Änderungen an einer Replikationsgruppe vornehmen.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyReplicationGroup
```

```
&AutomaticFailoverEnabled=true
&Mutli-AZEnabled=true
&ReplicationGroupId=myReplGroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Weitere Informationen zum ElastiCache ModifyReplicationGroup API-Vorgang finden Sie unter [ModifyReplicationGroup](#)

Löschen einer Replikationsgruppe

Wenn Sie einen Ihrer Cluster mit Replikaten (in API/CLI als Replikationsgruppen bezeichnet) nicht mehr benötigen, können Sie ihn löschen. Wenn Sie eine Replikationsgruppe ElastiCache löschen, werden alle Knoten in dieser Gruppe gelöscht.

Sobald die Operation gestartet wurde, kann sie nicht mehr unterbrochen oder abgebrochen werden.

Warning

- Wenn Sie einen ElastiCache für Redis OSS-Cluster löschen, bleiben Ihre manuellen Snapshots erhalten. Sie erhalten außerdem die Möglichkeit, einen letzten Snapshot zu erstellen, bevor der Cluster gelöscht wird. Automatisch erstellte Cache-Snapshots werden nicht aufbewahrt.
- `CreateSnapshot` Für die Erstellung eines endgültigen Snapshots ist eine Genehmigung erforderlich. Ohne diese Genehmigung schlägt der API-Aufruf mit einer `Access Denied` Ausnahme fehl.

Löschen einer Replikationsgruppe (Konsole)

Zum Löschen eines Clusters mit Replikaten informieren Sie sich unter [Löschen eines Clusters in ElastiCache](#).

Löschen einer Replikationsgruppe (AWS CLI)

Mit dem Befehl [delete-replication-group](#) können Sie eine Replikationsgruppe löschen.

```
aws elasticache delete-replication-group --replication-group-id my-repgroup
```

Sie werden aufgefordert, Ihre Entscheidung zu bestätigen. Geben Sie y (Ja) ein, um die Operation sofort zu starten. Einmal gestartet kann der Vorgang nicht mehr rückgängig gemacht werden.

```
After you begin deleting this replication group, all of its nodes will be deleted as well.
```

```
Are you sure you want to delete this replication group? [Ny]y
```

```
REPLICATIONGROUP my-repgroup My replication group deleting
```

Löschen einer Replikationsgruppe (ElastiCache API)

Rufen Sie [DeleteReplicationGroup](#) mit dem Parameter `ReplicationGroup` auf.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DeleteReplicationGroup  
&ReplicationGroupId=my-repgroup  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Note

Wenn Sie den Parameter `RetainPrimaryCluster` auf `true` einstellen, werden alle Read Replicas gelöscht, der primäre Cluster wird jedoch beibehalten.

Ändern der Anzahl von Replikaten

Sie können die Anzahl der Read Replicas in Ihrer Valkey- oder Redis OSS-Replikationsgruppe mithilfe der AWS-Managementkonsole, der oder der API dynamisch erhöhen oder verringern. AWS CLI ElastiCache Wenn es sich bei Ihrer Replikationsgruppe um eine Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus aktiviert) handelt, können Sie auswählen, mit welchen Shards (Knotengruppen) die Anzahl der Replikate erhöht oder verringert werden soll.

Um die Anzahl der Replikate in Ihrer Replikationsgruppe dynamisch zu ändern, wählen Sie den Vorgang aus der folgenden Tabelle aus, der zu Ihrer Situation passt.

So gehen Sie vor	Für Valkey oder Redis OSS (Clustermodus aktiviert)	Für Valkey oder Redis OSS (Clustermodus deaktiviert)
Replikate hinzufügen	Erhöhen der Anzahl der Replikate in einem Shard	Erhöhen der Anzahl der Replikate in einem Shard Hinzufügen einer Read Replica für Valkey oder Redis OSS (Cluster-Modus deaktiviert)
Replikate löschen	Verringern der Anzahl der Replikate in einem Shard	Verringern der Anzahl der Replikate in einem Shard Löschen einer Read Replica für Valkey oder Redis OSS (Cluster-Modus deaktiviert)

Erhöhen der Anzahl der Replikate in einem Shard

Sie können die Anzahl der Replikate in einer Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus aktiviert) oder einer Valkey- oder Redis-OSS-Replikationsgruppe (Cluster-Modus deaktiviert) auf maximal fünf erhöhen. Sie können dazu die, oder die API AWS-Managementkonsole verwenden.

AWS CLI ElastiCache

Themen

- [Mit dem AWS-Managementkonsole](#)
- [Unter Verwendung der AWS CLI](#)
- [Verwendung der API ElastiCache](#)

Mit dem AWS-Managementkonsole

Das folgende Verfahren verwendet die Konsole, um die Anzahl der Replikate in einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) zu erhöhen.

Um die Anzahl der Replikate in Shards zu erhöhen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS und dann den Namen der Replikationsgruppe aus, zu der Sie Replikate hinzufügen möchten.
3. Aktivieren Sie das Kontrollkästchen eines jeden Shards, den Sie zu Replikaten hinzufügen möchten.
4. Wählen Sie Add replicas (Replikate hinzufügen).
5. Füllen Sie die Seite Add Replicas to Shards (Replikate zu Shards hinzufügen) aus:
 - Geben Sie für New number of replicas/shard (Neue Anzahl der Replikate pro Shard) die Anzahl der Replikate ein, über die alle Ihre ausgewählten Shards verfügen sollen. Dieser Wert muss größer oder gleich dem Wert für Current Number of Replicas per shard (Aktuelle Anzahl der Replicas pro Shard) und kleiner oder gleich fünf sein. Als funktionelles Minimum empfehlen wir mindestens zwei Replikate.
 - Wählen Sie für Availability Zones entweder Keine Präferenz aus, um für jedes neue Replikat eine Availability Zone ElastiCache ausgewählt zu haben, oder Availability Zones angeben, um für jedes neue Replikat eine Availability Zone auszuwählen.

Geben Sie bei Auswahl von Availability Zones an, für jedes neue Replikat eine Availability Zone aus der Liste an.

6. Klicken Sie auf Add (Hinzufügen), um die Replikate hinzuzufügen, oder auf Cancel (Abbrechen), um den Vorgang abzubrechen.

Unter Verwendung der AWS CLI

Um die Anzahl der Replikate in einem Valkey- oder Redis OSS-Shard zu erhöhen, verwenden Sie den `increase-replica-count` Befehl mit den folgenden Parametern:

- `--replication-group-id` – Erforderlich. Identifiziert, in welcher Replikationsgruppe Sie die Anzahl von Replikaten erhöhen möchten.
- `--apply-immediately` oder `--no-apply-immediately` – Erforderlich. Gibt an, ob die Replikatanzahl sofort (`--apply-immediately`) oder während des nächsten Wartungsfensters (`--no-apply-immediately`) erhöht werden soll. `--no-apply-immediately` wird derzeit nicht unterstützt.
- `--new-replica-count` – Optional. Gibt die gewünschte Anzahl von Replikatknoten nach der Fertigstellung an, bis zu maximal fünf. Verwenden Sie diesen Parameter für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert), wenn es nur eine Knotengruppe oder Valkey- oder Redis OSS-Gruppe (Clustermodus aktiviert) gibt oder wenn Sie möchten, dass alle Knotengruppen dieselbe Anzahl von Replikaten haben. Wenn dieser Wert nicht größer als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.
- `--replica-configuration` – Optional. Ermöglicht Ihnen, die Anzahl der Replikate und Availability Zones für jede Knotengruppe separat festzulegen. Verwenden Sie diesen Parameter für Valkey- oder Redis OSS-Gruppen (Clustermodus aktiviert), bei denen Sie jede Knotengruppe unabhängig konfigurieren möchten.

`--replica-configuration` verfügt über drei optionale Mitglieder:

- `NodeId` Die vierziffrige ID der Knotengruppe, die Sie derzeit konfigurieren. Für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert) ist die Shard-ID immer `0001`. Informationen zur ID einer Valkey- oder Redis OSS-Knotengruppe (Shards) (Clustermodus aktiviert) finden Sie unter [Auffinden der ID eines Shards](#).
- `NewReplicaCount` Die Anzahl der Replikate, die sich am Ende dieser Operation in dieser Knotengruppe befinden sollen. Der Wert muss höher als die aktuelle Anzahl der Replikate sein.

und darf maximal fünf betragen. Wenn dieser Wert nicht größer als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.

- **PreferredAvailabilityZones** Eine Liste der PreferredAvailabilityZone-Zeichenfolgen, die angeben, in welchen Availability Zones sich die Knoten der Replikationsgruppe befinden sollen. Die Anzahl der PreferredAvailabilityZone-Werte muss gleich dem Wert von NewReplicaCount plus 1 sein, um dem primären Knoten Rechnung zu tragen. Wenn dieses Mitglied von weggelassen --replica-configuration wird, wählt OSS ElastiCache für Redis die Availability Zone für jedes der neuen Replikate.

Important

Sie müssen in Ihren Aufruf entweder den Parameter --new-replica-count oder den Parameter --replica-configuration, aber nicht beide, einbeziehen.

Example

Im folgenden Beispiel wird die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf drei erhöht. Wenn das Beispiel abgeschlossen ist, befinden sich in jeder Knotengruppe drei Replikate. Diese Zahl gilt unabhängig davon, ob es sich um eine Valkey- oder Redis OSS-Gruppe (Clustermodus deaktiviert) mit einer einzelnen Knotengruppe oder um eine Valkey- oder Redis OSS-Gruppe (Clustermodus aktiviert) mit mehreren Knotengruppen handelt.

Für Linux, macOS oder Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id sample-repl-group \  
  --new-replica-count 3 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id sample-repl-group ^  
  --new-replica-count 3 ^  
  --apply-immediately
```

Im folgenden Beispiel erhöht sich die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf den Wert, der für die beiden angegebenen Knotengruppen festgelegt ist. Da es mehrere Knotengruppen gibt, handelt es sich um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert). Wenn optional ein Wert für `PreferredAvailabilityZones` angegeben wird, muss die Anzahl der aufgelisteten Availability Zones gleich dem Wert von `NewReplicaCount` plus 1 sein. Diese Methode trägt dem primären Knoten für die durch `NodeGroupId` identifizierte Gruppe Rechnung.

Für Linux, macOS oder Unix:

```
aws elasticache increase-replica-count \
  --replication-group-id sample-repl-group \
  --replica-configuration \
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b \
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
  --apply-immediately
```

Für Windows:

```
aws elasticache increase-replica-count ^
  --replication-group-id sample-repl-group ^
  --replica-configuration ^
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b ^
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
  --apply-immediately
```

Weitere Informationen zur Erhöhung der Anzahl von Replikaten mithilfe der CLI finden Sie [increase-replica-count](#) in der ElastiCache Amazon-Befehlszeilenreferenz.

Verwendung der API ElastiCache

Verwenden Sie die `IncreaseReplicaCount` Aktion mit den folgenden Parametern, um die Anzahl der Replikate in einem Valkey- oder Redis OSS-Shard zu erhöhen:

- `ReplicationGroupId` – Erforderlich. Identifiziert, in welcher Replikationsgruppe Sie die Anzahl von Replikaten erhöhen möchten.

- **ApplyImmediately** – Erforderlich. Gibt an, ob die Replikatanzahl sofort (**ApplyImmediately=True**) oder während des nächsten Wartungsfensters (**ApplyImmediately=False**) erhöht werden soll. **ApplyImmediately=False** wird derzeit nicht unterstützt.
- **NewReplicaCount** – Optional. Gibt die gewünschte Anzahl von Replikatknoten nach der Fertigstellung an, bis zu maximal fünf. Verwenden Sie diesen Parameter für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert), bei denen es nur eine Knotengruppe gibt, oder für Valkey- oder Redis OSS-Gruppen (Clustermodus aktiviert), bei denen alle Knotengruppen dieselbe Anzahl von Replikaten haben sollen. Wenn dieser Wert nicht größer als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.
- **ReplicaConfiguration** – Optional. Ermöglicht Ihnen, die Anzahl der Replikate und Availability Zones für jede Knotengruppe separat festzulegen. Verwenden Sie diesen Parameter für Valkey- oder Redis OSS-Gruppen (Clustermodus aktiviert), bei denen Sie jede Knotengruppe unabhängig konfigurieren möchten.

ReplicaConfiguration verfügt über drei optionale Mitglieder:

- **NodeId** Die vierstellige ID für die Knotengruppe, die Sie derzeit konfigurieren. Für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert) ist die Knotengruppen-ID (Shard) immer `0001`. Informationen zur ID einer Valkey- oder Redis OSS-Knotengruppe (Shards) (Clustermodus aktiviert) finden Sie unter [Auffinden der ID eines Shards](#).
- **NewReplicaCount** Die Anzahl der Replikate, die sich am Ende dieser Operation in dieser Knotengruppe befinden sollen. Der Wert muss höher als die aktuelle Anzahl der Replikate sein und darf maximal fünf betragen. Wenn dieser Wert nicht größer als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.
- **PreferredAvailabilityZones** Eine Liste der **PreferredAvailabilityZone**-Zeichenfolgen, die angeben, in welchen Availability Zones sich die Knoten der Replikationsgruppe befinden sollen. Die Anzahl der **PreferredAvailabilityZone**-Werte muss gleich dem Wert von **NewReplicaCount** plus 1 sein, um dem primären Knoten Rechnung zu tragen. Wenn dieses Mitglied von **ReplicaConfiguration** weggelassen wird, wählt OSS ElastiCache für Redis die Availability Zone für jedes der neuen Replikate.

 Important

Sie müssen in Ihren Aufruf entweder den Parameter **NewReplicaCount** oder den Parameter **ReplicaConfiguration**, aber nicht beide, einbeziehen.

Example

Im folgenden Beispiel wird die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf drei erhöht. Wenn das Beispiel abgeschlossen ist, befinden sich in jeder Knotengruppe drei Replikate. Diese Zahl gilt unabhängig davon, ob es sich um eine Valkey- oder Redis OSS-Gruppe (Clustermodus deaktiviert) mit einer einzelnen Knotengruppe oder um eine Valkey- oder Redis OSS-Gruppe (Clustermodus aktiviert) mit mehreren Knotengruppen handelt.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=IncreaseReplicaCount  
  &ApplyImmediately=True  
  &NewReplicaCount=3  
  &ReplicationGroupId=sample-repl-group  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Im folgenden Beispiel erhöht sich die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf den Wert, der für die beiden angegebenen Knotengruppen festgelegt ist. Da es mehrere Knotengruppen gibt, handelt es sich um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert). Wenn optional ein Wert für `PreferredAvailabilityZones` angegeben wird, muss die Anzahl der aufgelisteten Availability Zones gleich dem Wert von `NewReplicaCount` plus 1 sein. Diese Methode trägt dem primären Knoten für die durch `NodeGroupId` identifizierte Gruppe Rechnung.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=IncreaseReplicaCount  
  &ApplyImmediately=True  
  &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001  
  &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=2  
  
  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=  
east-1a  
  
  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=  
east-1c  
  
  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.3=  
east-1b
```

```
&ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
&ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=3

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1c

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
  &ReplicationGroupId=sample-repl-group
  &Version=2015-02-02
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20150202T192317Z
  &X-Amz-Credential=<credential>
```

Weitere Informationen zur Erhöhung der Anzahl von Replikaten, die die API verwenden, finden Sie [IncreaseReplicaCount](#) in der Amazon ElastiCache API-Referenz.

Verringern der Anzahl der Replikate in einem Shard

Sie können die Anzahl der Replikate in einem Shard für Valkey oder Redis OSS (Clustermodus aktiviert) oder in einer Replikationsgruppe für Valkey oder Redis OSS (Clustermodus deaktiviert) verringern:

- Für Valkey oder Redis OSS (Clustermodus deaktiviert) können Sie die Anzahl der Replikate auf eins reduzieren, wenn Multi-AZ aktiviert ist, und auf Null, wenn es nicht aktiviert ist.
- Für Valkey oder Redis OSS (Clustermodus aktiviert) können Sie die Anzahl der Replikate auf Null reduzieren. Es ist jedoch kein Failover auf ein Replikat möglich, wenn Ihr primärer Knoten ausfällt.

Sie können die AWS-Managementkonsole, die AWS CLI oder die ElastiCache API verwenden, um die Anzahl der Replikate in einer Knotengruppe (Shard) oder Replikationsgruppe zu verringern.

Themen

- [Mit dem AWS-Managementkonsole](#)
- [Mit dem AWS CLI](#)
- [Verwendung der API ElastiCache](#)

Mit dem AWS-Managementkonsole

Das folgende Verfahren verwendet die Konsole, um die Anzahl der Replikate in einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) zu verringern.

Um die Anzahl der Replikate in einem Valkey- oder Redis OSS-Shard zu verringern

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS und dann den Namen der Replikationsgruppe aus, aus der Sie Replikate löschen möchten.
3. Aktivieren Sie das Kontrollkästchen eines jeden Shards, aus dem Sie einen Replikatknoten entfernen möchten.
4. Wählen Sie Delete replicas (Replikate löschen).
5. Füllen Sie die Seite Delete Replicas from to Shards (Replikate aus Shards löschen) aus:
 - a. Geben Sie für New number of replicas/shard (Neue Anzahl der Replikate pro Shard) die Anzahl der Replikate ein, über die die ausgewählten Shards verfügen sollen. Dieser Wert

muss größer oder gleich 1 sein. Als funktionelles Minimum empfehlen wir mindestens zwei Replikate pro Shard.

- b. Klicken Sie auf Delete (Löschen), um die Replikate zu löschen, oder auf Cancel (Abbrechen), um den Vorgang abubrechen.

Important

- Wenn Sie die zu löschenden Replikatknoten nicht angeben, wählt OSS ElastiCache für Redis automatisch Replikatknoten zum Löschen aus. Dabei versucht OSS ElastiCache für Redis, die Multi-AZ-Architektur für Ihre Replikationsgruppe beizubehalten, gefolgt von der Beibehaltung von Replikaten mit minimaler Replikationsverzögerung gegenüber der Primärgruppe.
- Primäre oder Master-Knoten in einer Replikationsgruppe können nicht gelöscht werden. Wenn Sie einen primären Knoten für die Löschoption angeben, schlägt dies mit einem Fehlerereignis fehl, das darauf hinweist, dass der primäre Knoten zum Löschen ausgewählt wurde.

Mit dem AWS CLI

Um die Anzahl der Replikate in einem Valkey- oder Redis OSS-Shard zu verringern, verwenden Sie den `decrease-replica-count` Befehl mit den folgenden Parametern:

- `--replication-group-id` – Erforderlich. Identifiziert, in welcher Replikationsgruppe Sie die Anzahl der Replikate verringern möchten.
- `--apply-immediately` oder `--no-apply-immediately` – Erforderlich. Gibt an, ob die Replikatanzahl sofort (`--apply-immediately`) oder während des nächsten Wartungsfensters (`--no-apply-immediately`) verringert werden soll. `--no-apply-immediately` wird derzeit nicht unterstützt.
- `--new-replica-count` – Optional. Gibt die Anzahl der von Ihnen gewünschten Replikatknoten an. Der Wert `--new-replica-count` muss ein gültiger Wert kleiner als die aktuelle Anzahl der Replikate in den Knotengruppen sein. Informationen zu den zulässigen Mindestwerten finden Sie unter [Verringern der Anzahl der Replikate in einem Shard](#). Wenn der Wert von `--new-replica-count` dieser Anforderung nicht entspricht, schlägt der Aufruf fehl.
- `--replicas-to-remove` – Optional. Enthält eine Liste von Knoten, die die zu entfernenden Replikatknoten IDs angeben.

- `--replica-configuration` – Optional. Ermöglicht Ihnen, die Anzahl der Replikate und Availability Zones für jede Knotengruppe separat festzulegen. Verwenden Sie diesen Parameter für Valkey- oder Redis OSS-Gruppen (Clustermodus aktiviert), bei denen Sie jede Knotengruppe unabhängig konfigurieren möchten.

`--replica-configuration` verfügt über drei optionale Mitglieder:

- `NodeId` Die vierziffrige ID der Knotengruppe, die Sie derzeit konfigurieren. Für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert) ist die Shard-ID immer `0001`. Informationen zur ID einer Valkey- oder Redis OSS-Knotengruppe (Shards) (Clustermodus aktiviert) finden Sie unter [Auffinden der ID eines Shards](#).
- `NewReplicaCount` Ein optionaler Parameter, der die Anzahl der von Ihnen gewünschten Replikatknoten angibt. Der Wert `NewReplicaCount` muss ein gültiger Wert kleiner als die aktuelle Anzahl der Replikate in den Knotengruppen sein. Informationen zu den zulässigen Mindestwerten finden Sie unter [Verringern der Anzahl der Replikate in einem Shard](#). Wenn der Wert von `NewReplicaCount` dieser Anforderung nicht entspricht, schlägt der Aufruf fehl.
- `PreferredAvailabilityZones` Eine Liste von `PreferredAvailabilityZone`-Zeichenfolgen, die angeben, in welchen Availability Zones sich die Knoten der Replikationsgruppen befinden. Die Anzahl der `PreferredAvailabilityZone`-Werte muss gleich dem Wert von `NewReplicaCount` plus 1 sein, um dem primären Knoten Rechnung zu tragen. Wenn dieses Mitglied von weggelassen `--replica-configuration` wird, wählt OSS ElastiCache für Redis die Availability Zone für jedes der neuen Replikate.

Important

Sie dürfen nur genau einen der `--new-replica-count`-, `--replicas-to-remove`- oder `--replica-configuration`-Parameter einschließen.

Example

Im folgenden Beispiel wird mit `--new-replica-count` die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` um eins verringert. Wenn das Beispiel abgeschlossen ist, befindet sich ein Replikat in jeder Knotengruppe. Diese Zahl gilt unabhängig davon, ob es sich um eine Valkey- oder Redis OSS-Gruppe (Clustermodus deaktiviert) mit einer einzelnen Knotengruppe oder um eine Valkey- oder Redis OSS-Gruppe (Clustermodus aktiviert) mit mehreren Knotengruppen handelt.

Für Linux, macOS oder Unix:

```
aws elasticache decrease-replica-count
  --replication-group-id sample-repl-group \
  --new-replica-count 1 \
  --apply-immediately
```

Für Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id sample-repl-group ^
  --new-replica-count 1 ^
  --apply-immediately
```

Im folgenden Beispiel wird die Anzahl der Replikate in der Replikationsgruppe *sample-repl-group* verringert, indem zwei bestimmte Replikate (*0001* und *0003*) aus der Knotengruppe entfernt werden.

Für Linux, macOS oder Unix:

```
aws elasticache decrease-replica-count \
  --replication-group-id sample-repl-group \
  --replicas-to-remove 0001,0003 \
  --apply-immediately
```

Für Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id sample-repl-group ^
  --replicas-to-remove 0001,0003 \
  --apply-immediately
```

Im folgenden Beispiel wird mit `--replica-configuration` die Anzahl der Replikate in der Replikationsgruppe *sample-repl-group* auf den Wert verringert, der für die beiden angegebenen Knotengruppen festgelegt ist. Da es mehrere Knotengruppen gibt, handelt es sich um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert). Wenn optional ein Wert für `PreferredAvailabilityZones` angegeben wird, muss die Anzahl der aufgelisteten Availability Zones gleich dem Wert von `NewReplicaCount` plus 1 sein. Diese Methode trägt dem primären Knoten für die durch `NodeGroupId` identifizierte Gruppe Rechnung.

Für Linux, macOS oder Unix:

```
aws elasticache decrease-replica-count \
  --replication-group-id sample-repl-group \
  --replica-configuration \
    NodeGroupId=0001,NewReplicaCount=1,PreferredAvailabilityZones=us-east-1a,us-east-1c \
    NodeGroupId=0003,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c \
  --apply-immediately
```

Für Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id sample-repl-group ^
  --replica-configuration ^
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c ^
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c \
  --apply-immediately
```

Weitere Informationen zur Verringerung der Anzahl von Replikaten mithilfe der CLI finden Sie [decrease-replica-count](#) in der ElastiCache Amazon-Befehlszeilenreferenz.

Verwendung der API ElastiCache

Verwenden Sie die DecreaseReplicaCount Aktion mit den folgenden Parametern, um die Anzahl der Replikate in einem Valkey- oder Redis OSS-Shard zu verringern:

- **ReplicationGroupId** – Erforderlich. Identifiziert, in welcher Replikationsgruppe Sie die Anzahl der Replikate verringern möchten.
- **ApplyImmediately** – Erforderlich. Gibt an, ob die Replikatanzahl sofort (**ApplyImmediately=True**) oder während des nächsten Wartungsfensters (**ApplyImmediately=False**) verringert werden soll. **ApplyImmediately=False** wird derzeit nicht unterstützt.
- **NewReplicaCount** – Optional. Gibt die Anzahl der von Ihnen gewünschten Replikatknoten an. Der Wert **NewReplicaCount** muss ein gültiger Wert kleiner als die aktuelle Anzahl der Replikate in den Knotengruppen sein. Informationen zu den zulässigen Mindestwerten finden Sie unter [Verringern der Anzahl der Replikate in einem Shard](#). Wenn der Wert von **--new-replica-count** dieser Anforderung nicht entspricht, schlägt der Aufruf fehl.

- **ReplicasToRemove** – Optional. Enthält eine Liste von Knoten, die die zu entfernenden Replikatknoten IDs angeben.
- **ReplicaConfiguration** – Optional. Enthält eine Liste von Knotengruppen, mit denen Sie die Anzahl der Replikate und Availability Zones separat für jede Knotengruppe festlegen können. Verwenden Sie diesen Parameter für Valkey- oder Redis OSS-Gruppen (Clustermodus aktiviert), bei denen Sie jede Knotengruppe unabhängig konfigurieren möchten.

ReplicaConfiguration verfügt über drei optionale Mitglieder:

- **NodeId** Die vierziffrige ID für die Knotengruppe, die Sie derzeit konfigurieren. Für Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert) ist die Knotengruppen-ID immer `0001`. Informationen zur ID einer Valkey- oder Redis OSS-Knotengruppe (Shards) (Clustermodus aktiviert) finden Sie unter [Auffinden der ID eines Shards](#)
- **NewReplicaCount** Die Anzahl der Replikate, die sich am Ende dieser Operation in dieser Knotengruppe befinden sollen. Der Wert muss weniger als die aktuelle Anzahl der Replikate sein. Der Mindestwert ist 1, wenn Multi-AZ aktiviert ist, bzw. 0, wenn Multi-AZ mit Automatic Failover nicht aktiviert ist. Wenn dieser Wert nicht kleiner als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.
- **PreferredAvailabilityZones** Eine Liste von **PreferredAvailabilityZone**-Zeichenfolgen, die angeben, in welchen Availability Zones sich die Knoten der Replikationsgruppen befinden. Die Anzahl der **PreferredAvailabilityZone**-Werte muss gleich dem Wert von **NewReplicaCount** plus 1 sein, um dem primären Knoten Rechnung zu tragen. Wenn dieses Mitglied von weggelassen **ReplicaConfiguration** wird, wählt OSS ElastiCache für Redis die Availability Zone für jedes der neuen Replikate.

Important

Sie dürfen nur genau einen der **NewReplicaCount**-, **ReplicasToRemove**- oder **ReplicaConfiguration**-Parameter einschließen.

Example

Im folgenden Beispiel wird mit **NewReplicaCount** die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` um eins verringert. Wenn das Beispiel abgeschlossen ist, befindet sich ein Replikat in jeder Knotengruppe. Diese Zahl gilt unabhängig davon, ob es sich um eine Valkey- oder Redis OSS-Gruppe (Clustermodus deaktiviert) mit einer einzelnen Knotengruppe

oder um eine Valkey- oder Redis OSS-Gruppe (Clustermodus aktiviert) mit mehreren Knotengruppen handelt.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DecreaseReplicaCount  
&ApplyImmediately=True  
&NewReplicaCount=1  
&ReplicationGroupId=sample-repl-group  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Im folgenden Beispiel wird die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` verringert, indem zwei bestimmte Replikate (`0001` und `0003`) aus der Knotengruppe entfernt werden.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DecreaseReplicaCount  
&ApplyImmediately=True  
&ReplicasToRemove.ReplicaToRemove.1=0001  
&ReplicasToRemove.ReplicaToRemove.2=0003  
&ReplicationGroupId=sample-repl-group  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Im folgenden Beispiel wird mit `ReplicaConfiguration` die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf den Wert verringert, der für die beiden angegebenen Knotengruppen festgelegt ist. Da es mehrere Knotengruppen gibt, handelt es sich um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert). Wenn optional ein Wert für `PreferredAvailabilityZones` angegeben wird, muss die Anzahl der aufgelisteten Availability Zones gleich dem Wert von `NewReplicaCount` plus 1 sein. Diese Methode trägt dem primären Knoten für die durch `NodeGroupId` identifizierte Gruppe Rechnung.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DecreaseReplicaCount  
&ApplyImmediately=True  
&ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
```

```
&ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=1

&ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

&ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c
    &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
    &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=2

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
    &ReplicationGroupId=sample-repl-group
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Weitere Informationen zur Verringerung der Anzahl von Replikaten, die die API verwenden, finden Sie [DecreaseReplicaCount](#) in der Amazon ElastiCache API-Referenz.

Hinzufügen einer Read Replica für Valkey oder Redis OSS (Cluster-Modus deaktiviert)

Die Informationen im folgenden Thema beziehen sich nur auf Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert).

Mit zunehmendem Datenverkehr durch Lesezugriffe empfiehlt es sich, diese Lesezugriffe über mehr Knoten zu verteilen und somit die Belastung durch Lesezugriffe auf jedem Knoten zu reduzieren. In diesem Thema erfahren Sie, wie Sie einem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) eine Read Replica hinzufügen.

Eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) kann maximal fünf Read Replicas haben. Bei dem Versuch, einer Replikationsgruppe mit bereits fünf Read Replicas ein weiteres Read Replica hinzuzufügen, schlägt die Operation fehl.

Informationen zum Hinzufügen von Replikaten zu einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) finden Sie im Folgenden:

- [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#)
- [Erhöhen der Anzahl der Replikate in einem Shard](#)

Sie können einem Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) mithilfe der ElastiCache Konsole, der oder der API eine Read Replica hinzufügen. AWS CLI ElastiCache

Verwandte Themen

- [Hinzufügen von Knoten zu einem ElastiCache Cluster](#)
- [Hinzufügen einer Read Replica zu einer Replikationsgruppe \(AWS CLI\)](#)
- [Hinzufügen eines Read Replica zu einer Replikationsgruppe mithilfe der API](#)

Hinzufügen einer Read Replica zu einer Replikationsgruppe (AWS CLI)

Um einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) eine Read Replica hinzuzufügen, verwenden Sie den AWS CLI `create-cache-cluster` Befehl mit dem Parameter, `--replication-group-id` um anzugeben, zu welcher Replikationsgruppe der Cluster (Knoten) hinzugefügt werden soll.

Im folgenden Beispiel wird der Cluster `my-read-replica` erstellt und zur Replikationsgruppe `my-replication-group` hinzugefügt. Die Knotentypen, Parametergruppen, Sicherheitsgruppen, das Wartungsfenster und andere Einstellungen für das Read Replica sind mit denen für die anderen Knoten in `my-replication-group` identisch.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id my-read-replica \  
  --replication-group-id my-replication-group
```

Für Windows:

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id my-read-replica ^  
  --replication-group-id my-replication-group
```


Weitere Informationen zum Hinzufügen einer Read Replica mithilfe der CLI finden Sie [create-cache-cluster](#) in der ElastiCache Amazon-Befehlszeilenreferenz.

Hinzufügen eines Read Replica zu einer Replikationsgruppe mithilfe der API

Um eine Read Replica zu einer Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus deaktiviert) hinzuzufügen, verwenden Sie den ElastiCache CreateCacheCluster Vorgang mit dem Parameter, ReplicationGroupId um anzugeben, zu welcher Replikationsgruppe der Cluster (Knoten) hinzugefügt werden soll.

Im folgenden Beispiel wird der Cluster myReadReplica erstellt und zur Replikationsgruppe myReplicationGroup hinzugefügt. Die Knotentypen, Parametergruppen, Sicherheitsgruppen, das Wartungsfenster und andere Einstellungen für das Read Replica sind mit denen für die anderen Knoten in myReplicationGroup identisch.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheCluster  
&CacheClusterId=myReadReplica  
&ReplicationGroupId=myReplicationGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen zum Hinzufügen einer Read Replica mithilfe der API finden Sie [CreateCacheCluster](#) in der Amazon ElastiCache API-Referenz.

Löschen einer Read Replica für Valkey oder Redis OSS (Cluster-Modus deaktiviert)

Die Informationen im folgenden Thema beziehen sich nur auf Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert).

Wenn sich der Lesedatenverkehr in Ihrer Valkey- oder Redis OSS-Replikationsgruppe ändert, möchten Sie möglicherweise Read Replicas hinzufügen oder entfernen. Das Entfernen eines Knotens aus einer -Replikationsgruppe ist mit dem Löschen eines Clusters vergleichbar, aber mit einigen Einschränkungen:

- Der primäre Knoten kann nicht aus einer Replikationsgruppe entfernt werden. Wenn Sie den primären Knoten entfernen möchten, verfahren Sie wie folgt:

1. Stufen Sie eine Read Replica zum primären Knoten herauf. Weitere Informationen zum Heraufstufen einer Read Replica zum primären Knoten finden Sie unter [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#).
 2. Löschen Sie den alten primären Knoten. Eine Einschränkung dieser Methode finden Sie im nächsten Gliederungspunkt.
- Wenn Multi-AZ für eine Replikationsgruppe aktiviert ist, kann das letzte Read Replica nicht aus der Replikationsgruppe entfernt werden. Führen Sie in diesem Fall folgende Schritte aus:
 1. Bearbeiten Sie die Replikationsgruppe, indem Sie Multi-AZ deaktivieren. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).
 2. Löschen Sie das Read Replica.

Sie können eine Read Replica mit der ElastiCache Konsole, dem For oder der API aus einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) entfernen. AWS CLI ElastiCache ElastiCache

Anweisungen zum Löschen eines Clusters aus einer Valkey- oder Redis OSS-Replikationsgruppe finden Sie im Folgenden:

- [Mit dem AWS-Managementkonsole](#)
- [Verwenden Sie den AWS CLI, um einen ElastiCache Cluster zu löschen](#)
- [Verwenden der ElastiCache API](#)
- [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#)
- [Verringern der Anzahl der Replikate in einem Shard](#)

Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey oder Redis OSS (Clustermodus deaktiviert)

Die Informationen im folgenden Thema beziehen sich nur auf Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus deaktiviert).

Sie können eine Valkey- oder Redis OSS-Lesereplik (Clustermodus deaktiviert) mithilfe der AWS-Managementkonsole, oder der API zur primären Replikation AWS CLI heraufstufen. ElastiCache Ein Read Replica kann nicht zum primären Knoten heraufgestuft werden, wenn für die -Replikationsgruppe Multi-AZ mit automatischem Failover aktiviert ist. Gehen Sie wie folgt vor, um ein Valkey- oder Redis OSS-Replikat (Clustermodus deaktiviert) in einer Multi-AZ-fähigen Replikationsgruppe zum Primärreplikat hochzustufen:

1. Bearbeiten Sie die Replikationsgruppe so, dass Multi-AZ deaktiviert wird (dazu müssen sich nicht alle Ihre Cluster in derselben Availability Zone befinden). Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).
2. Stufen Sie das Lesereplikat zum primären Knoten herauf.
3. Bearbeiten Sie die Replikationsgruppe so, dass Multi-AZ wieder aktiviert ist.

Multi-AZ ist für Replikationsgruppen, auf denen Redis OSS 2.6.13 oder früher ausgeführt wird, nicht verfügbar.

Unter Verwendung des AWS-Managementkonsole

Im folgenden Verfahren wird ein Replikatknoten mithilfe der Konsole zum primären Knoten heraufgestuft.

Heraufstufen eines Lesereplikats zur primären (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wenn das Replikat, das Sie heraufstufen möchten, Mitglied einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) ist, in der Multi-AZ aktiviert ist, ändern Sie die Replikationsgruppe so, dass Multi-AZ deaktiviert wird, bevor Sie fortfahren. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).
3. Wählen Sie Valkey oder Redis OSS aus und wählen Sie dann aus der Clusterliste die Replikationsgruppe aus, die Sie ändern möchten. Diese Replikationsgruppe muss die "Redis"-

Engine und nicht die "Clustered Redis"-Engine ausführen und muss aus mindestens zwei Knoten bestehen.

4. Wählen Sie aus der Liste der Knoten den Replikationsknoten aus, den Sie zum primären Knoten heraufstufen möchten. Wählen Sie dann für Actions (Aktionen) die Option Promote (Heraufstufen) aus.
5. Führen Sie im Dialogfeld Promote Read Replica (Read Replica heraufstufen) die folgenden Schritte aus:
 - a. Wählen Sie für Apply Immediately (Sofort anwenden) die Option Yes (Ja) aus, wenn das Read Replica sofort heraufgestuft werden soll, oder die Option No (Nein), um es im nächsten Wartungsfenster des Clusters heraufzustufen.
 - b. Wählen Sie Promote, um die Read Replica heraufzustufen, oder Cancel, um die Operation abubrechen.
6. Wenn für den Cluster vor Beginn des Hochstufens Multi-AZ aktiviert war, warten Sie, bis der Status der Replikationsgruppe available (verfügbar) lautet, bevor Sie den Cluster so bearbeiten, dass Multi-AZ wieder aktiviert wird. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

Unter Verwendung der AWS CLI

Sie können eine Read Replica nicht zum primären Knoten heraufstufen, wenn für die Replikationsgruppe Multi-AZ aktiviert ist. In einigen Fällen ist das heraufzustufende Replikat möglicherweise Teil einer Replikationsgruppe, in der Multi-AZ aktiviert ist. Sollte dies der Fall sein, muss Multi-AZ vor dem Fortfahren für die Replikationsgruppe deaktiviert werden. Für dieses Verfahren müssen sich nicht alle Ihre Cluster in der gleichen Availability Zone befinden. Weitere Informationen zum Ändern einer Replikationsgruppe finden Sie unter [Ändern einer Replikationsgruppe](#).

Mit dem folgenden AWS CLI Befehl wird die Replikationsgruppe geändert `sample-repl-group`, sodass die Read Replica zur primären Replica in `my-replica-1` der Replikationsgruppe wird.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id sample-repl-group \  
  --primary-cluster-id my-replica-1
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --primary-cluster-id my-replica-1
```

Weitere Informationen zum Ändern einer Replikationsgruppe finden Sie [modify-replication-group](#) in der Amazon ElastiCache Command Line Reference.

Verwenden der ElastiCache API

Sie können eine Read Replica nicht zum primären Knoten heraufstufen, wenn für die Replikationsgruppe Multi-AZ aktiviert ist. In einigen Fällen ist das heraufzustufende Replikat möglicherweise Teil einer Replikationsgruppe, in der Multi-AZ aktiviert ist. Sollte dies der Fall sein, muss Multi-AZ vor dem Fortfahren für die Replikationsgruppe deaktiviert werden. Für dieses Verfahren müssen sich nicht alle Ihre Cluster in der gleichen Availability Zone befinden. Weitere Informationen zum Ändern einer Replikationsgruppe finden Sie unter [Ändern einer Replikationsgruppe](#).

Mit der folgenden ElastiCache API-Aktion wird die Replikationsgruppe geändert `myReplGroup`, sodass die Read Replica zur primären Replica in `myReplica-1` der Replikationsgruppe wird.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup  
  &ReplicationGroupId=myReplGroup  
  &PrimaryClusterId=myReplica-1  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen zum Ändern einer Replikationsgruppe finden Sie [ModifyReplicationGroup](#) in der Amazon ElastiCache API-Referenz.

Verwaltung der ElastiCache Cluster-Wartung

Jeder Cluster verfügt über ein wöchentliches Wartungsfenster, während dem alle Systemänderungen angewendet werden. Bei Valkey und Redis OSS haben Replikationsgruppen dasselbe wöchentliche Wartungsfenster. Wenn Sie bei der Erstellung oder Änderung eines Clusters oder einer Replikationsgruppe kein bevorzugtes Wartungsfenster angeben, ElastiCache weisen Sie innerhalb des Wartungsfensters Ihrer Region an einem zufällig ausgewählten Wochentag ein 60-minütiges Wartungsfenster zu.

Das 60-minütige Wartungsfenster wird zufällig aus einem 8-Stunden-Zeitraum pro Region ausgewählt. Die folgende Tabelle listet die Blöcke für jede Region auf, von denen die Standard-Wartungsfenster zugewiesen werden. Sie können ein bevorzugtes Wartungsfenster außerhalb des Wartungsfensterblocks der Region auswählen.

Regionscode	Name der Region	Regionale Wartungsfenster
ap-northeast-1	Region Asien-Pazifik (Tokio)	13:00 - 21:00 UHR UTC
ap-northeast-2	Region Asien-Pazifik (Seoul)	12:00 - 20:00 UTC
ap-northeast-3	Region Asien-Pazifik (Osaka)	12:00 - 20:00 UTC
ap-southeast-3	Region Asien-Pazifik (Jakarta)	14:00 - 22:00 UHR UTC
ap-south-1	Region Asien-Pazifik (Mumbai)	17:30 - 1:30 UHR UTC
ap-southeast-1	Region Asien-Pazifik (Singapur)	14:00 - 22:00 UHR UTC
cn-north-1	Region China (Peking)	14:00 - 22:00 UHR UTC
cn-northwest-1	Region China (Ningxia)	14:00 - 22:00 UHR UTC
ap-east-1	Region Asien-Pazifik (Hongkong)	13:00 - 21:00 UTC
ap-southeast-2	Region Asien-Pazifik (Sydney)	12:00 - 20:00 UHR UTC
eu-west-3	Region Europa (Paris)	23:59 - 07:29 UTC
af-south-1	Region Afrika (Kapstadt)	13:00 - 21:00 UTC

Regionscode	Name der Region	Regionale Wartungsfenster
eu-central-1	Region Europa (Frankfurt)	23:00 - 07:00 UTC
eu-west-1	Region Europa (Irland)	22:00 bis 06:00 Uhr UTC
eu-west-2	Region Europa (London)	23:00 - 07:00 UTC
me-south-1	Region Naher Osten (Bahrain)	13:00 - 21:00 UTC
me-central-1	Region „Naher Osten (VAE)“	13:00 - 21:00 UTC
eu-south-1	Region Europa (Mailand)	21:00 - 05:00 UTC
sa-east-1	Region Südamerika (São Paulo)	01:00 - 09:00 UHR UTC
us-east-1	Region USA Ost (Nord-Virginia)	03:00 - 11:00 UHR UTC
us-east-2	Region USA Ost (Ohio)	04:00 - 12:00 UHR UTC
us-gov-west-1	AWS GovCloud (US) Region	06:00 bis 14:00 Uhr UTC
us-west-1	Region USA West (Nordkalifornien)	06:00 - 14:00 UHR UTC
us-west-2	Region USA West (Oregon)	06:00 bis 14:00 Uhr UTC

Das Wartungsfenster Ihres Clusters oder Ihrer Replikationsgruppe ändern

Das Wartungsfenster sollte in den Zeitraum mit der geringsten Nutzung fallen und daher unter Umständen von Zeit zu Zeit geändert werden. Sie können den Cluster oder die Replikationsgruppe ändern und einen Zeitraum mit einer Dauer von bis zu 24 Stunden festlegen, in dem alle angeforderten Wartungsaktivitäten durchgeführt werden sollen. In diesem Zeitraum werden alle verzögerten oder ausstehenden Cluster-Änderungen, die Sie angefordert haben, ausgeführt.

Note

Wenn Sie Änderungen am Knotentyp vornehmen möchten, aktivieren Sie das Kästchen **Jetzt anwenden**, indem AWS-Managementkonsole Sie die Option **and/or Engine-Upgrades sofort anwenden**. Andernfalls werden diese Änderungen während des nächsten geplanten

Wartungsfensters angewendet. Informationen zur Verwendung der API finden Sie unter [modify-replication-group](#) oder [modify-cache-cluster](#).

Weitere Informationen

Informationen zu Ihrem Wartungsfenster und dem Austausch von Knoten finden Sie unter:

- [ElastiCache Wartung](#) — Häufig gestellte Fragen zur Wartung und zum Austausch von Knoten
- [Knoten ersetzen \(Memcached\)](#) — Verwaltung des Knotenaustauschs für Memcached
- [Einen ElastiCache Cluster ändern](#) – Ändern des Wartungsfensters für einen Cluster
- [Knoten ersetzen \(Valkey und Redis OSS\)](#) – Verwalten des Knotenaustauschs
- [Ändern einer Replikationsgruppe](#) – Ändern des Wartungsfensters einer Replikationsgruppe

Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen

Amazon ElastiCache verwendet Parameter, um die Laufzeiteigenschaften Ihrer Knoten und Cluster zu steuern. In der Regel enthalten neuere Engine-Versionen zusätzliche Parameter zur Unterstützung der neueren Funktionalität. Tabellen mit Memcached-Parametern finden Sie unter. [Memcached-spezifische Parameter](#) Tabellen mit Valkey- und Redis-OSS-Parametern finden Sie unter. [Valkey- und Redis OSS-Parameter](#)

Wie zu erwarten, werden einige Parameterwerte, wie z. B. `maxmemory` durch die Engine und den Knotentyp bestimmt. Eine Tabelle dieser Memcached-Parameterwerte nach Knotentyp finden Sie unter. [Spezifische Parameter des Memcached-Knotentyps](#) Eine Tabelle dieser Valkey- und Redis OSS-Parameterwerte nach Knotentyp finden Sie unter. [Redis OSS-Knotentyp-spezifische Parameter](#)

Note

Eine Liste der Memcached-spezifischen Parameter finden Sie unter [Memcached-spezifische Parameter](#).

Themen

- [Parameterverwaltung in ElastiCache](#)

- [Cache-Parametergruppen werden in ElastiCache](#)
- [Eine ElastiCache Parametergruppe erstellen](#)
- [ElastiCache Parametergruppen nach Namen auflisten](#)
- [Werte einer ElastiCache Parametergruppe auflisten](#)
- [Änderung einer ElastiCache Parametergruppe](#)
- [Löschen einer ElastiCache Parametergruppe](#)
- [Motorspezifische Parameter](#)

Parameterverwaltung in ElastiCache

ElastiCache Parameter werden zur einfacheren Parameterverwaltung in benannten Parametergruppen zusammengefasst. Eine Parametergruppe stellt eine Kombination spezifischer Werte für die Parameter dar, die der Engine-Software während des Systemstarts übergeben werden. Diese Werte bestimmen, wie sich die Engine-Prozesse auf jedem Knoten während der Laufzeit verhalten. Die Parameterwerte für eine bestimmte Parametergruppe gelten für alle Knoten, die der betreffenden Gruppe zugehören. Dabei spielt es keine Rolle, zu welchem Cluster sie gehören.

Zur Feinabstimmung der Leistung Ihres Clusters können Sie einige Parameterwerte bearbeiten oder die Parametergruppe des Clusters ändern.

- Standard-Parametergruppen können nicht geändert oder gelöscht werden. Wenn Sie benutzerdefinierte Parameterwerte benötigen, müssen Sie eine benutzerdefinierte Parametergruppe erstellen.
- Für Memcached müssen die Parametergruppenfamilie und der Cluster, dem Sie sie zuweisen, kompatibel sein. Wenn Ihr Cluster z. B. Memcached Version 1.4.8 ausführt, können Sie nur Parametergruppen, Standard oder benutzerdefiniert, von der Memcached 1.4-Familie verwenden.

Für Redis OSS müssen die Parametergruppenfamilie und der Cluster, dem Sie sie zuweisen, kompatibel sein. Wenn auf Ihrem Cluster beispielsweise Redis OSS Version 3.2.10 ausgeführt wird, können Sie nur Standardparametergruppen oder benutzerdefinierte Parametergruppen aus der Redis OSS 3.2-Familie verwenden.

- Wenn Sie die Parametergruppe eines Clusters ändern, müssen die Werte aller bedingungsabhängig veränderbaren Parameter in der aktuellen und in der neuen Parametergruppe übereinstimmen.
- Wenn Sie bei Memcached die Parameter eines Clusters ändern, wird die Änderung sofort auf den Cluster angewendet. Dies gilt unabhängig davon, ob Sie die Parametergruppe des Clusters oder einen Parameterwert innerhalb der Parametergruppe des Clusters ändern. Anhand der Spalte **Changes Take Effect** (Änderungen treten in Kraft) in den Tabellen für [Memcached-spezifische Parameter](#) können Sie bestimmen, wann eine bestimmte Parameteränderung übernommen wird. Weitere Informationen zum Neustarten der Knoten eines Clusters finden Sie unter [Neustarten von Clustern](#).
- Wenn Sie bei Redis OSS die Parameter eines Clusters ändern, wird die Änderung entweder sofort oder, mit den unten genannten Ausnahmen, nach dem Neustart der Clusterknoten auf den Cluster angewendet. Dies gilt unabhängig davon, ob Sie die Parametergruppe des Clusters oder einen Parameterwert innerhalb der Parametergruppe des Clusters ändern. Anhand der Spalte **Changes**

Take Effect (Änderungen treten in Kraft) in den Tabellen für [Valkey- und Redis OSS-Parameter](#) können Sie bestimmen, wann eine bestimmte Parameteränderung übernommen wird.

Weitere Informationen zum Neustart von Valkey- oder Redis OSS-Knoten finden Sie unter [Knoten neu starten](#)

i Änderungen der Parameter Valkey oder Redis OSS (Cluster Mode Enabled)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
- databases
- 1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
- 2. Löschen Sie den Cluster. Siehe [Löschen von Clustern](#).
- 3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

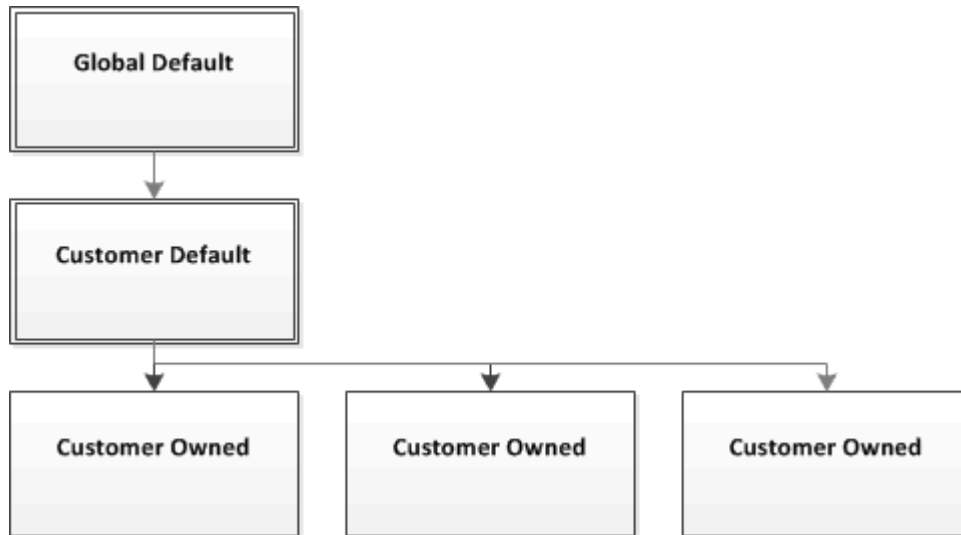
- Sie können Parametergruppen globalen Valkey- und Redis OSS-Datenspeichern zuordnen. Globale Datenspeicher sind eine Sammlung von einem oder mehreren Clustern, die sich über Regionen erstrecken. AWS In diesem Fall wird die Parametergruppe von allen Clustern gemeinsam genutzt, die den globalen Datenspeicher bilden. Alle Änderungen an der Parametergruppe des primären Clusters werden auf alle verbleibenden Cluster im globalen Datenspeicher repliziert. Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#).

Sie können überprüfen, ob eine Parametergruppe Teil eines globalen Datenspeichers ist, indem Sie an folgenden Stellen suchen:

- In der ElastiCache Konsole auf der Seite „Parametergruppen“ das globale Attribut yes/no
- Die yes/no IsGlobal Eigenschaft des [CacheParameterGroup](#)API-Vorgangs

Cache-Parametergruppen werden in ElastiCache

Amazon ElastiCache hat drei Stufen von Cache-Parametergruppen, wie im Folgenden dargestellt.



ElastiCache Amazon-Parametergruppenstufen

Global Default

Die Stammparametergruppe der obersten Ebene für alle ElastiCache Amazon-Kunden in der Region.

Die globale Standard-Cache-Parametergruppe:

- Ist für den Kunden reserviert ElastiCache und steht ihm nicht zur Verfügung.

Customer Default

Eine Kopie der Cache-Parametergruppe „Global Default“, die für die Verwendung durch den Kunden erstellt wird.

Die Cache-Parametergruppe „Customer Default“:

- Ist erstellt und gehört ihm ElastiCache.
- Ist für den Kunden zur Verwendung als Cache-Parametergruppe für alle Cluster verfügbar, die eine Engine-Version ausführen, die von dieser Cache-Parametergruppe unterstützt wird.
- Kann vom Kunden nicht bearbeitet werden.

Customer Owned

Eine Kopie der Cache-Parametergruppe „Customer Default“. Die Cache-Parametergruppe „Customer Owned“ wird immer dann erstellt, wenn der Kunde eine Cache-Parametergruppe anlegt.

Die Cache-Parametergruppe „Customer Owned“:

- Wird vom Kunden erstellt und ist dessen Eigentum.
- Kann beliebigen kompatiblen Clustern des Kunden zugeordnet werden.
- Kann vom Kunden in eine benutzerdefinierte Cache-Parametergruppe abgewandelt werden.

Nicht alle Parameterwerte können geändert werden. Weitere Informationen zu Memcached-Werten finden Sie unter. [Memcached-spezifische Parameter](#) Weitere Informationen zu Valkey- und Redis-OSS-Werten finden Sie unter. [Valkey- und Redis OSS-Parameter](#)

Eine ElastiCache Parametergruppe erstellen

Sie müssen eine neue Parametergruppe erstellen, wenn Sie die Standardwerte für einen oder mehrere Parameterwerte ändern möchten. Sie können eine Parametergruppe mithilfe der ElastiCache Konsole AWS CLI, der oder der ElastiCache API erstellen.

Eine ElastiCache Parametergruppe erstellen (Konsole)

Mit dem folgenden Verfahren wird eine Parametergruppe mithilfe der ElastiCache-Konsole erstellt.

Um eine Parametergruppe mit der ElastiCache Konsole zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.
3. Um eine Parametergruppe zu erstellen, wählen Sie Create Parameter Group.

Der Bildschirm Create Parameter Group wird angezeigt.

4. Wählen Sie aus der Liste Family die Familie der Parametergruppe aus, die als Vorlage für Ihre Parametergruppe dienen soll.

Die Parametergruppenfamilie, wie z. B. memcached1.4 oder redis3.2, definiert die tatsächlichen Parameter in Ihrer Parametergruppe und deren Anfangswerte. Die Familie der Parametergruppe muss mit der Engine und Version des Clusters übereinstimmen.

5. Geben Sie in das Feld Name einen eindeutigen Namen für diese Parametergruppe ein.

Wenn Sie einen Cluster erstellen oder die Parametergruppe eines Clusters ändern, wählen Sie die Parametergruppe nach Namen aus. Daher wird empfohlen, einen informativen und die Familie der Parametergruppe identifizierenden Namen zu wählen.

Namenseinschränkungen für die Parametergruppe:

- Er muss mit einem ASCII-Buchstaben beginnen.
 - Er kann nur ASCII-Buchstaben, Ziffern und Bindestriche („-“) enthalten.
 - Er muss 1 – 255 Zeichen enthalten.
 - Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
 - Er darf nicht mit einem Bindestrich enden.
6. Geben Sie in das Feld Description eine Beschreibung für die Parametergruppe ein.
 7. Um die Parametergruppe zu erstellen, wählen Sie Create.

Um den Vorgang zu beenden, ohne die Parametergruppe zu erstellen, wählen Sie Cancel.

8. Die erstellte Parametergruppe besitzt die Standardwerte der Familie. Zum Ändern der Standardwerte müssen Sie die Parametergruppe abändern. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

Eine Parametergruppe erstellen () ElastiCache AWS CLI

Um eine Parametergruppe mit dem zu erstellen AWS CLI, verwenden Sie den Befehl `create-cache-parameter-group` mit diesen Parametern.

- `--cache-parameter-group-name` Der Name der Parametergruppe.

Namenseinschränkungen für die Parametergruppe:

- Er muss mit einem ASCII-Buchstaben beginnen.
 - Er kann nur ASCII-Buchstaben, Ziffern und Bindestriche („-“) enthalten.
 - Er muss 1 – 255 Zeichen enthalten.
 - Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
 - Er darf nicht mit einem Bindestrich enden.
- `--cache-parameter-group-family` Die Engine und Versionsfamilie der Parametergruppe.
 - `--description` Eine vom Benutzer eingegebene Beschreibung der Parametergruppe.

Example

Im folgenden Beispiel wird anhand der memcached1.4-Familie als Vorlage eine Parametergruppe namens myMem14 erstellt.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name myMem14 \  
  --cache-parameter-group-family memcached1.4 \  
  --description "My first parameter group"
```

Für Windows:

```
aws elasticache create-cache-parameter-group ^  
  --cache-parameter-group-name myMem14 ^  
  --cache-parameter-group-family memcached1.4 ^  
  --description "My first parameter group"
```

Die Ausgabe dieses Befehls sollte in etwa folgendermaßen aussehen.

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "myMem14",  
    "CacheParameterGroupFamily": "memcached1.4",  
    "Description": "My first parameter group"  
  }  
}
```

Example

Im folgenden Beispiel wird anhand der redis2.8-Familie als Vorlage eine Parametergruppe namens myRed28 erstellt.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name myRed28 \  
  --cache-parameter-group-family redis2.8 \  
  --description "My first parameter group"
```

Für Windows:

```
aws elasticache create-cache-parameter-group ^
  --cache-parameter-group-name myRed28 ^
  --cache-parameter-group-family redis2.8 ^
  --description "My first parameter group"
```

Die Ausgabe dieses Befehls sollte in etwa folgendermaßen aussehen.

```
{
  "CacheParameterGroup": {
    "CacheParameterGroupName": "myRed28",
    "CacheParameterGroupFamily": "redis2.8",
    "Description": "My first parameter group"
  }
}
```

Die erstellte Parametergruppe besitzt die Standardwerte der Familie. Zum Ändern der Standardwerte müssen Sie die Parametergruppe abändern. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

Weitere Informationen finden Sie unter [create-cache-parameter-group](#).

Eine ElastiCache Parametergruppe (ElastiCache API) erstellen

Um eine Parametergruppe mithilfe der ElastiCache API zu erstellen, verwenden Sie die `CreateCacheParameterGroup` Aktion mit diesen Parametern.

- `ParameterGroupName` Der Name der Parametergruppe.

Namenseinschränkungen für die Parametergruppe:

- Er muss mit einem ASCII-Buchstaben beginnen.
- Er kann nur ASCII-Buchstaben, Ziffern und Bindestriche („-“) enthalten.
- Er muss 1 – 255 Zeichen enthalten.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.
- `CacheParameterGroupFamily` Die Engine und Versionsfamilie der Parametergruppe. Beispiel, `memcached1.4`.
- `CacheParameterGroupFamily` Die Engine und Versionsfamilie der Parametergruppe. Beispiel, `redis2.8`.

- **Description** Eine vom Benutzer eingegebene Beschreibung der Parametergruppe.

Example

Im folgenden Beispiel wird anhand der memcached1.4-Familie als Vorlage eine Parametergruppe namens myMem14 erstellt.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateCacheParameterGroup
&CacheParameterGroupFamily=memcached1.4
&CacheParameterGroupName=myMem14
&Description=My%20first%20parameter%20group
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Die Antwort auf diese Aktion sollte in etwa folgendermaßen aussehen.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <CreateCacheParameterGroupResult>
    <CacheParameterGroup>
      <CacheParameterGroupName>myMem14</CacheParameterGroupName>
      <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
      <Description>My first parameter group</Description>
    </CacheParameterGroup>
  </CreateCacheParameterGroupResult>
  <ResponseMetadata>
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
  </ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

Example

Im folgenden Beispiel wird anhand der redis2.8-Familie als Vorlage eine Parametergruppe namens myRed28 erstellt.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateCacheParameterGroup
```

```
&CacheParameterGroupFamily=redis2.8
&CacheParameterGroupName=myRed28
&Description=My%20first%20parameter%20group
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Die Antwort auf diese Aktion sollte in etwa folgendermaßen aussehen.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <CreateCacheParameterGroupResult>
    <CacheParameterGroup>
      <CacheParameterGroupName>myRed28</CacheParameterGroupName>
      <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
      <Description>My first parameter group</Description>
    </CacheParameterGroup>
  </CreateCacheParameterGroupResult>
  <ResponseMetadata>
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
  </ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

Die erstellte Parametergruppe besitzt die Standardwerte der Familie. Zum Ändern der Standardwerte müssen Sie die Parametergruppe abändern. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

Weitere Informationen finden Sie unter [CreateCacheParameterGroup](#).

ElastiCache Parametergruppen nach Namen auflisten

Sie können die Parametergruppen mithilfe der ElastiCache Konsole AWS CLI, der oder der ElastiCache API auflisten.

Auflisten von Parametergruppen nach Namen (Konsole)

Mit dem folgenden Verfahren wird eine Liste der Parametergruppen mithilfe der ElastiCache -Konsole angezeigt.

Um Parametergruppen mithilfe der ElastiCache Konsole aufzulisten

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.

ElastiCache Parametergruppen nach Namen auflisten (AWS CLI)

Verwenden Sie den Befehl, um mit dem AWS CLI eine Liste von Parametergruppen zu generierendescribe-cache-parameter-groups. Wenn Sie den Namen einer Parametergruppe angeben, wird nur die betreffende Parametergruppe aufgelistet. Wenn Sie keinen Namen einer Parametergruppe angeben, werden bis zu --max-records Parametergruppen aufgelistet. In beiden Fälle werden Name, Familie und Beschreibung der Parametergruppe aufgelistet.

Example

Der folgenden Beispielcode listet die Parametergruppe myMem14 auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameter-groups \  
  --cache-parameter-group-name myMem14
```

Für Windows:

```
aws elasticache describe-cache-parameter-groups ^  
  --cache-parameter-group-name myMem14
```

Die Ausgabe dieses Befehls sieht folgendermaßen aus und listet Name, Familie und Beschreibung der Parametergruppe auf.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myMem14",
      "CacheParameterGroupFamily": "memcached1.4",
      "Description": "My first parameter group"
    }
  ]
}
```

Example

Der folgenden Beispielcode listet die Parametergruppe myRed28 auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameter-groups \
  --cache-parameter-group-name myRed28
```

Für Windows:

```
aws elasticache describe-cache-parameter-groups ^
  --cache-parameter-group-name myRed28
```

Die Ausgabe dieses Befehls sieht folgendermaßen aus und listet Name, Familie und Beschreibung der Parametergruppe auf.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myRed28",
      "CacheParameterGroupFamily": "redis2.8",
      "Description": "My first parameter group"
    }
  ]
}
```

Example

Der folgende Beispielcode listet die Parametergruppe myRed56 für Parametergruppen auf, die auf der Redis OSS-Engine ab Version 5.0.6 ausgeführt werden. Wenn die Parametergruppe Teil einer

[AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#) ist, ist der in der Ausgabe zurückgegebene `IsGlobal`-Eigenschaftswert `Yes`.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameter-groups \
  --cache-parameter-group-name myRed56
```

Für Windows:

```
aws elasticache describe-cache-parameter-groups ^
  --cache-parameter-group-name myRed56
```

Die Ausgabe dieses Befehls sieht folgendermaßen aus und listet den Namen, die Familie „isGlobal“ und die Beschreibung der Parametergruppe auf.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myRed56",
      "CacheParameterGroupFamily": "redis5.0",
      "Description": "My first parameter group",
      "IsGlobal": "yes"
    }
  ]
}
```

Example

Der folgenden Beispielcode listet bis zu 10 Parametergruppen auf.

```
aws elasticache describe-cache-parameter-groups --max-records 10
```

Die JSON-Ausgabe dieses Befehls sieht ungefähr so aus und listet für jede Parametergruppe den Namen, die Familie, die Beschreibung und im Fall von redis5.6, ob die Parametergruppe Teil eines globalen Datenspeichers (`IsGlobal`) ist, auf.

```
{
  "CacheParameterGroups": [
```

```

{
  "CacheParameterGroupName": "custom-redis32",
  "CacheParameterGroupFamily": "redis3.2",
  "Description": "custom parameter group with reserved-memory > 0"
},
{
  "CacheParameterGroupName": "default.memcached1.4",
  "CacheParameterGroupFamily": "memcached1.4",
  "Description": "Default parameter group for memcached1.4"
},
{
  "CacheParameterGroupName": "default.redis2.6",
  "CacheParameterGroupFamily": "redis2.6",
  "Description": "Default parameter group for redis2.6"
},
{
  "CacheParameterGroupName": "default.redis2.8",
  "CacheParameterGroupFamily": "redis2.8",
  "Description": "Default parameter group for redis2.8"
},
{
  "CacheParameterGroupName": "default.redis3.2",
  "CacheParameterGroupFamily": "redis3.2",
  "Description": "Default parameter group for redis3.2"
},
{
  "CacheParameterGroupName": "default.redis3.2.cluster.on",
  "CacheParameterGroupFamily": "redis3.2",
  "Description": "Customized default parameter group for redis3.2 with
cluster mode on"
},
{
  "CacheParameterGroupName": "default.redis5.6.cluster.on",
  "CacheParameterGroupFamily": "redis5.0",
  "Description": "Customized default parameter group for redis5.6 with
cluster mode on",
  "isGlobal": "yes"
},
]
}

```

Weitere Informationen finden Sie unter [describe-cache-parameter-groups](#).

ElastiCache Parametergruppen nach Namen auflisten (API) ElastiCache

Verwenden Sie die `DescribeCacheParameterGroups` Aktion, um mithilfe der ElastiCache API eine Liste von Parametergruppen zu generieren. Wenn Sie den Namen einer Parametergruppe angeben, wird nur die betreffende Parametergruppe aufgelistet. Wenn Sie keinen Namen einer Parametergruppe angeben, werden bis zu `MaxRecords` Parametergruppen aufgelistet. In beiden Fällen werden Name, Familie und Beschreibung der Parametergruppe aufgelistet.

Example

Der folgenden Beispielcode listet die Parametergruppe `myMem14` auf.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheParameterGroups  
&CacheParameterGroupName=myMem14  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Die Antwort auf diese Aktion sieht folgendermaßen aus und listet Name, Familie und Beschreibung jeder Parametergruppe auf.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParameterGroupsResult>  
    <CacheParameterGroups>  
      <CacheParameterGroup>  
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>  
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>  
        <Description>My custom Memcached 1.4 parameter group</Description>  
      </CacheParameterGroup>  
    </CacheParameterGroups>  
  </DescribeCacheParameterGroupsResult>  
  <ResponseMetadata>  
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>  
  </ResponseMetadata>  
</DescribeCacheParameterGroupsResponse>
```

Example

Der folgenden Beispielcode listet bis zu 10 Parametergruppen auf.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&MaxRecords=10
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Die Antwort dieser Aktion sieht ungefähr so aus und listet für jede Parametergruppe den Namen, die Familie, die Beschreibung und im Fall von redis5.6, ob die Parametergruppe zu einem globalen Datenspeicher (IsGlobal) gehört, auf.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRedis28</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
        <Description>My custom Redis 2.8 parameter group</Description>
      </CacheParameterGroup>
      <CacheParameterGroup>
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
        <Description>My custom Memcached 1.4 parameter group</Description>
      </CacheParameterGroup>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRedis56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
```



```
</DescribeCacheParameterGroupsResponse>
```

Example

Der folgenden Beispielcode listet die Parametergruppe myRed28 auf.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheParameterGroups  
&CacheParameterGroupName=myRed28  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Die Antwort auf diese Aktion sieht folgendermaßen aus und listet den Namen, die Familie und die Beschreibung auf.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParameterGroupsResult>  
    <CacheParameterGroups>  
      <CacheParameterGroup>  
        <CacheParameterGroupName>myRed28</CacheParameterGroupName>  
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>  
        <Description>My custom Redis 2.8 parameter group</Description>  
      </CacheParameterGroup>  
    </CacheParameterGroups>  
  </DescribeCacheParameterGroupsResult>  
  <ResponseMetadata>  
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>  
  </ResponseMetadata>  
</DescribeCacheParameterGroupsResponse>
```

Example

Der folgende Beispiel-Code listet die Parametergruppe myRed56 auf.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheParameterGroups  
&CacheParameterGroupName=myRed56  
&SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Die Antwort dieser Aktion sieht ungefähr so aus und listet den Namen, die Familie, die Beschreibung, und ob die Parametergruppe Teil eines globalen Datenspeichers (IsGlobal) ist, auf.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRed56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom Redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Weitere Informationen finden Sie unter [DescribeCacheParameterGroups](#).

Werte einer ElastiCache Parametergruppe auflisten

Sie können die Parameter und ihre Werte für eine Parametergruppe mithilfe der ElastiCache Konsole, der AWS CLI, oder der ElastiCache API auflisten.

Die Werte einer ElastiCache Parametergruppe auflisten (Konsole)

Das folgende Verfahren zeigt, wie Sie die Parameter und ihre Werte für eine Parametergruppe mithilfe der ElastiCache Konsole auflisten.

So listen Sie die Parameter und deren Werte für eine Parametergruppe mithilfe der ElastiCache-Konsole auf

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.
3. Wählen Sie die Parametergruppe aus, für die Sie die Parameter und Werte auflisten möchten, indem Sie das Kontrollkästchen links neben dem Namen der Parametergruppe aktivieren.

Die Parameter und deren Werte werden unten auf dem Bildschirm aufgelistet. Aufgrund der Anzahl der Parameter müssen Sie möglicherweise nach oben und unten scrollen, um den Parameter zu finden, an dem Sie interessiert sind.

Auflisten der Werte einer Parametergruppe (AWS CLI)

Um die Parameter einer Parametergruppe und ihre Werte mithilfe von AWS CLI aufzulisten, verwenden Sie den Befehl `describe-cache-parameters`.

Example

Der folgende Beispielcode listet alle Memcached-Parameter und ihre Werte für die Parametergruppe `MyMem14` auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameters \
  --cache-parameter-group-name myMem14
```

Für Windows:

```
aws elasticache describe-cache-parameters ^  
  --cache-parameter-group-name myMem14
```

Example

Der folgende Beispielcode listet alle Parameter und deren Werte für die Parametergruppe myRedis28 auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name myRedis28
```

Für Windows:

```
aws elasticache describe-cache-parameters ^  
  --cache-parameter-group-name myRed28
```

Weitere Informationen finden Sie unter [describe-cache-parameters](#).

Auflisten der Werte einer Parametergruppe (API) ElastiCache

Verwenden Sie die DescribeCacheParameters Aktion, um die Parameter einer Parametergruppe und ihre Werte mithilfe der ElastiCache API aufzulisten.

Example

Der folgende Beispielcode listet alle Memcached-Parameter für die Parametergruppe MyMem14 auf.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheParameters  
&CacheParameterGroupName=myMem14  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Die Antwort auf diese Aktion sieht in etwa folgendermaßen aus. Diese Antwort wurde abgeschnitten.

```

<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParametersResult>
    <CacheClusterClassSpecificParameters>
      <CacheNodeTypeSpecificParameter>
        <DataType>integer</DataType>
        <Source>system</Source>
        <IsModifiable>false</IsModifiable>
        <Description>The maximum configurable amount of memory to use to store items,
in megabytes.</Description>
        <CacheNodeTypeSpecificValues>
          <CacheNodeTypeSpecificValue>
            <Value>1000</Value>
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>6000</Value>
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>7100</Value>
            <CacheClusterClass>cache.m1.large</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>1300</Value>
            <CacheClusterClass>cache.m1.small</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
        </CacheNodeTypeSpecificValues>
      </CacheNodeTypeSpecificParameter>
    </CacheClusterClassSpecificParameters>
  </DescribeCacheParametersResult>
  <ResponseMetadata>
    <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParametersResponse>
...output omitted...

```

Example

Der folgende Beispielcode listet alle Parameter für die Parametergruppe myRed28 auf.

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameters

```

```

&CacheParameterGroupName=myRed28
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>

```

Die Antwort auf diese Aktion sieht in etwa folgendermaßen aus. Diese Antwort wurde abgeschnitten.

```

<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParametersResult>
    <CacheClusterClassSpecificParameters>
      <CacheNodeTypeSpecificParameter>
        <DataType>integer</DataType>
        <Source>system</Source>
        <IsModifiable>false</IsModifiable>
        <Description>The maximum configurable amount of memory to use to store items,
in megabytes.</Description>
        <CacheNodeTypeSpecificValues>
          <CacheNodeTypeSpecificValue>
            <Value>1000</Value>
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>6000</Value>
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>7100</Value>
            <CacheClusterClass>cache.m1.large</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>1300</Value>
            <CacheClusterClass>cache.m1.small</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
        </CacheNodeTypeSpecificValues>
      </CacheNodeTypeSpecificParameter>
    </CacheClusterClassSpecificParameters>
  </DescribeCacheParametersResult>
  <ResponseMetadata>
    <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParametersResponse>

```

...output omitted...

```
</ResponseMetadata>  
</DescribeCacheParametersResponse>
```

Weitere Informationen finden Sie unter [DescribeCacheParameters](#).

Änderung einer ElastiCache Parametergruppe

Important

Die Standard-Parametergruppe kann nicht abgeändert werden.

Einige Parameterwerte in einer Parametergruppe können bearbeitet werden. Diese Parameterwerte werden auf die zur Parametergruppe gehörenden Cluster angewendet. Weitere Informationen dazu, wann eine Änderung eines Parameterwerts auf eine Parametergruppe angewendet wird, finden Sie unter [Valkey- und Redis OSS-Parameter](#) und [Memcached-spezifische Parameter](#).

Ändern einer Parametergruppe (Konsole)

Das folgende Verfahren zeigt, wie Sie den Wert des `cluster-enabled` Parameters mithilfe der ElastiCache Konsole ändern können. Dieses Verfahren ist bei der Bearbeitung des Wertes aller Parameter gleich.

So ändern Sie den Wert eines Parameters mithilfe der ElastiCache Konsole

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.
3. Wählen Sie die Parametergruppe aus, die Sie bearbeiten möchten, indem Sie das Kontrollkästchen links neben dem Namen der Parametergruppe aktivieren.

Die Parameter der Parametergruppe werden unten auf dem Bildschirm aufgelistet. Sie können die Liste seitenweise durchlaufen, um alle Parameter zu sehen.

4. Um einen oder mehrere Parameter zu bearbeiten, wählen Sie Edit Parameters.
5. Scrollen Sie auf dem Bildschirm Edit Parameter Group: mit der linken und rechten Pfeiltaste, bis Sie zum Parameter `binding_protocol` gelangen. Geben Sie dann `ascii` in die Spalte Value ein.

6. Wählen Sie Save Changes.
7. Informationen zum Namen des Parameters, den Sie geändert haben, finden Sie unter [Memcached-spezifische Parameter](#). Wenn als Zeitpunkt für die Übernahme der Parameteränderung After restart festgelegt ist, starten Sie jeden Cluster neu, der von dieser Parametergruppe Gebrauch macht. Weitere Informationen finden Sie unter [Neustarten von Clustern](#).
8. Bei Valkey und Redis OSS finden Sie den Namen des Parameters, den Sie geändert haben, unter [Valkey- und Redis OSS-Parameter](#). Wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) haben und Änderungen an den folgenden Parametern vornehmen, müssen Sie die Knoten im Cluster neu starten:
 - activerehashing
 - databases

Weitere Informationen finden Sie unter [Knoten neu starten](#).

 Änderungen der Parameter Valkey oder Redis OSS (Clustermodus aktiviert)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
 - databases
1. Mit Redis OSS können Sie ein manuelles Backup Ihres Clusters erstellen. Siehe [Erstellen manueller Backups](#).
 2. Löschen Sie den Cluster. Siehe [Löschen von Clustern](#).
 3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

Ändern einer Parametergruppe (AWS CLI)

Um den Wert eines Parameters mit dem zu ändern AWS CLI, verwenden Sie den Befehl. `modify-cache-parameter-group`

Example

Bei Memcached finden Sie den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, unter [Memcached-spezifische Parameter](#)

Der folgende Beispiel-Code legt den Wert zweier Parameter fest: `chunk_size` und `chunk_size_growth_fact` auf der Parametergruppe `myMem14`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name myMem14 \  
  --parameter-name-values \  
    ParameterName=chunk_size,ParameterValue=96 \  
    ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

Für Windows:

```
aws elasticache modify-cache-parameter-group ^  
  --cache-parameter-group-name myMem14 ^  
  --parameter-name-values ^  
    ParameterName=chunk_size,ParameterValue=96 ^  
    ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

Die Ausgabe dieses Befehls sieht in etwa folgendermaßen aus.

```
{  
  "CacheParameterGroupName": "myMem14"  
}
```

Example

Bei Valkey und Redis OSS finden Sie den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, unter [Valkey- und Redis OSS-Parameter](#)

Im folgenden Beispielcode wird der Wert von zwei Parametern festgelegt `reserved-memory-percent` und `reserved-memory-percent` für die Parametergruppe `myredis32-on-30`. Wir setzen `reserved-`

memory-percent auf 30 (30 Prozent) und cluster-enabled auf, yes sodass die Parametergruppe mit Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert) (Replikationsgruppen) verwendet werden kann.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name myredis32-on-30 \  
  --parameter-name-values \  
    ParameterName=reserved-memory-percent,ParameterValue=30 \  
    ParameterName=cluster-enabled,ParameterValue=yes
```

Für Windows:

```
aws elasticache modify-cache-parameter-group ^  
  --cache-parameter-group-name myredis32-on-30 ^  
  --parameter-name-values ^  
    ParameterName=reserved-memory-percent,ParameterValue=30 ^  
    ParameterName=cluster-enabled,ParameterValue=yes
```

Die Ausgabe dieses Befehls sieht in etwa folgendermaßen aus.

```
{  
  "CacheParameterGroupName": "my-redis32-on-30"  
}
```

Weitere Informationen finden Sie unter [modify-cache-parameter-group](#).

Den Namen des Parameters, den Sie in einem dieser Themen bearbeitet haben, finden Sie unter [Valkey- und Redis OSS-Parameter](#).

Wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) haben und Änderungen an den folgenden Parametern vornehmen, müssen Sie die Knoten im Cluster neu starten:

- activerehashing
- databases

Weitere Informationen finden Sie unter [Knoten neu starten](#).

Änderungen der Parameter Valkey oder Redis OSS (Clustermodus aktiviert)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
 - databases
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
 2. Löschen Sie den Cluster. Siehe [Löschen von Clustern](#).
 3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

Ändern einer Parametergruppe (API) ElastiCache

Verwenden Sie die `ModifyCacheParameterGroup` Aktion, um die Parameterwerte einer Parametergruppe mithilfe der ElastiCache API zu ändern.

Example

Bei Memcached finden Sie den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, unter [Memcached-spezifische Parameter](#)

Der folgende Beispiel-Code legt den Wert zweier Parameter fest: `chunk_size` und `chunk_size_growth_fact` auf der Parametergruppe `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheParameterGroup  
&CacheParameterGroupName=myMem14  
&ParameterNameValues.member.1.ParameterName=chunk_size  
&ParameterNameValues.member.1.ParameterValue=96  
&ParameterNameValues.member.2.ParameterName=chunk_size_growth_fact  
&ParameterNameValues.member.2.ParameterValue=1.5  
&SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Example

Bei Valkey und Redis OSS finden Sie den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, unter [Valkey- und Redis OSS-Parameter](#)

Im folgenden Beispielcode wird der Wert von zwei Parametern festgelegt reserved-memory-percent und für die Parametergruppe ein Cluster aktiviert. myredis32-on-30 Wir setzen reserved-memory-percent auf 30 (30 Prozent) und cluster-enabled auf, yes sodass die Parametergruppe mit Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert) (Replikationsgruppen) verwendet werden kann.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheParameterGroup
&CacheParameterGroupName=myredis32-on-30
&ParameterNameValues.member.1.ParameterName=reserved-memory-percent
&ParameterNameValues.member.1.ParameterValue=30
&ParameterNameValues.member.2.ParameterName=cluster-enabled
&ParameterNameValues.member.2.ParameterValue=yes
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie unter [ModifyCacheParameterGroup](#).

Wenn Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) haben und Änderungen an den folgenden Parametern vornehmen, müssen Sie die Knoten im Cluster neu starten:

- activerehashing
- databases

Weitere Informationen finden Sie unter [Knoten neu starten](#).

Änderungen der Parameter Valkey oder Redis OSS (Clustermodus aktiviert)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
 - databases
-
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
 2. Löschen Sie den Cluster. Siehe [Löschen eines Clusters in ElastiCache](#).
 3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

Löschen einer ElastiCache Parametergruppe

Sie können eine benutzerdefinierte Parametergruppe mithilfe der ElastiCache Konsole AWS CLI, der oder der ElastiCache API löschen.

Parametergruppen, die Clustern zugeordnet sind, können nicht gelöscht werden. Standard-Parametergruppen können ebenfalls nicht gelöscht werden.

Löschen einer Parametergruppe (Konsole)

Mit dem folgenden Verfahren wird eine Parametergruppe mithilfe der ElastiCache-Konsole gelöscht.

Um eine Parametergruppe mit der ElastiCache Konsole zu löschen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.
3. Wählen Sie die Parametergruppen aus, die Sie löschen möchten, indem Sie das Kontrollkästchen links neben dem Namen der Parametergruppe aktivieren.

Die Schaltfläche Löschen wird aktiv.

4. Wählen Sie Löschen aus.

Der Bestätigungsbildschirm Delete Parameter Groups wird angezeigt.

5. Zum Löschen der Parametergruppen wählen Sie auf dem Bestätigungsbildschirm Delete Parameter Groups die Option Delete aus.

Um die Parametergruppen beizubehalten, wählen Sie Cancel.

Löschen einer Parametergruppe (AWS CLI)

Um eine Parametergruppe mit dem zu löschen AWS CLI, verwenden Sie den Befehl `delete-cache-parameter-group`. Der durch `--cache-parameter-group-name` angegebenen Parametergruppe dürfen eine Cluster zugeordnet sein und es darf sich nicht um eine Standard-Parametergruppe handeln, damit sie gelöscht wird.

Der folgenden Beispielcode löscht die Parametergruppe `myMem14`.

Example

Für Linux, macOS oder Unix:

```
aws elasticache delete-cache-parameter-group \  
  --cache-parameter-group-name myRed28
```

Für Windows:

```
aws elasticache delete-cache-parameter-group ^  
  --cache-parameter-group-name myRed28
```

Weitere Informationen finden Sie unter [delete-cache-parameter-group](#).

Löschen einer Parametergruppe (ElastiCache API)

Verwenden Sie die `DeleteCacheParameterGroup` Aktion, um eine Parametergruppe mithilfe der ElastiCache API zu löschen. Der durch `CacheParameterGroupName` angegebenen Parametergruppe dürfen eine Cluster zugeordnet sein und es darf sich nicht um eine Standard-Parametergruppe handeln, damit sie gelöscht wird.

Example

Bei Memcached löscht der folgende Beispielcode die `MyMem14`-Parametergruppe.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DeleteCacheParameterGroup  
  &CacheParameterGroupName=myMem14  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

Example

Der folgenden Beispielcode löscht die Parametergruppe `myRed28`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DeleteCacheParameterGroup  
  &CacheParameterGroupName=myRed28
```

```
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie unter [DeleteCacheParameterGroup](#).

Motorspezifische Parameter

Valkey und Redis OSS

Die meisten Valkey 8-Parameter sind mit den Redis OSS 7.1-Parametern kompatibel. Die Valkey 7.2-Parameter sind dieselben wie die Redis OSS 7-Parameter.

Wenn Sie keine Parametergruppe für Ihren Valkey- oder Redis OSS-Cluster angeben, wird eine Standardparametergruppe verwendet, die Ihrer Engine-Version entspricht. Sie können die Werte von Parametern in der Standard-Parametergruppe nicht ändern. Sie können jedoch jederzeit eine benutzerdefinierte Parametergruppe erstellen und sie Ihrem Cluster zuordnen, solange die Werte von bedingungsabhängig veränderbaren Parametern in beiden Parametergruppen gleich sind. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

Themen

- [Valkey- und Redis OSS-Parameter](#)
- [Memcached-spezifische Parameter](#)

Valkey- und Redis OSS-Parameter

Themen

- [Änderungen der Valkey 8.2-Parameter](#)
- [Valkey 8.1-Parameteränderungen](#)
- [Valkey 8.0-Parameteränderungen](#)
- [Änderungen der Parameter von Valkey 7.2 und Redis OSS 7](#)
- [Änderungen der Parameter in Redis OSS 6.x](#)
- [Änderungen an den Parametern von Redis OSS 5.0.3](#)
- [Änderungen der Parameter von Redis OSS 5.0.0](#)
- [Änderungen an den Parametern von Redis OSS 4.0.10](#)
- [Änderungen an den Parametern von Redis OSS 3.2.10](#)
- [Änderungen der Parameter in Redis OSS 3.2.6](#)
- [Änderungen der Parameter in Redis OSS 3.2.4](#)
- [Redis OSS 2.8.24 \(erweitert\) hat Parameter hinzugefügt](#)
- [Redis OSS 2.8.23 \(erweitert\) hat Parameter hinzugefügt](#)
- [Redis OSS 2.8.22 \(erweitert\) hat Parameter hinzugefügt](#)
- [Redis OSS 2.8.21 hat Parameter hinzugefügt](#)
- [Redis OSS 2.8.19 hat Parameter hinzugefügt](#)
- [Redis OSS 2.8.6 hat Parameter hinzugefügt](#)
- [Redis OSS 2.6.13-Parameter](#)
- [Redis OSS-Knotentyp-spezifische Parameter](#)

Änderungen der Valkey 8.2-Parameter

Parametergruppenfamilie: valkey8

Note

- Die Parameteränderungen von Valkey 8.2 gelten nicht für Valkey 8.1
- Parametergruppen von Valkey 8.0 und höher sind mit Redis OSS 7.2.4 nicht kompatibel.

- In Valkey 8.2 sind die folgenden Befehle für serverlose Caches nicht verfügbar:,,, und `commandlog commandlog get commandlog help commandlog len commandlog reset`.

Neue Parametergruppen in Valkey 8.2

Name	Details	Description
search-fanout-target-mode (in 8.2 hinzugefügt)	Standard: Client Type: Zeichenkette Veränderbar: Ja Änderungen werden wirksam: sofort	Der search-fanout-target-mode Konfigurationsparameter steuert, wie Suchanfragen auf die Knoten in einer Valkey-Cluster-Umgebung verteilt werden. Diese Einstellung akzeptiert zwei Werte: „Throughput“, der für maximalen Durchsatz optimiert wird, indem Suchabfragen unabhängig vom Clienttyp oder READONLY-Status nach dem Zufallsprinzip auf alle Clusterknoten verteilt werden, und „Client“, der die Verbindungseigenschaften der Clients respektiert, indem nicht schreibgeschützte Clients nur an primäre Knoten, READONLY-Clients nur auf Replikatverbindungen zu Replikatknoten und READONLY-Clients auf primären Verbindungen nach dem Zufallsprinzip über

Name	Details	Description
		alle Knoten weitergeleitet werden. Das Standardverhalten ist der „Client“-Modus, was bedeutet, dass das System die Client-Verbindungstypen und den READONLY-Status bei Routing-Entscheidungen für Abfragen berücksichtigt. Verwenden Sie den Durchsatzmodus für umfangreiche Such-Workloads, bei denen eine maximale Auslastung der Clusterressourcen gewünscht wird, und den Client-Modus, wenn Sie die read/write Trennung beibehalten und READONLY-Verbindungsmuster auf Anwendungsebene respektieren möchten.
search-default-timeout-ms	Standard: 50000 Zulässige Werte: 1 bis 60000 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: sofort	Das standardmäßige Timeout für Valkey-Suchanfragen (in Millisekunden).

Name	Details	Description
search-enable-partial-results	Standard: yes Zulässige Werte: ja, nein Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: sofort	Konfiguriert das Verhalten bei Abfragefehlern für die Valkey-Suche. Wenn diese Option aktiviert ist, geben Suchanfragen unvollständige Ergebnisse zurück, wenn Timeouts bei einem oder mehreren Shards auftreten. Wenn diese Option deaktiviert ist, führt jedes Shard-Timeout dazu, dass die gesamte Suchabfrage fehlschlägt und ein Fehler zurückgegeben wird.

Valkey 8.1-Parameteränderungen

Parametergruppenfamilie: valkey8

Note

- Valkey 8.1-Parameteränderungen gelten nicht für Valkey 8.0
- Parametergruppen von Valkey 8.0 und höher sind mit Redis OSS 7.2.4 nicht kompatibel.
- In Valkey 8.1 sind die folgenden Befehle für serverlose Caches nicht verfügbar:,,, und `commandlog commandlog get commandlog help commandlog len commandlog reset`.

Neue Parametergruppen in Valkey 8.1

Name	Details	Description
commandlog-large-request-max-len (in 8.1 hinzugefügt)	Standard: 1048576	Die maximale Größe in Byte für Anfragen, die von der Valkey Command Log-Funkt

Name	Details	Description
	Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: sofort	ion protokolliert werden können.
commandlog-large-request-max-len (in 8.1 hinzugefügt)	Standard: 128 Zulässige Werte: 0-1024 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: sofort	Die maximale Länge des Valkey-Befehlsprotokolls für Anfragen.
commandlog-reply-larger-than (in 8.1 hinzugefügt)	Standard: 1048576 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: sofort	Die maximale Größe (in Byte) für Antworten, die von der Valkey Command Log-Funktion protokolliert werden können.
commandlog-large-reply-max-len (in 8.1 hinzugefügt)	Standard: 128 Zulässige Werte: 0-1024 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: sofort	Die maximale Länge des Valkey-Befehlsprotokolls für Antworten.

Valkey 8.0-Parameteränderungen

Parametergruppenfamilie: valkey8

Note

Redis OSS 7.2.4 ist nicht mit Parametergruppen von Valkey 8 und höher kompatibel.

Spezifische Parameteränderungen in Valkey 8.0

Name	Details	Description
repl-backlog-size	Standard: 10485760 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: sofort	Die Größe, in Byte, des Backlogs des primären Knotens. Das Backlog dient zur Aufzeichnung von Aktualisierungen an den Daten auf dem primären Knoten. Wenn eine Read Replica eine Verbindung zum Primärknoten herstellt, versucht sie, eine Teilsynchronisierung (psync) durchzuführen, bei der sie Daten aus dem Backlog anwendet, um catch dem Primärknoten Schritt zu halten. Wenn der Psync fehlschlägt, ist eine vollständige Synchronisierung erforderlich. Der Mindestwert für diesen Parameter ist 16384. Hinweis: Ab Redis OSS 2.8.22 gilt dieser Parameter sowohl für den primären Cluster als auch für die Read Replicas.

Name	Details	Description
maxmemory-samples	Standard: 3 Zulässige Werte: 1 bis 64 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: sofort	Bei Berechnungen least-recently-used (LRU) und time-to-live (TTL) stellt dieser Parameter die Stichprobengröße der zu überprüfenden Schlüssel dar. Standardmäßig wählt Redis OSS 3 Schlüssel aus und verwendet den Schlüssel, der zuletzt verwendet wurde.

Neue Parametergruppen in Valkey 8.0

Name	Details	Description
extended-redis-compatibility	Zulässige Werte: ja, nein Standard: yes Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Im erweiterten Redis OSS-Kompatibilitätsmodus gibt Valkey vor, Redis OSS 7.2 zu sein. Aktivieren Sie diese Option nur, wenn Sie Probleme mit Tools oder Clients haben. Auswirkungen auf Kunden: • LOADING- Redis OSS lädt den Datensatz in den Speicher • BUSY- Redis OSS ist ausgelastet • MISCONF- Redis OSS wird auf eine der folgenden Arten konfiguriert:

Name	Details	Description
		- Der HELLO Befehl gibt „server“ => „redis“ und „version“ => „7.2.4“ (unsere Redis OSS-Kompatibilität sversion) zurück. - Das INFO Feld für den Modus heißt „redis_mode“.

Parametergruppen wurden in Valkey 8.0 entfernt

Name	Details	Description
lazyfree-lazy-eviction	Zulässige Werte: ja, nein Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Führt eine asynchrone Löschung bei Bereinigungen durch.
lazyfree-lazy-expire	Zulässige Werte: ja, nein Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Führt eine asynchrone Löschung bei abgelaufenen Schlüsseln durch.

Name	Details	Description
lazyfree-lazy-server-del	Zulässige Werte: ja, nein Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Führt eine asynchrone Löschung bei Befehlen durch, die Werte aktualisieren.
lazyfree-lazy-user-del	Standard: no Typ: Zeichenfolge Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Wenn der Wert auf Ja gesetzt ist, verhält sich der Befehl DEL genauso wie UNLINK.
replica-lazy-flush	Standard: yes Typ: Boolesch Veränderbar: Nein Früherer Name: slave-lazy-flush	Führt einen asynchronen flushDB-Befehl während einer Replikatsynchronisierung durch

Änderungen der Parameter von Valkey 7.2 und Redis OSS 7

Parametergruppenfamilie: valkey7

Die Standardparametergruppen von Valkey 7.2 lauten wie folgt:

- `default.valkey7`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey-Cluster und Replikationsgruppen (Clustermodus deaktiviert).
- `default.valkey7.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey-Cluster und Replikationsgruppen (Clustermodus aktiviert).

Familie der Parametergruppe: `redis7`

Die Standardparametergruppen von Redis OSS 7 lauten wie folgt:

- `default.redis7`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis OSS-Cluster (Clustermodus deaktiviert) und Replikationsgruppen.
- `default.redis7.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis OSS-Cluster (Cluster Mode Enabled) und Replikationsgruppen.

Spezifische Parameteränderungen

Die in Redis OSS 7 hinzugefügten Parameter lauten wie folgt. Valkey 7.2 unterstützt auch diese Parameter.

Name	Details	Description
<code>cluster-allow-pubsubshard-when-down</code>	Gültige Werte: <code>yes</code>, <code>no</code> Standard: <code>yes</code> Typ: Zeichenfolge Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Wenn die Standardeinstellung „Ja“ lautet, können Knoten Pubsub-Shard-Datenverkehr bereitstellen, während sich der Cluster in einem ausgefallenen Zustand befindet, sofern er glauben, dass er die Slots besitzt.
<code>cluster-preferred-endpoint-type</code>	Gültige Werte: <code>ip</code>, <code>tls-dynamic</code> Standard: <code>tls-dynamic</code>	Dieser Wert steuert, welcher Endpunkt für MOVED/ASKING-Anforderungen zurückgegeben wird, sowie das Endpunktfeld für CLUSTER SLOTS und CLUSTER SHARDS. Wenn der Wert auf „ip“ festgelegt ist, gibt

Name	Details	Description
	Typ: Zeichenfolge Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	der Knoten seine IP-Adresse bekannt. Wenn der Wert auf <code>tls-dynamic</code> gesetzt ist, gibt der Knoten einen Hostnamen bekannt, wenn er aktiviert <code>encryption-in-transit</code> ist, und andernfalls eine IP-Adresse.
<code>latency-tracking</code>	Gültige Werte: <code>yes</code>, <code>no</code> Standard: <code>no</code> Typ: Zeichenfolge Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Wenn diese Option auf „Ja“ festgelegt ist, werden die Latenzen pro Befehl protokolliert und die Perzentilverteilung über den Latenzstatistiken-Befehl <code>INFO</code> wird aktiviert. Ebenso werden die kumulativen Latenzverteilungen (Histogramme) über den <code>LATENCY</code>-Befehl exportiert.
<code>hash-max-listpack-entries</code>	Zulässige Werte: <code>0+</code> Standard: <code>512</code> Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Die maximale Anzahl von Hash-Einträgen, damit der Datensatz komprimiert werden kann.

Name	Details	Description
hash-max-listpack-value	Zulässige Werte: 0+ Standard: 64 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Der Schwellenwert der maximalen Anzahl von Hash-Einträgen, damit der Datensatz komprimiert werden kann.
zset-max-listpack-entries	Zulässige Werte: 0+ Standard: 128 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Die maximale Anzahl von Einträgen zu sortierten Sätzen, damit der Datensatz komprimiert werden kann.
zset-max-listpack-value	Zulässige Werte: 0+ Standard: 64 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Der Schwellenwert der maximalen Anzahl von Einträgen zu sortierten Sätzen, damit der Datensatz komprimiert werden kann.

Die in Redis OSS 7 geänderten Parameter lauten wie folgt.

Name	Details	Description
activeresharding	Anpassbar: no. In Redis OSS 7 ist dieser Parameter ausgeblendet und standardmäßig aktiviert. Wenn Sie ihn deaktivieren möchten, müssen Sie einen Support-Fall erstellen .	Veränderbar war ja.

In Redis OSS 7 wurden die folgenden Parameter entfernt.

Name	Details	Description
hash-max-ziplist-entries	Zulässige Werte: 0+ Standard: 512 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Für die Darstellung kleiner Hash-Kodierungen listpack statt ziplist verwenden
hash-max-ziplist-value	Zulässige Werte: 0+ Standard: 64 Typ: Ganzzahl Veränderbar: Ja	Für die Darstellung kleiner Hash-Kodierungen listpack statt ziplist verwenden

Name	Details	Description
	Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	
<code>zset-max-ziplist-entries</code>	Zulässige Werte: 0+ Standard: 128 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Verwenden Sie <code>listpack</code> statt <code>ziplist</code> für die Darstellung kleiner Hash-Kodierungen.
<code>zset-max-ziplist-value</code>	Zulässige Werte: 0+ Standard: 64 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Verwenden Sie <code>listpack</code> statt <code>ziplist</code> für die Darstellung kleiner Hash-Kodierungen.

Name	Details	Description
<code>list-max-ziplist-size</code>	Gültige Werte: Standard: -2 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Die Anzahl der zulässigen Einträge pro internem Listenknoten.

Änderungen der Parameter in Redis OSS 6.x

Familie der Parametergruppe: `redis6.x`

Die Standardparametergruppen von Redis OSS 6.x lauten wie folgt:

- `default.redis6.x`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) und Replikationsgruppen.
- `default.redis6.x.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) und Replikationsgruppen.

Note

In der Redis OSS-Engine-Version 6.2, als die R6GD-Node-Familie für die Verwendung mit eingeführt wurde, wurden nur Noeviction -[Datenzuweisung ElastiCache](#), Volatile-Lru- und Allkeys-LRU-Memory-Richtlinien mit r6gd-Knotentypen unterstützt.

Weitere Informationen erhalten Sie unter [ElastiCache Version 6.2 für Redis OSS \(erweitert\)](#) und [ElastiCache Version 6.0 für Redis OSS \(erweitert\)](#).

Die in Redis OSS 6.x hinzugefügten Parameter lauten wie folgt.

Details	Description	
acl-pubsub-default (added in 6.2)	Gültige Werte: <code>resetchannels</code> , <code>allchannels</code> Standard: <code>allchannels</code> Typ: Zeichenfolge Veränderbar: Ja Die Änderungen werden wirksam: Die vorhandenen Redis OSS-Benutzer, die dem Cluster zugeordnet sind, werden weiterhin über bestehende Berechtigungen verfügen. Aktualisieren Sie entweder die Benutzer oder starten Sie den Cluster neu, um die vorhandenen Redis OSS-Benutzer zu aktualisieren.	Standardberechtigungen für Pub/Sub-Kanäle für ACL-Benutzer, die in diesem Cluster bereitgestellt werden.
cluster-allow-reads-when-down (added in 6.0)	Standard: <code>no</code> Typ: Zeichenfolge Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Wenn diese Option auf Ja gesetzt ist, verarbeitet eine Redis OSS-Replikationsgruppe (Clustermodus aktiviert) weiterhin Lesebefehle, auch wenn ein Knoten nicht in der Lage ist, ein Quorum von Primärdaten zu erreichen. Wenn der Standardwert Nein festgelegt ist, lehnt die Replikationsgruppe alle Befehle ab. Wir empfehlen, diesen Wert auf <code>yes</code> zu setzen, wenn Sie einen Cluster mit weniger als drei Knotengruppen verwenden oder Ihre Anwendung veraltete Lesevorgänge sicher verarbeiten kann.

Details	Description	
<code>tracking-table-max-keys</code> (added in 6.0)	Standard: 1.000.000 Typ: Zahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Um das clientseitige Caching zu unterstützen, unterstützt Redis OSS die Nachverfolgung, welche Clients auf welche Schlüssel zugegriffen haben. Wenn der verfolgte Schlüssel geändert wird, werden Invalidierungsnachrichten an alle Clients gesendet, um ihnen mitzuteilen, dass ihre zwischengespeicherten Werte nicht mehr gültig sind. Mit diesem Wert können Sie die Obergrenze dieser Tabelle angeben. Nachdem dieser Parameterwert überschritten wurde, werden Clients zufällig Invalidierung gesendet. Dieser Wert sollte so eingestellt werden, dass die Speicherauslastung begrenzt wird, während immer noch genügend Schlüssel verfolgt werden. Schlüssel werden auch unter niedrigen Speicherbedingungen ungültig gemacht.
<code>acllog-max-len</code> (added in 6.0)	Standard: 128 Typ: Zahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Dieser Wert entspricht der maximalen Anzahl von Einträgen im ACL-Protokoll.

Details	Description	
<code>active-expire-effort</code> (added in 6.0)	Standard: 1 Typ: Zahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Redis OSS löscht Schlüssel, deren Gültigkeitsdauer überschritten wurde, auf zwei Arten. In einem wird auf einen Schlüssel zugegriffen und festgestellt, dass er abgelaufen ist. In der anderen, ein periodischer Job Proben Schlüssel und bewirkt, dass diejenigen, die ihre Zeit überschritten haben, ablaufen. Dieser Parameter definiert den Aufwand, den Redis OSS aufwendet, um Elemente im periodischen Job ablaufen zu lassen. Der Standardwert von 1 versucht zu vermeiden, dass mehr als 10 Prozent der abgelaufenen Schlüssel noch im Speicher sind. Außerdem wird versucht, mehr als 25 Prozent des gesamten Arbeitsspeichers zu verbrauchen und das System Latenz zu erhöhen. Sie können diesen Wert auf bis zu 10 erhöhen, um den Aufwand für ablaufende Schlüssel zu erhöhen. Der Kompromiss ist eine höhere CPU und eine potenziell höhere Latenz. Wir empfehlen den Wert 1, es sei denn, Sie sehen eine hohe Speicherauslastung und können eine Erhöhung der CPU-Auslastung tolerieren.
<code>lazyfree-lazy-user-del</code> (added in 6.0)	Standard: no Typ: Zeichenfolge Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Wenn der Wert auf Ja gesetzt ist, verhält sich der DEL-Befehl genauso wie UNLINK.

In Redis OSS 6.x wurden die folgenden Parameter entfernt.

Name	Details	Description
lua-replicate-commands	Zulässige Werte: yes/no Standard: yes Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Sofort	Diese Parameter aktivieren immer Lua-Effekteplikation oder Nicht-in-Lua-Skripts.

Änderungen an den Parametern von Redis OSS 5.0.3

Familie der Parametergruppe: redis5.0

Standardparametergruppen für Redis OSS 5.0

- `default.redis5.0`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) und Replikationsgruppen.
- `default.redis5.0.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) und Replikationsgruppen.

In Redis OSS 5.0.3 hinzugefügte Parameter

Name	Details	Description
rename-commands	Standard: keiner Typ: Zeichenfolge Veränderbar: Ja	Eine durch Leerzeichen getrennte Liste umbenannter Redis OSS-Befehle. Im Folgenden finden Sie eine eingeschränkte Liste der verfügbaren Befehle: APPEND AUTH BITCOUNT BITFIELD BITOP BITPOS BLPOP BRPOP BR

Name	Details	Description
	Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	POPLUSH BZPOPMIN BZPOPMAX CLIENT CLUSTER COMMAND DBSIZE DECR DECRBY DEL DISCARD DUMP ECHO EVAL EVALSHA EXEC EXISTS EXPIRE EXPIREAT FLUSHALL FLUSHDB GEOADD GEOHASH GEOPOS GEODIST GEORADIUS GEORADIUSBYMEMBER GET GETBIT GETRANGE GETSET HDEL HEXISTS HGET HGETALL HINCRBY HINCRBYFL OAT HKEYS HLEN HMGET HMSET HSET HSETNX HSTRLEN HVALS INCR INCRBY INCRBYFLOAT INFO KEYS LASTSAVE LINDEX LINSERT LLEN LPOP LPU SH LPUSHX LRANGE LREM LSET LTRIM MEMORY MGET MONITOR MOVE MSET MSETNX MULTI OBJECT PERSIST PEXPIRE PEXPIREAT PFADD PFCOUNT PFMERGE PING PSETEX PSUBSCRIBE PUBSUB PTTL PUBLISH PUNSUBSCRIBE RANDOMKEY READONLY READWRITE RENAME RENAMENX RESTORE ROLE RPOP RPOPLUSH RPUSH RPUSHX SADD SCARD SCRIPT SDIFF SDIFFSTORE SELECT SET SETBIT SETEX SETNX SETRANGE SINTER SINTERSTORE SISMEMBER SLOWLOG SMEMBERS SMOVE SORT SPOP SRANDMEMBER SREM STRLEN SUBSCRIBE UNION UNIONSTORE SWAPDB TIME TOUCH TTL TYPE UNSUBSCRIBE UNLINK UNWATCH WAIT WATCH ZADD ZCARD ZCOUNT ZINCRBY ZINTERSTO RE ZLEXCOUNT ZPOPMAX ZPOPMIN ZRANGE ZRANGEBYLEX ZREVRANGE BYLEX ZRANGEBYSCORE ZRANK ZREM

Name	Details	Description
		ZREMRANGEBYLEX ZREMRANGEBYRANK ZREMRANGEBYSCORE ZREVRANGE ZREVRANGEBYSCORE ZREVRANK ZSCORE ZUNIONSTORE SCAN SSCAN HSCAN ZSCAN XINFO XADD XTRIM XDEL XRA NGE XREVRANGE XLEN XREAD XGROUP XREADGROUP XACK XCLAIM XPENDING GEORADIUS_RO GEORADIUSBYMEMBER_ RO LOLWUT XSETID SUBSTR

Weitere Informationen finden Sie unter [ElastiCache Version 5.0.6 für Redis OSS \(erweitert\)](#).

Änderungen der Parameter von Redis OSS 5.0.0

Familie der Parametergruppe: redis5.0

Standardparametergruppen für Redis OSS 5.0

- `default.redis5.0`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) und Replikationsgruppen.
- `default.redis5.0.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) und Replikationsgruppen.

In Redis OSS 5.0 hinzugefügte Parameter

Name	Details	Description
<code>stream-node-max-bytes</code>	Zulässige Werte: 0+ Standard: 4096 Typ: Ganzzahl Veränderbar: Ja	Die Struktur der Stream-Daten ist eine baumartige Struktur von Knoten, die mehrere Elemente im Innern codieren. Mit dieser Konfiguration legen Sie die maximale Größe (in Bytes) eines einzelnen Knotens in der Baumstruktur fest. Wird die Einstellung „0“ gewählt, ist die Größe des Baumknotens unbegrenzt.

Name	Details	Description
	Änderungen werden wirksam: Sofort	
stream-node-max-entries	Zulässige Werte: 0+ Standard: 100 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Die Struktur der Stream-Daten ist eine baumartige Struktur von Knoten, die mehrere Elemente im Innern codieren. Legen Sie mit dieser Konfiguration die maximale Anzahl der Elemente fest, die ein einzelner Knoten enthalten kann, bevor beim Anhängen neuer Stream-Einträge zu einem neuen Knoten gewechselt wird. Wird die Einstellung „0“ gewählt, ist die Anzahl der Elemente des Baumknotens unbegrenzt.
active-defrag-max-scan-fields	Zulässige Werte: 1 bis 1 000 000 Standard: 1000 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Maximale Anzahl von set/hash/zset/list Feldern, die beim Hauptwörterbuch-Scan verarbeitet werden
lua-replicate-commands	Zulässige Werte: yes/no Standard: yes Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Sofort	Diese Parameter aktivieren immer Lua-Effektreplikation oder Nicht-in-Lua-Skripts.

Name	Details	Description
<code>replica-ignore-maxmemory</code>	Standard: yes Typ: Boolesch Veränderbar: Nein	Dieser Parameter bestimmt, ob das Replikat die Einstellung <code>maxmemory</code> ignoriert, indem Elemente nicht unabhängig von der Primär-Instance bereinigt werden

Redis OSS hat als Reaktion auf das Feedback der Community mehrere Parameter in Engine-Version 5.0 umbenannt. Weitere Informationen finden Sie unter [Was ist neu in Redis OSS 5?](#) . In der folgenden Tabelle sind die neuen Parameternamen und deren Entsprechung in früheren Versionen angegeben.

Parameter wurden in Redis OSS 5.0 umbenannt

Name	Details	Description
<code>replica-lazy-flush</code>	Standard: yes Typ: Boolesch Veränderbar: Nein Früherer Name: <code>slave-lazy-flush</code>	Führt einen asynchronen <code>flushDB</code> -Befehl während einer Replikatsynchronisierung durch
<code>client-output-buffer-limit-replica-hard-limit</code>	Standard: Für Werte siehe Redis OSS-Knotentyp-spezifische Parameter Typ: Ganzzahl Veränderbar: Nein Früherer Name: <code>client-output-buffer-limit - slave-hard-limit</code>	Für Redis OSS Read Replicas: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird der Client getrennt.

Name	Details	Description
<code>client-output-buffer-limit-replica-soft-limit</code>	Standard: Für Werte siehe Redis OSS-Knotentyp-spezifische Parameter Typ: Ganzzahl Veränderbar: Nein Früherer Name: - client-output-buffer-limit slave-soft-limit	Für Redis OSS-Lesereplikate: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird der Client getrennt, aber nur, wenn dieser Zustand weiterhin besteht. <code>client-output-buffer-limit-replica-soft-seconds</code>
<code>client-output-buffer-limit-replica-soft-seconds</code>	Standard: 60 Typ: Ganzzahl Veränderbar: Nein Früherer Name: - client-output-buffer-limit slave-soft-seconds	Für Redis OSS-Lesereplikate: Wenn der Ausgabepuffer eines Clients länger als diese Anzahl von Sekunden auf <code>client-output-buffer-limit-replica-soft-limit</code> Byte verbleibt, wird die Verbindung zum Client getrennt.
<code>replica-allow-chaining</code>	Standard: no Typ: Zeichenfolge Veränderbar: Nein Früherer Name: slave-allow-chaining	Bestimmt, ob eine Read Replica in Redis OSS eigene Read Replicas haben kann.

Name	Details	Description
<code>min-replicas-to-write</code>	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Früherer Name: min-slaves-to-write Änderungen werden wirksam: Sofort	Die Mindestanzahl an Read Replicas, die verfügbar sein müssen, damit der primäre Knoten Schreib Anforderungen von Clients akzeptiert. Wenn die Anzahl der verfügbaren Replikate unter diesen Wert abfällt, dann akzeptiert der primäre Knoten keine Schreib Anforderungen mehr. Wenn entweder dieser Parameter oder 0 min-replicas-max-lag ist, akzeptiert der primäre Knoten immer Schreib Anforderungen, auch wenn keine Replikate verfügbar sind.
<code>min-replicas-max-lag</code>	Standard: 10 Typ: Ganzzahl Veränderbar: Ja Früherer Name: min-slaves-max-lag Änderungen werden wirksam: Sofort	Die Anzahl von Sekunden, innerhalb der der primäre Knoten eine Ping-Anforderung von einer Read Replica empfangen muss. Wenn dieser Zeitraum überschritten wird, ohne dass der primäre Knoten einen Ping erhält, dann wird das Replikat nicht mehr als verfügbar angesehen. Wenn die Anzahl der verfügbaren Replikate unter den Wert fällt min-replicas-to-write, akzeptiert das primäre Replikat zu diesem Zeitpunkt keine Schreibvorgänge mehr. Wenn entweder dieser Parameter oder 0 min-replicas-to-write ist, akzeptiert der primäre Knoten immer Schreib Anforderungen, auch wenn keine Replikate verfügbar sind.

Name	Details	Description
<code>close-on-replica-write</code>	Standard: yes Typ: Boolesch Veränderbar: Ja Früherer Name: close-on-slave-write Änderungen werden wirksam: Sofort	Wenn er aktiviert ist, wird die Verbindung mit Clients, die versuchen, in ein schreibgeschütztes Replikat zu schreiben, unterbrochen.

In Redis OSS 5.0 entfernte Parameter

Name	Details	Description
<code>repl-timeout</code>	Standard: 60 Veränderbar: Nein	Parameter ist in dieser Version nicht verfügbar

Änderungen an den Parametern von Redis OSS 4.0.10

Familie der Parametergruppe: redis4.0

Standardparametergruppen für Redis OSS 4.0.x

- `default.redis4.0`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) und Replikationsgruppen.
- `default.redis4.0.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) und Replikationsgruppen.

Die Parameter wurden in Redis OSS 4.0.10 geändert

Name	Details	Description
maxmemory-policy	Gültige Werte: allkeys-lru , volatile-lru , allkeys-lfu , volatile-lfu , allkeys-random , volatile-random , volatile-ttl , noeviction Standard: volatile-lru Typ: Zeichenfolge Veränderbar: Ja Änderungen treten in Kraft: sofort	maxmemory-policy wurde in Version 2.6.13 hinzugefügt. In Version 4.0.10 wurden zwei neue zulässige Werte hinzugefügt: allkeys-lfu , womit jeder beliebige Schlüssel mittels angenähertem LFU-Wert bereinigt wird, und volatile-lfu , womit unter den Schlüssel n mit einem festgelegten Ablaufdatum mittels angenährtem LFU-Wert bereinigt wird. In Version 6.2, als die R6gd-Knotenfamilie zur Verwendung mit Daten-Tiering eingeführt wurde, werden nur die maxmemory-Richtlinien noeviction , volatile-lru und allkeys-lru mit R6gd-Knotentypen unterstützt.

In Redis OSS 4.0.10 hinzugefügte Parameter

Name	Details	Description
Parameter für asynchrones Löschen		
lazyfree-lazy-eviction	Zulässige Werte: yes/no Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Führt eine asynchrone Löschung bei Bereinigungen durch.

Name	Details	Description
lazyfree-lazy-expire	Zulässige Werte: yes/no Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Führt eine asynchrone Löschung bei abgelaufenen Schlüsseln durch.
lazyfree-lazy-server-del	Zulässige Werte: yes/no Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Führt eine asynchrone Löschung bei Befehlen durch, die Werte aktualisieren.
slave-lazy-flush	Zulässige Werte: Nicht zutreffend Standard: no Typ: Boolesch Veränderbar: Nein Änderungen treten in Kraft: Nicht zutreffend	Führt einen asynchrones flushDB-Befehl während einer Slave-Synchronisierung durch.
LFU-Parameter		

Name	Details	Description
lfu-log-factor	Zulässige Werte: jede Ganzzahl	Legen den Protokollfaktor fest, der die Anzahl der Schlüssel treffer zur Deckung des Schlüsselzählers bestimmt.
	Standard: 10	
	Typ: Ganzzahl	
	Veränderbar: Ja	
	Änderungen treten in Kraft: sofort	
lfu-decay-time	Zulässige Werte: jede Ganzzahl	Zeitdauer in Minuten für die Verringerung des Schlüssel zählers.
	Standard: 1	
	Typ: Ganzzahl	
	Veränderbar: Ja	
	Änderungen treten in Kraft: sofort	
Parameter für aktive Defragmentierung		

Name	Details	Description
activedefrag	Zulässige Werte: yes/no Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Aktiviert die aktive Defragmentierung.  Note In den OSS-Versionen 7.0 und höher von Valkey und Redis AWS kann unabhängig von dieser Einstellung automatisch eine Defragmentierung durchführen, wenn dies betrieblich erforderlich ist.
active-defrag-ignore-bytes	Zulässige Werte: 10485760-104857600 Standard: 104857600 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Mindestmenge an Fragmentierungsresten für das Starten der aktiven Defragmentierung.

Name	Details	Description
active-defrag-threshold-lower	Zulässige Werte: 1–100 Standard: 10 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Mindestprozentsatz der Fragmentierung zum Starten der aktiven Defragmentierung.
active-defrag-threshold-upper	Zulässige Werte: 1–100 Standard: 100 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Maximaler Prozentsatz der Fragmentierung, bei dem der maximale Einsatz aufgewandt wird.
active-defrag-cycle-min	Zulässige Werte: 1–75 Standard: 25 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Minimaler Aufwand für die Defragmentierung als CPU-Prozentsatz.

Name	Details	Description
<code>active-defrag-cycle-max</code>	Zulässige Werte: 1–75 Standard: 75 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Maximaler Aufwand für die Defragmentierung als CPU-Prozentsatz.
Client-Ausgabepuffer-Parameter		
<code>client-query-buffer-limit</code>	Zulässige Werte: 1048576-1073741824 Standard: 1073741824 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Maximale Größe eines einzelnen Client-Abfragepuffers.
<code>proto-max-bulk-len</code>	Zulässige Werte: 1048576-536870912 Standard: 536870912 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Maximale Größe einer einzelnen Elementanforderung.

Änderungen an den Parametern von Redis OSS 3.2.10

Familie der Parametergruppe: redis3.2

ElastiCache für Redis OSS 3.2.10 werden keine zusätzlichen Parameter unterstützt.

Änderungen der Parameter in Redis OSS 3.2.6

Familie der Parametergruppe: redis3.2

Für Redis OSS 3.2.6 werden keine zusätzlichen Parameter unterstützt.

Änderungen der Parameter in Redis OSS 3.2.4

Familie der Parametergruppe: redis3.2

Ab Redis OSS 3.2.4 gibt es zwei Standardparametergruppen.

- `default.redis3.2`— Wenn Sie Redis OSS 3.2.4 ausführen, geben Sie diese oder eine davon abgeleitete Parametergruppe an, wenn Sie eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) erstellen und trotzdem die zusätzlichen Funktionen von Redis OSS 3.2.4 verwenden möchten.
- `default.redis3.2.cluster.on`— Geben Sie diese oder eine davon abgeleitete Parametergruppe an, wenn Sie eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) erstellen möchten.

Themen

- [Neue Parameter für Redis OSS 3.2.4](#)
- [Die Parameter wurden in Redis OSS 3.2.4 geändert \(erweitert\)](#)

Neue Parameter für Redis OSS 3.2.4

Familie der Parametergruppe: redis3.2

Für Redis OSS 3.2.4 werden die folgenden zusätzlichen Parameter unterstützt.

Name	Details	Description
<code>list-max-ziplist-size</code>	Standard: -2 Typ: Ganzzahl Veränderbar: Nein	Listen sind auf besondere Weise codiert, um Platz zu sparen. Die Anzahl der Einträge, die pro internem Listenknoten zulässig sind, können als eine feste maximale Größe oder als maximale Anzahl von Elementen angegeben werden. Verwenden Sie für eine feste maximale Größe -5 bis -1 mit der folgenden Bedeutung: • -5: max. Größe: 64 Kb – wird nicht für normale Arbeitslasten empfohlen • -4: max. Größe: 32 Kb – wird nicht empfohlen • -3: max. Größe: 16 Kb – wird nicht empfohlen • -2: max. Größe: 8 Kb – wird empfohlen • -1: max. Größe: 4 Kb – wird empfohlen • Positive Zahlen bedeuten, bis hin zu genau dieser Anzahl von Elementen pro Listenknoten zu speichern.
<code>list-compress-depth</code>	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Listen können auch komprimiert werden. Die Komprimierungstiefe ist die Anzahl der quicklist ziplist-Knoten ab jeder Seite der Liste, die von der Komprimierung ausgeschlossen werden sollen. Anfang und Ende der Liste sind für schnelle push- und pop-Operationen immer dekomprimiert. Die Einstellungen sind: • 0: Gesamte Komprimierung deaktivieren. •

Name	Details	Description
		1: Komprimierung jeweils einen Knoten nach innen vom Anfang und Ende starten. [head]->node->node->...->node->[tail] Alle Knoten außer [head] und [tail] werden komprimiert. • 2: Komprimierung jeweils zwei Knoten nach innen vom Anfang und Ende starten. [head]->[next]->node->node->...->node->[prev]->[tail] [head], [next], [prev], [tail] werden nicht komprimiert. Alle anderen Knoten werden komprimiert. • usw.

Name	Details	Description
<code>cluster-enabled</code>	Standard: * no/yes Type: Zeichenkette Veränderbar: Nein	Gibt an, ob es sich um eine Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus aktiviert) im Cluster-Modus (ja) oder um eine Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus aktiviert) im Nicht-Cluster-Modus (nein) handelt. Valkey- oder Redis OSS-Replikationsgruppen (Clustermodus aktiviert) im Clustermodus können ihre Daten auf bis zu 500 Knotengruppen partitionieren. * Redis OSS 3.2. x hat zwei Standardparametergruppen. • <code>default.redis3.2</code> – Standardwert no. • <code>default.redis3.2.cluster.on</code> – Standardwert yes. .

Name	Details	Description
<code>cluster-require-full-coverage</code>	Standard: no Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Sofort	Wenn auf <code>gesetztes</code>, akzeptieren Valkey- oder Redis OSS-Knoten (Clustermodus aktiviert) im Clustermodus keine Anfragen mehr, wenn sie feststellen, dass mindestens ein Hash-Slot aufgedeckt ist (kein verfügbarer Knoten bedient ihn). Auf diese Weise ist ein Cluster nicht mehr verfügbar, falls er teilweise ausfällt. Er wird automatisch wieder verfügbar, sobald alle Slots wieder abgedeckt sind. Manchmal ist es jedoch wünschenswert, dass der Teilbereich des Clusters, der funktioniert, für den immer noch abgedeckten Teil des Keyspace weiterhin Abfragen akzeptiert. Stellen Sie dazu die Option <code>cluster-require-full-coverage</code> auf <code>no</code> ein.

Name	Details	Description
hll-spars e-max-byt es	Standard: 3000 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	HyperLogLog Limit für spärliche Repräsentations-Bytes. Das Limit umfasst den 16-Byte-Header. Wenn eine Darstellung HyperLogLog mit geringer Dichte diese Grenze überschreitet, wird sie in eine Darstellung mit hoher Dichte umgewandelt. Ein Wert größer als 16000 wird nicht empfohlen , da ab einem solchen Wert die dichte Repräsentation effizienter ist. Es wird ein Wert von ca. 3 000 empfohlen, um die Vorteile der platzsparenden Codierung zu nutzen, ohne PFADD zu sehr zu verlangsamen. Dies entspricht der Einstellung O(N) bei der Sparse-Codierung. Der Wert kann auf ~10000 angehoben werden, wenn die CPU kein Problem darstellt, sondern der Speicherplatz, und der Datensatz besteht aus vielen Datensätzen HyperLogLogs mit einer Kardinalität zwischen 0 und 15000.

Name	Details	Description
<code>reserved-memory-percent</code>	Standard: 25 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Der reservierte Prozentsatz des Knotenspeichers, der nicht für Daten verwendet werden soll. Standardmäßig wächst der Redis OSS-Datenbedarf, bis er den gesamten Speicher des Knotens verbraucht. In diesem Fall wird die Knotenleistung aufgrund der übermäßigen Speicherauslagerung beeinträchtigt. Indem Sie Speicher reservieren, können Sie einen Teil des verfügbaren Speichers für andere Zwecke als Redis OSS reservieren, um den Umfang des Paging zu reduzieren. Dieser Parameter ist spezifisch für die ElastiCache Redis-OSS-Standarddistribution und nicht Teil dieser. Weitere Informationen erhalten Sie unter <code>reserved-memory</code> und Verwaltung von reserviertem Speicher für Valkey und Redis OSS.

Die Parameter wurden in Redis OSS 3.2.4 geändert (erweitert)

Familie der Parametergruppe: redis3.2

Für Redis OSS 3.2.4 wurden die folgenden Parameter geändert.

Name	Details	Änderungen
<code>activeresharding</code>	Modifizierbar: Ja, wenn die Parametergruppe keinem Cluster zugeordnet ist. Ansonsten nein.	Veränderbar war Nein.

Name	Details	Änderungen
databases	Änderbar: Ja, wenn die Parametergruppe keinem Cluster zugeordnet ist. Ansonsten nein.	Veränderbar war Nein.
appendonly	Standard: deaktiviert Veränderbar: Nein	Wenn Sie ein Upgrade von einer früheren Redis OSS-Version durchführen möchten, müssen Sie das System zuerst ausschalt enappendonly .
appendfsync	Standard: deaktiviert Veränderbar: Nein	Wenn Sie ein Upgrade von einer früheren Redis OSS-Version durchführen möchten, müssen Sie das Gerät zuerst ausschalt enappendfsync .
repl-timeout	Standard: 60 Veränderbar: Nein	Ist nun nicht mehr veränderbar mit einem Standard von 60.
tcp-keepalive	Standard: 300	Standard war 0.
list-max-ziplist-entries		Der Parameter ist nicht mehr verfügbar.
list-max-ziplist-value		Der Parameter ist nicht mehr verfügbar.

Redis OSS 2.8.24 (erweitert) hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.24 werden keine zusätzlichen Parameter unterstützt.

Redis OSS 2.8.23 (erweitert) hat Parameter hinzugefügt

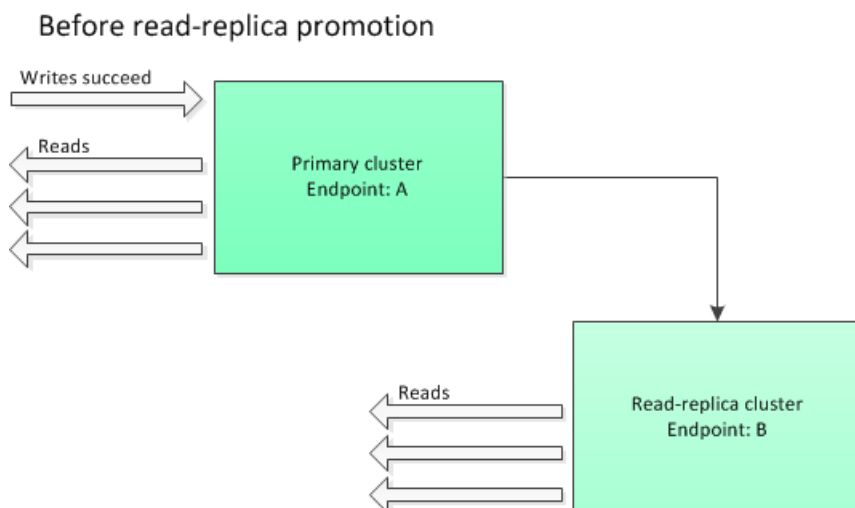
Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.23 wird der folgende zusätzliche Parameter unterstützt.

Name	Details	Description
close-on-slave-write	Standard: yes Typ: Zeichenfolge (yes/no) Veränderbar: Ja Änderungen werden wirksam: Sofort	Wenn er aktiviert ist, wird die Verbindung mit Clients, die versuchen, in ein schreibgeschütztes Replikat zu schreiben, unterbrochen.

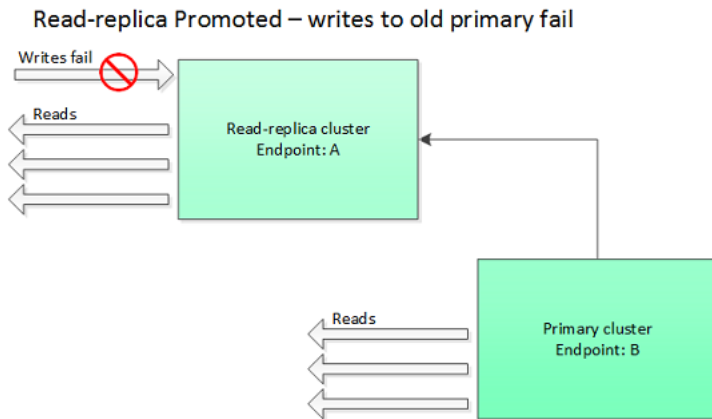
Funktionsweise von close-on-slave-write

Der `close-on-slave-write` Parameter wurde von Amazon eingeführt, ElastiCache um Ihnen mehr Kontrolle darüber zu geben, wie Ihr Cluster reagiert, wenn ein primärer Knoten und ein Read Replica-Knoten aufgrund der Heraufstufung einer Read Replica zum primären Knoten die Rollen wechseln.

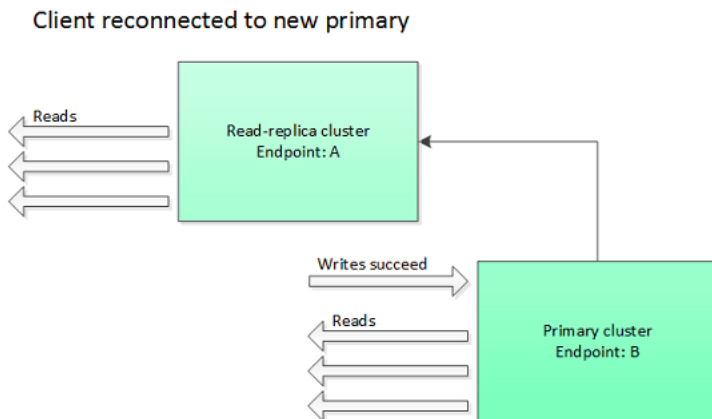


Wenn der Read Replica-Cluster nur aufgrund des Failovers einer Multi-AZ-fähigen Replikation heraufgestuft wird, versucht der Client weiterhin, in den Endpunkt A zu schreiben. Da Endpunkt A nun der Endpunkt für eine Read Replica ist, schlagen diese Schreibvorgänge fehl. Dies ist das Verhalten

für Redis OSS vor der ElastiCache Einführung `close-on-replica-write` und das Verhalten, wenn Sie es deaktivieren `close-on-replica-write`.



Wenn `close-on-replica-write` aktiviert ist, wird jedes Mal bei dem Versuch des Clients, in eine Read Replica zu schreiben, die Client-Verbindung mit dem Cluster unterbrochen. Die Logik Ihrer Anwendung sollte die Unterbrechung erkennen, die DNS-Tabelle überprüfen und wieder eine Verbindung zum primären Endpunkt herstellen, wobei es sich um Endpunkt B handeln würde.



Wann könnten Sie deaktivieren `close-on-replica-write`

Wenn `close-on-replica-write`-Resultate beim Schreiben in fehlerhafte Cluster deaktiviert werden, warum sollte `close-on-replica-write` deaktiviert werden?

Wie zuvor erwähnt, wird jedes Mal, wenn `close-on-replica-write` aktiviert ist, bei dem Versuch des Clients, in eine Read Replica zu schreiben, die Client-Verbindung mit dem Cluster unterbrochen. Das Herstellen einer neuen Verbindung zum Knoten nimmt einige Zeit in Anspruch. Daher beeinträchtigt das Trennen und Wiederherstellen einer Verbindung als Ergebnis einer Schreibsanforderung in das Replica auch die Latenz von Leseanforderungen, die über die gleiche Verbindung bedient werden. Dieser Effekt bleibt bestehen bis eine neue Verbindung hergestellt wird.

Wenn Ihre Anwendung besonders leseintensiv oder latenzempfindlich ist, können Sie die Verbindung Ihrer Clients aufrecht erhalten, um eine Beeinträchtigung der Leseleistung zu vermeiden.

Redis OSS 2.8.22 (erweitert) hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.22 werden keine zusätzlichen Parameter unterstützt.

 Important

- `repl-backlog-size` Gilt ab Redis OSS Version 2.8.22 sowohl für den primären Cluster als auch für Replikatcluster.
- Ab Redis OSS Version 2.8.22 wird der Parameter nicht unterstützt. `repl-timeout` Wenn er geändert wird, ElastiCache wird er mit der Standardeinstellung (60s) überschrieben, wie wir es bei tun. `appendonly`

Die folgenden Parameter werden nicht mehr unterstützt.

- `appendonly`
- `appendfsync`
- `repl-timeout`

Redis OSS 2.8.21 hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.21 werden keine zusätzlichen Parameter unterstützt.

Redis OSS 2.8.19 hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.19 werden keine zusätzlichen Parameter unterstützt.

Redis OSS 2.8.6 hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.6 werden die folgenden zusätzlichen Parameter unterstützt.

Name	Details	Description
<code>min-slaves-max-lag</code>	Standard: 10 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Die Anzahl von Sekunden, innerhalb der der primäre Knoten eine Ping-Anforderung von einer Read Replica empfangen muss. Wenn dieser Zeitraum überschritten wird, ohne dass der primäre Knoten einen Ping erhält, dann wird das Replikat nicht mehr als verfügbar angesehen. Wenn die Anzahl der verfügbaren Replikate unter den Wert fällt <code>min-slaves-to-write</code>, akzeptiert der primäre Server zu diesem Zeitpunkt keine Schreibvorgänge mehr. Wenn entweder dieser Parameter oder 0 <code>min-slaves-to-write</code> ist, akzeptiert der primäre Knoten immer Schreibanforderungen, auch wenn keine Replikate verfügbar sind.
<code>min-slaves-to-write</code>	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Die Mindestanzahl an Read Replicas, die verfügbar sein müssen, damit der primäre Knoten Schreibanforderungen von Clients akzeptiert. Wenn die Anzahl der verfügbaren Replikate unter diesen Wert abfällt, dann akzeptiert der primäre Knoten keine Schreibanforderungen mehr. Wenn entweder dieser Parameter oder 0 <code>min-slaves-max-lag</code> ist, akzeptiert der primäre Knoten

Name	Details	Description
		immer Schreib Anforderungen, auch wenn keine Replike verfügbar sind.

Name	Details	Description
notify-keyspace-events	Standardwert: (eine leere Zeichenfolge) Typ: Zeichenfolge Veränderbar: Ja Änderungen werden wirksam: Sofort	Die Arten von Schlüsselraumereignissen, über die Redis OSS Clients informieren kann. Jede Art von Ereignis wird durch einen einzelnen Buchstaben repräsentiert: • K— Keyspace-Ereignisse, die mit dem Präfix __keyspace@<db> __ • E – Key-event-Ereignisse, die mit dem Präfix __keyevent@<db>__ veröffentlicht werden • g – Generische, unspezifische Befehle, wie z. B. DEL, EXPIRE, RENAME usw. • \$ – Zeichenfolgenbefehle • l — List-Befehle • s — Set-Befehle • h — Hash-Befehle • z — Sortierte Set-Befehle • x — Abgelaufene Ereignisse (Ereignisse, die jedes Mal generiert werden, wenn ein Schlüssel abläuft) •

Name	Details	Description
		e — Ausgeschiedene Ereigniss e (Ereignisse, die generiert werden, wenn ein Schlüssel für maxmemory entfernt wird) • A — Ein Alias für g\$!shzxe Es kann eine beliebige Kombinati on dieser Ereignistypen vorliegen. AKE bedeutet beispielsweise, dass Redis OSS Benachrichtigungen aller Ereignistypen veröffentlichen kann. Verwenden Sie nur die oben aufgelisteten Zeichen. Die versuchte Eingabe anderer Zeichen führt zu Fehlermeldungen. Für diesen Parameter ist standardmäßig eine leere Zeichenfolge eingestellt. Dies bedeutet, dass die Keyspace- Ereignisbenachrichtigung deaktive rt ist.

Name	Details	Description
repl-backlog-size	Standard: 1048576 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Die Größe, in Byte, des Backlogs des primären Knotens. Das Backlog dient zur Aufzeichnung von Aktualisierungen an den Daten auf dem primären Knoten. Wenn sich eine Read Replica mit dem primären Knoten verbindet , versucht sie, eine teilweise Synchronisierung (psync) durchzuführen. Dabei wendet sie die Daten aus dem Backlog an, um den primären Knoten auf den neuesten Stand zu bringen. Wenn psync fehlschlägt, ist eine vollständige Synchronisierung erforderlich. Der Mindestwert für diesen Parameter ist 16384.  Note Ab Redis OSS 2.8.22 gilt dieser Parameter sowohl für den primären Cluster als auch für die Read Replicas.

Name	Details	Description
repl-backlog-ttl	Standard: 3600 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Die Anzahl von Sekunden, für die der primäre Knoten den Backlog-Puffer beibehält. Ab dem Zeitpunkt, an dem die Verbindung des letzte Replikatknotens getrennt wurde, bleiben die Daten im Backlog intakt, bis repl-backlog-ttl abläuft. Wenn das Replikat innerhalb dieses Zeitraums keine Verbindung zum primären Knoten hergestellt hat, dann gibt der primäre Knoten den Backlog-Puffer frei. Wenn das Replikat sich schließlich verbindet, muss es sich einer vollständigen Synchronisierung mit dem primären Knoten unterziehen. Wenn dieser Parameter auf „0“ eingestellt ist, wird der Backlog-Puffer niemals freigegeben.

Name	Details	Description
<code>repl-timeout</code>	Standard: 60 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Stellt den Zeitraum für die Zeitüberschreitung in Sekunden dar für: • Massendatenübertragung während der Synchronisierung, aus der Perspektive des Read Replica • Zeitüberschreitung des primären Knotens aus der Perspektive des Replikats • Zeitüberschreitung des Replikats aus der Perspektive des primären Knotens

Redis OSS 2.6.13-Parameter

Familie der Parametergruppe: redis2.6

Redis OSS 2.6.13 war die erste Version von Redis OSS, die von unterstützt wurde. ElastiCache Die folgende Tabelle zeigt die unterstützten Redis OSS 2.6.13-Parameter. ElastiCache

Name	Details	Description
<code>activerhashing</code>	Standard: yes Typ: Zeichenfolge (yes/no) Veränderbar: Ja Änderungen finden statt: Bei der Erstellung	Bestimmt, ob die aktive Rehashing-Funktion von Redis aktiviert werden soll. Der Hashvorgang für die Haupt-Hash-Tabelle wird zehnmal pro Sekunde erneut durchgeführt. Jeder erneut durchgeführte Hashvorgang verbraucht 1 Millisekunde der CPU-Zeit. Es empfiehlt sich, diesen Wert gering zu halten. Wenn einem Cluster eine neue Parameter

Name	Details	Description
		gruppe zugewiesen wird, muss dieser Wert in der alten und in der neuen Parametergruppe identisch sein.
appendonly	Standard: no Typ: Zeichenfolge Veränderbar: Ja Änderungen werden wirksam: Sofort	Aktiviert oder deaktiviert die Append-Only-File-Funktion (AOF) von Redis. AOF erfasst alle Redis OSS-Befehle, die Daten im Cache ändern, und wird zur Wiederherstellung nach bestimmten Knotenausfällen verwendet. Der Standardwert no bedeutet, dass AOF ausgeschaltet ist. Legen Sie für diesen Parameter yes fest, um AOF zu aktivieren. Weitere Informationen finden Sie unter Minimieren von Ausfällen.  Note Append-Only-Files (AOF) wird für cache.t1.micro- und cache.t2.*-Knoten nicht unterstützt. Für Knoten dieser Art wird der Parameterwert appendonly ignoriert.  Note Für Multi-AZ-Replikationsgruppen ist AOF nicht zulässig.

Name	Details	Description
<code>appendfsync</code>	Standard: <code>everysec</code> Typ: Zeichenfolge Veränderbar: Ja Änderungen werden wirksam: Sofort	Steuert, wenn <code>appendonly</code> auf „Yes“ (Ja) festgelegt, wie oft der AOF-Ausgabepuffer auf den Datenträger geschrieben wird: • <code>no</code> – der Puffer wird nach Bedarf auf die Festplatte geschrieben. • <code>everysec</code> – Der Puffer wird einmal pro Sekunde geleert. Dies ist die Standardinstellung. • <code>always</code> (immer) – der Puffer wird jedes Mal geleert, wenn Daten im Cluster geändert werden. • <code>Appendfsync</code> wird ab der Version 2.8.22 nicht mehr unterstützt.
<code>client-output-buffer-limit-normal-hard-limit</code>	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Bytes erreicht, wird die Verbindung des Clients getrennt. Der Standard ist null (kein festes Limit).
<code>client-output-buffer-limit-normal-soft-limit</code>	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Bytes erreicht, wird die Verbindung des Clients getrennt, aber nur dann, wenn diese Bedingung <code>client-output-buffer-limit-normal-soft-seconds</code> lang andauert. Der Standard ist null (kein weiches Limit).

Name	Details	Description
<code>client-output-buffer-limit-normal-soft-seconds</code>	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Wenn der Ausgabepuffer eines Clients länger als die angegebene Anzahl von Sekunden bei <code>client-output-buffer-limit-normal-soft-limit</code> Bytes verbleibt, wird die Verbindung des Clients getrennt. Der Standard ist null (kein Zeitlimit).
<code>client-output-buffer-limit-pubsub-hard-limit</code>	Standard: 33554432 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Für Redis publish/subscribe OSS-Clients: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird die Verbindung zum Client getrennt.
<code>client-output-buffer-limit-pubsub-soft-limit</code>	Standard: 8388608 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Für Redis publish/subscribe OSS-Clients: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird der Client getrennt, aber nur, wenn dieser Zustand weiterhin besteht. <code>client-output-buffer-limit-pubsub-soft-seconds</code>
<code>client-output-buffer-limit-pubsub-soft-seconds</code>	Standard: 60 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Für Redis publish/subscribe OSS-Clients: Wenn der Ausgabepuffer eines Clients länger als diese Anzahl von Sekunden auf <code>client-output-buffer-limit-pubsub-soft-limit</code> Byte bleibt, wird die Verbindung zum Client getrennt.

Name	Details	Description
<code>client-output-buffer-limit-slave-hard-limit</code>	Standard: Für Werte siehe Redis OSS-Knotentyp-spezifische Parameter Typ: Ganzzahl Veränderbar: Nein	Für Redis OSS Read Replicas: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird die Verbindung zum Client getrennt.
<code>client-output-buffer-limit-slave-soft-limit</code>	Standard: Für Werte siehe Redis OSS-Knotentyp-spezifische Parameter Typ: Ganzzahl Veränderbar: Nein	Für Redis OSS-Lesereplikate: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird der Client getrennt, aber nur, wenn dieser Zustand weiterhin besteht. <code>client-output-buffer-limit-slave-soft-seconds</code>
<code>client-output-buffer-limit-slave-soft-seconds</code>	Standard: 60 Typ: Ganzzahl Veränderbar: Nein	Für Redis OSS-Lesereplikate: Wenn der Ausgabepuffer eines Clients länger als diese Anzahl von Sekunden auf <code>client-output-buffer-limit-slave-soft-limit</code> Byte verbleibt, wird die Verbindung zum Client getrennt.
<code>databases</code>	Standard: 16 Typ: Ganzzahl Veränderbar: Nein Änderungen finden statt: Bei der Erstellung	Die Anzahl der logischen Partitionen, in die die Datenbanken aufgeteilt sind. Es empfiehlt sich, diesen Wert gering zu halten. Es empfiehlt sich, diesen Wert gering zu halten. Wenn einem Cluster eine neue Parametergruppe zugewiesen wird, muss dieser Wert in der alten und in der neuen Parametergruppe identisch sein.

Name	Details	Description
hash-max-ziplist-entries	Standard: 512 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für Hashes verwendete Speichermenge. Hashes mit weniger als der angegebenen Anzahl von Einträgen werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
hash-max-ziplist-value	Standard: 64 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für Hashes verwendete Speichermenge. Hashes mit kleineren Einträgen als die angegebene Anzahl von Bytes werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
list-max-ziplist-entries	Standard: 512 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für Listen verwendete Speichermenge. Listen mit weniger als der angegebenen Anzahl von Einträgen werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
list-max-ziplist-value	Standard: 64 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für Listen verwendete Speichermenge. Listen mit kleineren Einträgen als die angegebene Anzahl von Bytes werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.

Name	Details	Description
<code>lua-time-limit</code>	Standard: 5000 Typ: Ganzzahl Veränderbar: Nein	Die maximale Ausführungszeit für ein Lua-Skript in Millisekunden, bevor ElastiCache Maßnahmen ergriffen werden, um das Skript zu beenden. Wenn sie überschritten <i>lua-time-limit</i> wird, geben alle Redis OSS-Befehle einen Fehler der Form ____-BUSY zurück. Da dieser Status viele wichtige Redis OSS-Operationen stören kann, ElastiCache wird zunächst ein SCRIPT KILL-Befehl ausgegeben. Wenn dies nicht erfolgreich ist, ElastiCache wird Redis OSS zwangsweise neu gestartet.
<code>maxclients</code> – Dieser Wert gilt für alle Instance-Typen mit Ausnahme der explizit angegebenen.	Standard: 65000 Typ: Ganzzahl Veränderbar: Nein t2.medium Standard: 20 000 Typ: Ganzzahl Veränderbar: Nein t2.small Standard: 20 000 Typ: Ganzzahl Veränderbar: Nein t2.micro Standard: 20 000 Typ: Ganzzahl Veränderbar: Nein	Die maximale Anzahl von Clients, die zu jedem beliebigen Zeitpunkt angeschlossen sein können.

Name	Details	Description
	t4g.micro Standard: 20 000	
	Typ: Ganzzahl	
	Veränderbar: Nein	
	t3.medium Standard: 46 000	
	Typ: Ganzzahl	
	Veränderbar: Nein	
	t3.small Standard: 46 000	
	Typ: Ganzzahl	
	Veränderbar: Nein	
	t3.micro Standard: 20 000	
	Typ: Ganzzahl	
	Veränderbar: Nein	
maxmemory-policy	Standard: volatile-lru Typ: Zeichenfolge Veränderbar: Ja Änderungen werden wirksam: Sofort	Die Bereinigungsrichtlinie für Schlüssel, wenn die maximale Speichernutzung erreicht ist. Gültige Werte sind: volatile-lru allkeys-lru volatile-random allkeys-random volatile-ttl noeviction Weitere Informationen finden Sie unter Verwenden von Valkey oder Redis OSS als LRU-Cache.

Name	Details	Description
maxmemory-samples	Standard: 3 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bei Berechnungen least-recently-used (LRU) und time-to-live (TTL) stellt dieser Parameter die Stichprobengröße der zu überprüfenden Schlüssel dar. Standardmäßig wählt Redis OSS 3 Schlüssel aus und verwendet den Schlüssel, der zuletzt verwendet wurde.
reserved-memory	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Der Gesamtspeicher, in Byte, der nicht für Daten verwendet werden soll. Standardmäßig wächst der Redis OSS-Knoten so lange, bis er die des Knotens verbraucht maxmemory (siehe). Redis OSS-Knotentyp-spezifische Parameter In diesem Fall wird die Knotenleistung aufgrund der übermäßigen Speicherauslagerung beeinträchtigt. Indem Sie Speicher reservieren, können Sie einen Teil des verfügbaren Speichers für andere Zwecke als Redis OSS reservieren, um den Umfang des Paging zu reduzieren. Dieser Parameter ist spezifisch für die ElastiCache Redis-OSS-Standarddistribution und nicht Teil dieser. Weitere Informationen erhalten Sie unter reserved-memory-percent und Verwaltung von reserviertem Speicher für Valkey und Redis OSS.

Name	Details	Description
<code>set-max-intset-entries</code>	Standard: 512 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für bestimmte Arten von Sätzen verwendete Speichermenge (Zeichenfolgen, die in Radix 10 Ganzzahlen im Bereich von signierten 64-Bit-Ganzzahlen sind). Solche Sätze mit weniger als der angegebenen Anzahl von Einträgen werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
<code>slave-allow-chaining</code>	Standard: no Typ: Zeichenfolge Veränderbar: Nein	Bestimmt, ob eine Read Replica in Redis OSS eigene Read Replicas haben kann.
<code>slowlog-log-slower-than</code>	Standard: 10000 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Die maximale Ausführungszeit in Mikrosekunden für Befehle, die von der Redis OSS Slow Log-Funktion protokolliert werden sollen.
<code>slowlog-max-len</code>	Standard: 128 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Die maximale Länge des Redis OSS Slow Log.

Name	Details	Description
tcp-keepalive	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Wenn dies auf einen Wert ungleich null (N) eingestellt wird, werden Knoten-Clients alle N Sekunden abgefragt, um sicherzustellen, dass sie noch verbunden sind. Bei der Standardinstellung „0“ finden keine solche Abfragen statt.  Important Einige Aspekte dieses Parameters wurden in Redis OSS Version 3.2.4 geändert. Siehe Die Parameter wurden in Redis OSS 3.2.4 geändert (erweitert).
timeout	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Die Anzahl von Sekunden, die ein Knoten wartet, bevor eine Zeitüberschreitung eintritt. Die Werte sind: • 0 – Die Verbindung eines ungenutzten Client niemals trennen. • 1-19 – Ungültige Werte. • >=20 – Die Anzahl von Sekunden, die ein Knoten wartet, bevor die Verbindung eines ungenutzten Clients getrennt wird.

Name	Details	Description
<code>zset-max-ziplist-entries</code>	Standard: 128 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für sortierte Sätze verwendete Speichermenge. Sortierte Sätze mit weniger als der angegebenen Anzahl von Elementen werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
<code>zset-max-ziplist-value</code>	Standard: 64 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für sortierte Sätze verwendete Speichermenge. Sortierte Sätze mit kleineren Einträgen als die angegebene Anzahl von Bytes werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.

Note

Wenn Sie keine Parametergruppe für Ihren Redis OSS 2.6.13-Cluster angeben, wird eine Standardparametergruppe (`default.redis2.6`) verwendet. Sie können die Werte der Parameter in der Parametergruppe nicht ändern. Sie können jedoch jederzeit immer eine benutzerdefinierte Parametergruppe erstellen und Sie Ihrem Cluster zuweisen.

Redis OSS-Knotentyp-spezifische Parameter

Obwohl die meisten Parameter über einen einzelnen Wert verfügen, ist bei einigen Parametern der jeweilige Wert vom verwendeten Knotentyp abhängig. Die folgende Tabelle zeigt die Standardwerte der Parameter `maxmemory`, `client-output-buffer-limit-slave-hard-limit` und `client-output-buffer-limit-slave-soft-limit` für jeden Knotentyp. Der Wert `maxmemory` ist die maximale Anzahl von Bytes, die für Ihre Verwendung, für Daten oder für andere Zwecke auf dem Knoten verfügbar sind. Weitere Informationen finden Sie unter [Verfügbarer Speicher](#).

Note

Der Parameter `maxmemory` kann nicht geändert werden.

Knotentyp	Maxmemory	C lient-output-buffe r-limit - slave-hard- limit	C lient-output-buffe r-limit - slave-soft- limit
cache.t1.micro	142606336	14260633	14260633
cache.t2.micro	581959680	58195968	58195968
cache.t2.small	1665138688	166513868	166513868
cache.t2.medium	3461349376	346134937	346134937
cache.t3.micro	536870912	53687091	53687091
cache.t3.small	1471026299	147102629	147102629
cache.t3.medium	3317862236	331786223	331786223
cache.t4g.micro	536870912	53687091	53687091
cache.t4g.small	1471026299	147102629	147102629
cache.t4g.medium	3317862236	331786223	331786223
cache.m1.small	943718400	94371840	94371840
cache.m1.medium	3093299200	309329920	309329920
cache.m1.large	7025459200	702545920	702545920
cache.m1.xlarge	14889779200	1488977920	1488977920
cache.m2.xlarge	17091788800	1709178880	1709178880
cache.m2.2xlarge	35022438400	3502243840	3502243840

Knotentyp	Maxmemory	C lient-output-buffe r-limit - slave-hard- limit	C lient-output-buffe r-limit - slave-soft- limit
cache.m2.4xlarge	70883737600	7088373760	7088373760
cache.m3.medium	2988441600	309329920	309329920
cache.m3.large	6501171200	650117120	650117120
cache.m3.xlarge	14260633600	1426063360	1426063360
cache.m3.2xlarge	29989273600	2998927360	2998927360
cache.m4.large	6892593152	689259315	689259315
cache.m4.xlarge	15328501760	1532850176	1532850176
cache.m4.2xlarge	31889126359	3188912636	3188912636
cache.m4.4xlarge	65257290629	6525729063	6525729063
cache.m4.10xlarge	166047614239	16604761424	16604761424
cache.m5.large	6854542746	685454275	685454275
cache.m5.xlarge	13891921715	1389192172	1389192172
cache.m5.2xlarge	27966669210	2796666921	2796666921
cache.m5.4xlarge	56116178125	5611617812	5611617812
cache.m5.12xlarge	168715971994	16871597199	16871597199
cache.m5.24xlarge	337500562842	33750056284	33750056284
cache.m6g.large	6854542746	685454275	685454275
cache.m6g.xlarge	13891921715	1389192172	1389192172
cache.m6g.2xlarge	27966669210	2796666921	2796666921

Knotentyp	Maxmemory	C lient-output-buffe r-limit - slave-hard- limit	C lient-output-buffe r-limit - slave-soft- limit
cache.m6g.4xlarge	56116178125	5611617812	5611617812
cache.m6g.8xlarge	111325552312	11132555231	11132555231
cache.m6g.12xlarge	168715971994	16871597199	16871597199
cache.m6g.16xlarge	225000375228	22500037523	22500037523
cache.c1.xlarge	6501171200	650117120	650117120
cache.r3.large	14470348800	1468006400	1468006400
cache.r3.xlarge	30513561600	3040870400	3040870400
cache.r3.2xlarge	62495129600	6081740800	6081740800
cache.r3.4xlarge	126458265600	12268339200	12268339200
cache.r3.8xlarge	254384537600	24536678400	24536678400
cache.r4.large	13201781556	1320178155	1320178155
cache.r4.xlarge	26898228839	2689822883	2689822883
cache.r4.2xlarge	54197537997	5419753799	5419753799
cache.r4.4xlarge	108858546586	10885854658	10885854658
cache.r4.8xlarge	218255432090	21825543209	21825543209
cache.r4.16xlarge	437021573120	43702157312	43702157312
cache.r5.large	14037181030	1403718103	1403718103
cache.r5.xlarge	28261849702	2826184970	2826184970
cache.r5.2xlarge	56711183565	5671118356	5671118356

Knotentyp	Maxmemory	C client-output-buffer-limit - slave-hard-limit	C client-output-buffer-limit - slave-soft-limit
cache.r5.4xlarge	113609865216	11360986522	11360986522
cache.r5.12xlarge	341206346547	34120634655	34120634655
cache.r5.24xlarge	682485973811	68248597381	68248597381
cache.r6g.large	14037181030	1403718103	1403718103
cache.r6g.xlarge	28261849702	2826184970	2826184970
cache.r6g.2xlarge	56711183565	5671118356	5671118356
cache.r6g.4xlarge	113609865216	11360986522	11360986522
cache.r6g.8xlarge	225000375228	22500037523	22500037523
cache.r6g.12xlarge	341206346547	34120634655	34120634655
cache.r6g.16xlarge	450000750456	45000075046	45000075046
cache.r6gd.xlarge	28261849702	2826184970	2826184970
cache.r6gd.2xlarge	56711183565	5671118356	5671118356
cache.r6gd.4xlarge	113609865216	11360986522	11360986522
cache.r6gd.8xlarge	225000375228	22500037523	22500037523
cache.r6gd.12xlarge	341206346547	34120634655	34120634655
cache.r6gd.16xlarge	450000750456	45000075046	45000075046
cache.r7g.large	14037181030	1403718103	1403718103
cache.r7g.xlarge	28261849702	2826184970	2826184970
cache.r7g.2xlarge	56711183565	5671118356	5671118356

Knotentyp	Maxmemory	C client-output-buffer-limit - slave-hard-limit	C client-output-buffer-limit - slave-soft-limit
cache.r7g.4xlarge	113609865216	11360986522	11360986522
cache.r7g.8xlarge	225000375228	22500037523	22500037523
cache.r7g.12xlarge	341206346547	34120634655	34120634655
cache.r7g.16xlarge	450000750456	45000075046	45000075046
cache.m7g.large	6854542746	685454275	685454275
cache.m7g.xlarge	13891921715	1389192172	1389192172
cache.m7g.2xlarge	27966669210	2796666921	2796666921
cache.m7g.4xlarge	56116178125	5611617812	5611617812
cache.m7g.8xlarge	111325552312	11132555231	11132555231
cache.m7g.12xlarge	168715971994	16871597199	16871597199
cache.m7g.16xlarge	225000375228	22500037523	22500037523
cache.c7gn.large	3317862236	1403718103	1403718103
cache.c7gn.xlarge	6854542746	2826184970	2826184970
cache.c7gn.2xlarge	13891921715	5671118356	5671118356
cache.c7gn.4xlarge	27966669210	11360986522	11360986522
cache.c7gn.8xlarge	56116178125	22500037523	22500037523
cache.c7gn.12xlarge	84357985997	34120634655	34120634655
cache.c7gn.16xlarge	113609865216	45000075046	45000075046

 Note

Alle Instance-Typen der aktuellen Generation werden standardmäßig in Amazon Virtual Private Cloud VPC erstellt.

T1-Instances unterstützen Multi-AZ nicht.

T1- und T2-Instances unterstützen Redis OSS AOF nicht.

Redis OSS-Konfigurationsvariablen `appendonly` und `appendfsync` werden in Redis OSS Version 2.8.22 und höher nicht unterstützt.

Memcached-spezifische Parameter

Memcached

Wenn Sie für Ihren Memcached-Cluster keine Parametergruppe angeben, wird eine für Ihre Engine-Version geeignete Standard-Parametergruppe verwendet. Sie können die Werte von Parametern in einer Standard-Parametergruppe nicht ändern. Sie können jedoch jederzeit eine benutzerdefinierte Parametergruppe erstellen und sie Ihrem Cluster zuordnen. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

Themen

- [Änderungen an Memcached 1.6.17](#)
- [In Memcached 1.6.6 hinzugefügte Parameter](#)
- [Parameteränderungen für Memcached 1.5.10](#)
- [In Memcached 1.4.34 hinzugefügte Parameter](#)
- [In Memcached 1.4.33 hinzugefügte Parameter](#)
- [In Memcached 1.4.24 hinzugefügte Parameter](#)
- [In Memcached 1.4.14 hinzugefügte Parameter](#)
- [In Memcached 1.4.5 unterstützte Parameter](#)
- [Overhead von Memcached-Verbindungen](#)
- [Spezifische Parameter des Memcached-Knotentyps](#)

Änderungen an Memcached 1.6.17

Ab Memcached 1.6.17 unterstützen wir die folgenden administrativen Befehle nicht mehr: `lru_crawler`, `lru` und `slabs`. Mit diesen Änderungen wird es Ihnen zur Laufzeit

nicht möglich sein, `enable/disable lru_crawler` Befehle zu verwenden. Bitte `enable/disable lru_crawler` ändern Sie Ihre benutzerdefinierte Parametergruppe.

In Memcached 1.6.6 hinzugefügte Parameter

Für Memcached 1.6.6 werden keine zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.6`

Parameteränderungen für Memcached 1.5.10

Für Memcached 1.5.10 werden die folgenden zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.5`

Name	Details	Description
<code>no_modern</code>	Standard: 1 Typ: Boolesch Veränderbar: Ja Zulässige Werte: 0, 1 Änderungen werden wirksam: Beim Start	Ein Alias für die Deaktivierung von <code>maxconns_fast</code> Befehlen <code>slab_reassign</code> <code>lru_maintainer_thread</code> <code>lru_segmented</code> „ und. Wenn Sie Memcached 1.5 und höher verwenden, wird der <code>hash_algorithm no_modern</code> auch auf gesetzt. <code>jenkins</code> Bei Verwendung von Memcached 1.5.10 wird dies außerdem durch den Parameter gesteuert <code>.inline_ascii_reponse_parallelly</code> Das heißt, wenn deaktiviert ist, <code>no_modern</code> ist es auch deaktiviert. <code>inline_ascii_reponse</code> Ab Memcached Engine 1.5.16 gilt der <code>inline_as</code>

Name	Details	Description
		<code>cii_response</code> Parameter nicht mehr, sodass die Aktivierung oder Deaktivierung keine <code>no_modern</code> Auswirkung auf hat. <code>inline_ascii_reponse</code> Wenn er deaktiviert <code>no_modern</code> ist, dann werden <code>slab_reassign</code>, <code>lru_maintainer_thread</code>, <code>lru_segmented</code>, und <code>maxconns_fast</code> aktiviert. Da es sich bei den <code>hash_algorithm</code> Parametern <code>slab_automove</code> und nicht um <code>SWITCH</code>-Parameter handelt, basiert ihre Einstellung auf den Konfigurationen in der Parametergruppe. Wenn Sie den Parameter deaktivieren <code>no_modern</code> und zu ihm zurückkehren möchten <code>modern</code>, müssen Sie eine benutzerdefinierte Parametergruppe konfigurieren, um diesen Parameter zu deaktivieren, und dann einen Neustart durchführen, damit diese Änderungen wirksam werden.  Note Der Standardkonfigurationswert für diesen Parameter wurde am 20. August 2021 von 0 in

Name	Details	Description
		1 geändert. Der aktualisierte Standardwert wird nach dem 20. August 2021 automatisch von neuen ElastiCache Benutzern für jede Region übernommen. Bestehende ElastiCache Benutzer in den Regionen vor dem 20. August 2021 müssen ihre benutzerdefinierten Parametergruppen manuell ändern, um diese neue Änderung zu übernehmen.
<code>inline_ascii_resp</code>	Standard: 0 Typ: Boolesch Veränderbar: Ja Zulässige Werte: 0, 1 Änderungen werden wirksam: Beim Start	Speichert Zahlen der VALUE-Antwort innerhalb eines Elements mit bis zu 24 Bytes. Geringe Verlangsamung für ASCII-Sätze <code>get</code> und <code>faster</code> .

Für Memcached 1.5.10 werden die folgenden Parameter entfernt.

Name	Details	Description
<code>expirezero_does_not_evict</code>	Standard: 0 Typ: Boolesch Veränderbar: Ja	Wird in dieser Version nicht länger unterstützt

Name	Details	Description
	Zulässige Werte: 0, 1 Änderungen werden wirksam: Beim Start	
<code>modern</code>	Standard: 1 Typ: Boolesch Anpassbar: Ja (erfordert Neustart, wenn auf <code>no_modern</code> gesetzt) Zulässige Werte: 0, 1 Änderungen werden wirksam: Beim Start	Wird in dieser Version nicht länger unterstützt Ab dieser Version ist <code>no-modern</code> standardmäßig bei jedem Start oder Neustart aktiviert.

In Memcached 1.4.34 hinzugefügte Parameter

Für Memcached 1.4.34 werden keine zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.4`

In Memcached 1.4.33 hinzugefügte Parameter

Für Memcached 1.4.33 werden die folgenden zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.4`

Name	Details	Description
<code>modern</code>	Standard: aktiviert Typ: Boolesch Veränderbar: Ja	Ein Alias für mehrere Funktionen. Das Aktivieren von <code>modern</code> ist gleichbedeutend mit dem Einschalten der folgenden Befehle und der Verwendung eines murmur3-

Name	Details	Description
	Änderungen werden wirksam: Beim Start	Hash-Algorithmus: <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , <code>maxconns_fast</code> und <code>hash_algorithm=murmur3</code> .
<code>watch</code>	Standard: aktiviert Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Sofort Protokolle werden möglicherweise verworfen, wenn der Benutzer die Grenzwerte für <code>watcher_logbuf_size</code> und <code>worker_logbuf_size</code> erreicht.	Protokollabrufe, Bereinigungen oder Mutationen. Wenn der Benutzer z. B. <code>watch</code> aktiviert, kann er Protokolle sehen, wenn <code>get</code> , <code>set</code> , <code>delete</code> oder <code>update</code> auftreten.
<code>idle_timeout</code>	Standard: 0 (deaktiviert) Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Beim Start	Die minimale Anzahl von Sekunden, die sich ein Client im Leerlauf befinden kann, bevor er zum Schließen aufgefordert wird. Wertebereich: 0 bis 86400.

Name	Details	Description
<code>track_sizes</code>	Standard: deaktiviert Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Beim Start	Zeigt die Größen an, die jede Slab-Gruppe verbraucht hat. Wenn <code>track_sizes</code> aktiviert ist, können Sie <code>stats sizes</code> ausführen, ohne <code>stats sizes_enable</code> ausführen zu müssen.
<code>watcher_logbuf_size</code>	Standard: 256 (KB) Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Beim Start	Der Befehl <code>watch</code> schaltet die Stream-Protokollierung für Memcached ein. <code>watch</code> kann jedoch Protokolle verwerfen, wenn die Rate der Bereinigungen, Mutationen oder Abrufe so hoch ist, dass der Protokollierungspuffer voll wird. In solchen Fällen kann der Benutzer die Puffergröße erhöhen, um die Gefahr von Protokollverlusten zu verringern.
<code>worker_logbuf_size</code>	Standard: 64 (KB) Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Beim Start	Der Befehl <code>watch</code> schaltet die Stream-Protokollierung für Memcached ein. <code>watch</code> kann jedoch Protokolle verwerfen, wenn die Rate der Bereinigungen, Mutationen oder Abrufe so hoch ist, dass der Protokollierungspuffer voll wird. In solchen Fällen kann der Benutzer die Puffergröße erhöhen, um die Gefahr von Protokollverlusten zu verringern.

Name	Details	Description
<code>slab_chunk_max</code>	Standard: 524288 (Byte) Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Beim Start	Gibt die maximale Größe eines Slabs an. Bei Festlegen einer kleineren Slab-Größe wird der Speicher effizienter genutzt. Elemente, die größer als <code>slab_chunk_max</code> sind, werden über mehrere Slabs hinweg aufgeteilt.
<code>lru_crawler metadump [all 1 2 3]</code>	Standard: deaktiviert Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Sofort	Wenn <code>lru_crawler</code> aktiviert ist, sichert dieser Befehl ein Abbild aller Schlüssel. <code>all 1 2 3</code> – alle Slabs oder geben Sie eine bestimmte Slab-Nummer an

In Memcached 1.4.24 hinzugefügte Parameter

Für Memcached 1.4.24 werden die folgenden zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.4`

Name	Details	Description
<code>disable_flush_all</code>	Standard: 0 (deaktiviert) Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Beim Start	Hinzufügen von Parameter (-F), um <code>flush_all</code> zu deaktivieren. Nützlich, wenn Sie niemals eine volle Leerung für Produktions-Instances ausführen möchten.

Name	Details	Description
		Werte: 0, 1 (Benutzer kann einen <code>flush_all</code> ausführen, wenn der Wert 0 ist).
hash_algorithm	Standard: jenkins Typ: Zeichenfolge Veränderbar: Ja Änderungen werden wirksam: Beim Start	Der zu verwendende Hash-Algorithmus. Zulässige Werte: murmur3 und jenkins.

Name	Details	Description
<code>lru_crawler</code>	Standard: 0 (deaktiviert) Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart  Note Sie können <code>lru_crawler</code> vorübergehend zur Laufzeit über die Befehlszeile aktivieren. Weitere Informationen finden Sie in der Spalte „Description“.	Bereinigt Slab-Klassen abgelaufener Elemente. Dies ist ein Vorgang mit geringen Auswirkungen, der im Hintergrund ausgeführt wird. Er erfordert zurzeit, dass das Durchsuchen mit einem manuellen Befehl initiiert wird. Um ihn auf bestimmte Zeit zu aktivieren, führen Sie <code>lru_crawler enable</code> über die Befehlszeile aus. <code>lru_crawler 1,3,5</code> durchläuft Slab-Klassen 1, 3 und 5 auf der Suche nach abgelaufenen Elementen, die zur Freelist hinzugefügt werden sollen. Werte: 0,1  Note Wenn <code>lru_crawler</code> über die Befehlszeile aktiviert wird, wird der Crawler so lange aktiviert, bis er über die Befehlszeile oder beim nächsten Neustart deaktiviert wird. Um ihn permanent zu aktivieren, müssen Sie den Parameterwert ändern. Weitere Informationen finden Sie unter

Name	Details	Description
		Änderung einer ElastiCache Parametergruppe.
<code>lru_maintainer</code>	Standard: 0 (deaktiviert) Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Beim Start	Ein Hintergrund-Thread, der Elemente zwischen den Objekten hin und her mischt, LRUs sobald die Kapazitäten erreicht sind. Werte: 0, 1.
<code>expirezero_does_not_evict</code>	Standard: 0 (deaktiviert) Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Beim Start	Bewirkt bei Verwendung zusammen mit <code>lru_maintainer</code> , dass Elemente mit der Ablaufzeit 0 nicht bereinigt werden können.  Warning Dadurch kann der Speicher für andere Elemente, die bereinigt werden können, knapp werden. Kann so eingestellt werden, dass <code>lru_maintainer</code> ignoriert wird.

In Memcached 1.4.14 hinzugefügte Parameter

Für Memcached 1.4.14 werden die folgenden zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: memcached1.4

In Memcached 1.4.14 hinzugefügte Parameter

Name	Description
config_max	Die maximale Anzahl von ElastiCache Konfigurationseinträgen.
config_size_max	Die maximale Größe der Konfigurationseinträge in Byte.
hashpower_init	Die Anfangsgröße der ElastiCache Hashtabelle, ausgedrückt als Zweierpotenz. Der Standard ist 16 (2^{16}), oder 65536 Schlüssel.
maxconns_fast	Ändert, auf welche Art neue Verbindungsanforderungen gehandhabt werden, wenn das maximale Verbindungslimit erreicht ist. Wenn dieser Parameter auf 0 (Null) eingestellt ist, werden neue Verbindungen zur Rückstandwarteschlange hinzugefügt und gewartet, bis andere Verbindungen geschlossen werden. Wenn der Parameter auf 1

Name	Description
	gesetzt ist, wird ein Fehler an den Client ElastiCache gesendet und die Verbindung sofort geschlossen.
slab_automove	Korrigiert den Slab-AutoMove-Algorithmus: Wenn dieser Parameter auf 0 (Null) eingestellt ist, ist der AutoMove-Algorithmus deaktiviert. Wenn er auf 1 eingestellt ist, geht ElastiCache beim automatischen Verschieben von Slabs langsam und zurückhaltend vor. Wenn der Wert auf 2 gesetzt ist, werden Platten bei jeder Räumung ElastiCache aggressiv bewegt. (Dieser Modus wird ausschließlich zu Testzwecken empfohlen.)

Name	Description
slab_reassign	Aktiviert oder deaktiviert die Slab-Neuzuweisung. Wenn dieser Parameter auf 1 eingestellt ist, können Sie mit dem Befehl „slabs reassign“ Speicher manuell neu zuweisen.

In Memcached 1.4.5 unterstützte Parameter

Familie der Parametergruppe: memcached1.4

Für Memcached 1.4.5 werden die folgenden Parameter unterstützt.

In Memcached 1.4.5 hinzugefügte Parameter

Name	Details	Description
backlog_queue_limit	Standard: 1024 Typ: Ganzzahl Veränderbar: Nein	Das Limit der Rückstandwarteschlange (Backlog Queue).
binding_protocol	Standard: auto	Das Bindungsprotokoll.

Name	Details	Description
	Typ: Zeichenfolge Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Zulässige Werte sind: <code>ascii</code> und <code>auto</code> . Eine Anleitung zum Ändern des Wertes von <code>binding_protocol</code> finden Sie unter Änderung einer ElastiCache Parametergruppe .
<code>cas_disabled</code>	Standard: 0(false) Typ: Boolescher Wert Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Bei dem Wert 1 (true) werden die Operationen <code>check</code> und <code>set</code> (CAS) deaktiviert und gespeicherte Elemente verbrauchen 8 Bytes weniger als bei aktiviertem CAS.
<code>chunk_size</code>	Standard: 48 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Die Mindestmenge, in Byte, von Speicher, die für den Schlüssel, Wert und die Flags des kleinsten Elements zuzuordnen sind.
<code>chunk_size_growth_factor</code>	Standard: 1.25 Typ: Gleitkommazahl Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Der Wachstumsfaktor, der die Größe aller nachfolgenden Memcached-Datenblöcke steuert. Jeder Datenblock wird <code>chunk_size_growth_factor</code> Mal größer als der vorherige Datenblock sein.

Name	Details	Description
<code>error_on_memory_exhausted</code>	Standard: 0(false) Typ: Boolescher Wert Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Bei dem Wert 1 (true) gibt Memcached einen Fehler zurück, wenn kein Speicher zum Speichern der Elemente mehr verfügbar ist, anstatt Elemente zu bereinigen.
<code>large_memory_pages</code>	Standard: 0(false) Typ: Boolescher Wert Veränderbar: Nein	Falls 1 (true), ElastiCache wird versucht, große Speicherseiten zu verwenden.
<code>lock_down_paged_memory</code>	Standard: 0(false) Typ: Boolescher Wert Veränderbar: Nein	Falls 1 (wahr), ElastiCache wird der gesamte ausgelagerte Speicher gesperrt.
<code>max_item_size</code>	Standard: 1048576 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Die Größe, in Byte, des größten Elements, das im Cluster gespeichert werden kann.
<code>max_simultaneous_connections</code>	Standard: 65000 Typ: Ganzzahl Veränderbar: Nein	Die maximale Anzahl gleichzeitiger Verbindungen.

Name	Details	Description
maximize_core_file_limit	Standard: 0(false) Typ: Boolescher Wert Anpassbar: Änderungen werden wirksam: Nach dem Neustart	Falls 1 (wahr), ElastiCache wird das Limit für die Kerndateien maximiert.
memcached_connections_overhead	Standard: 100 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Die Menge an Speicher, die für Memcached-Verbindungen und verschiedene Overhead-Anforderungen zu reservieren ist. Informationen zu diesem Parameter finden Sie unter Overhead von Memcached-Verbindungen .
requests_per_event	Standard: 20 Typ: Ganzzahl Veränderbar: Nein	Die maximale Anzahl von Anforderungen pro Ereignis für eine bestimmte Verbindung. Dieses Limit wird benötigt, um zu verhindern, dass die Ressourcen aufgebraucht werden.

Overhead von Memcached-Verbindungen

Der auf jedem Knoten zum Speichern von Elementen verfügbar gemachte Speicher ist der verfügbare Gesamtspeicher auf diesem Knoten (der im Parameter `max_cache_memory` gespeichert wird) minus des für Verbindungen und anderes Overhead verwendeten Speichers (der im Parameter `memcached_connections_overhead` gespeichert wird). Ein Knoten vom Typ `cache.m1.small` verfügt z. B. über `max_cache_memory` von 1300 MB. Bei einem `memcached_connections_overhead`-Standardwert von 100 MB hat der Memcached-Prozess 1200 MB zum Speichern von Elementen verfügbar.

Die Standardwerte für den Parameter `memcached_connections_overhead` genügen für die meisten Anwendungsfälle. Die erforderliche Menge, die für das Overhead der Verbindungen zugewiesen werden muss, kann abhängig von mehreren Faktoren, darunter Anforderungsrate, Nutzlastgröße und die Anzahl von Verbindungen, variieren.

Sie können den Wert von `memcached_connections_overhead` ändern, um den Anforderungen Ihrer Anwendung besser zu genügen. Wenn z. B. der Wert des Parameters `memcached_connections_overhead` erhöht wird, verringert sich die Menge des Speichers, der zum Speichern von Elementen verfügbar ist. Auch wird ein größerer Puffer für das Overhead von Verbindungen geschaffen. Wenn der Wert des Parameters `memcached_connections_overhead` verringert wird, erhalten Sie mehr Speicher zum Speichern von Elementen. Dadurch kann sich aber das Risiko bezüglich Swap-Nutzung und Leistungseinbußen erhöhen. Wenn Swap-Nutzung und Leistungseinbußen zu beobachten sind, versuchen Sie, den Wert des Parameters `memcached_connections_overhead` zu verringern.

 Important

Für den Knotentyp `cache.t1.micro` wird der Wert für `memcached_connections_overhead` folgendermaßen bestimmt:

- Wenn Ihr Cluster die Standardparametergruppe verwendet, ElastiCache wird der Wert für `memcached_connections_overhead` auf 13 MB gesetzt.
- Wenn Ihr Cluster eine Parametergruppe nutzt, die Sie selbst erstellt haben, können Sie als Wert für `memcached_connections_overhead` einen Wert Ihrer Wahl festlegen.

Spezifische Parameter des Memcached-Knotentyps

Obwohl die meisten Parameter über einen einzelnen Wert verfügen, ist bei einigen Parametern der jeweilige Wert vom verwendeten Knotentyp abhängig. Die folgende Tabelle zeigt die Standardwerte der Parameter `max_cache_memory` und `num_threads` für jeden Knotentyp. Die Werte für diese Parameter können nicht geändert werden.

Knotentyp	<code>max_cache_memory</code> (in Megabyte)	<code>num_threads</code>
<code>cache.t1.micro</code>	213	1
<code>cache.t2.micro</code>	555	1
<code>cache.t2.small</code>	1588	1
<code>cache.t2.medium</code>	3301	2

Knotentyp	max_cache_memory (in Megabyte)	num_threads
cache.t3.micro	512	2
cache.t3.small	1402	2
cache.t3.medium	3364	2
cache.t4g.micro	512	2
cache.t4g.small	1402	2
cache.t4g.medium	3164	2
cache.m1.small	1301	1
cache.m1.medium	3350	1
cache.m1.large	7100	2
cache.m1.xlarge	14600	4
cache.m2.xlarge	33800	2
cache.m2.2xlarge	30412	4
cache.m2.4xlarge	68000	16
cache.m3.medium	2850	1
cache.m3.large	6200	2
cache.m3.xlarge	13600	4
cache.m3.2xlarge	28600	8
cache.m4.large	6573	2
cache.m4.xlarge	11496	4
cache.m4.2xlarge	30412	8

Knotentyp	max_cache_memory (in Megabyte)	num_threads
cache.m4.4xlarge	62234	16
cache.m4.10xlarge	158355	40
cache.m5.large	6537	2
cache.m5.xlarge	13248	4
cache.m5.2xlarge	26671	8
cache.m5.4xlarge	53516	16
cache.m5.12xlarge	160900	48
cache.m5.24xlarge	321865	96
cache.m6g.large	6537	2
cache.m6g.xlarge	13248	4
cache.m6g.2xlarge	26671	8
cache.m6g.4xlarge	53516	16
cache.m6g.8xlarge	107000	32
cache.m6g.12xlarge	160900	48
cache.m6g.16xlarge	214577	64
cache.c1.xlarge	6600	8
cache.r3.large	13800	2
cache.r3.xlarge	29100	4
cache.r3.2xlarge	59600	8
cache.r3.4xlarge	120600	16

Knotentyp	max_cache_memory (in Megabyte)	num_threads
cache.r3.8xlarge	120600	32
cache.r4.large	12590	2
cache.r4.xlarge	25652	4
cache.r4.2xlarge	51686	8
cache.r4.4xlarge	103815	16
cache.r4.8xlarge	208144	32
cache.r4.16xlarge	416776	64
cache.r5.large	13387	2
cache.r5.xlarge	26953	4
cache.r5.2xlarge	54084	8
cache.r5.4xlarge	108347	16
cache.r5.12xlarge	325400	48
cache.r5.24xlarge	650869	96
cache.r6g.large	13387	2
cache.r6g.xlarge	26953	4
cache.r6g.2xlarge	54084	8
cache.r6g.4xlarge	108347	16
cache.r6g.8xlarge	214577	32
cache.r6g.12xlarge	325400	48
cache.r6g.16xlarge	429154	64

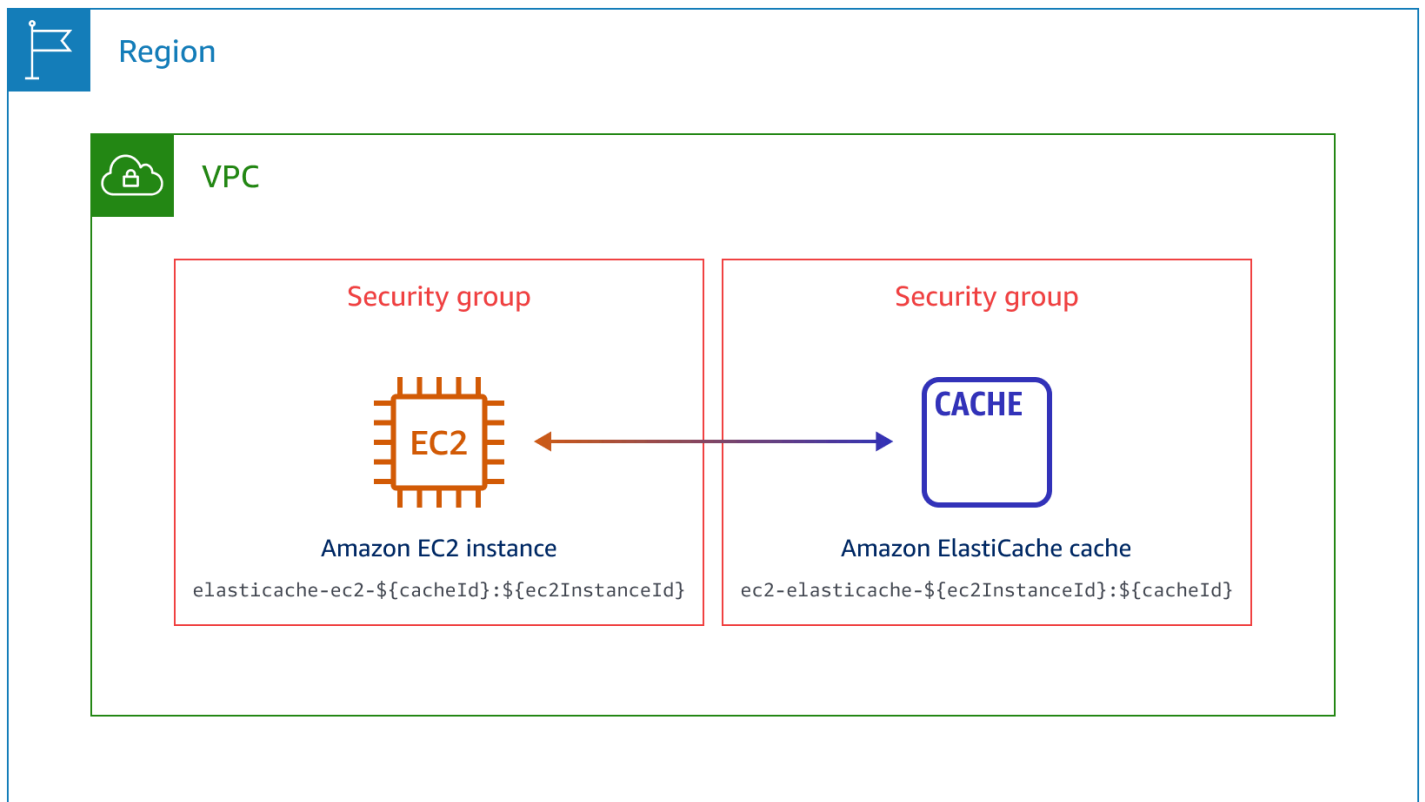
Knotentyp	max_cache_memory (in Megabyte)	num_threads
cache.c7gn.large	3164	2
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

 Note

Alle T2-Instances werden in einer Amazon Virtual Private Cloud (Amazon VPC) erstellt.

Automatisches Verbinden einer EC2 Instanz und eines ElastiCache Caches

Sie können die ElastiCache Konsole verwenden, um die Einrichtung einer Verbindung zwischen einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance und einem ElastiCache Cache zu vereinfachen. Oft befindet sich Ihr Cache in einem privaten Subnetz und Ihre EC2 Instance in einem öffentlichen Subnetz innerhalb einer VPC. Sie können einen SQL-Client auf Ihrer EC2 Instance verwenden, um eine Verbindung zu Ihrem Cache herzustellen. ElastiCache Die EC2 Instanz kann auch Webserver oder Anwendungen ausführen, die auf Ihren privaten ElastiCache Cache zugreifen.



Themen

- [Automatische Konnektivität mit einer EC2 Instanz](#)
- [Anzeigen verbundener Rechenressourcen](#)

Automatische Konnektivität mit einer EC2 Instanz

Wenn Sie eine Verbindung zwischen einer EC2 Instanz und einem ElastiCache Cache einrichten, konfiguriert ElastiCache automatisch die VPC-Sicherheitsgruppe für Ihre EC2 Instanz und für Ihren ElastiCache Cache.

Die folgenden Anforderungen müssen erfüllt sein, um eine EC2 Instanz mit einem ElastiCache Cache zu verbinden:

- Die EC2 Instanz muss in derselben VPC wie der ElastiCache Cache existieren.

Wenn in derselben VPC keine EC2 Instanzen vorhanden sind, bietet die Konsole einen Link zum Erstellen einer Instanz.

- Der Benutzer, der die Konnektivität einrichtet, muss über Berechtigungen verfügen, um die folgenden EC2 Amazon-Operationen auszuführen. Diese Berechtigungen werden in der Regel den EC2 Konten hinzugefügt, wenn sie erstellt werden. Weitere Informationen zu EC2 Berechtigungen finden Sie unter [Gewährung der erforderlichen Berechtigungen für EC2 Amazon-Ressourcen](#).
- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Wenn Sie eine Verbindung zu einer EC2 Instance einrichten, ElastiCache verhält es sich entsprechend der aktuellen Konfiguration der Sicherheitsgruppen, die dem ElastiCache Cache und der EC2 Instance zugeordnet sind, wie in der folgenden Tabelle beschrieben.

Aktuelle Konfiguration der ElastiCache Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	ElastiCache Aktion
Dem ElastiCache Cache sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe	Der EC2 Instanz sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine ausgehende Regel mit der VPC-Sicherheitsgruppe	ElastiCache ergreift keine Maßnahmen. Eine Verbindung zwischen der EC2 Instanz und dem ElastiCache Cache wurde bereits automatisch konfiguriert. Da bereits eine Verbindung zwischen der EC2 Instance und dem ElastiCache Cache besteht, werden die Sicherheitsgruppen nicht geändert.

Aktuelle Konfiguration der ElastiCache Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	ElastiCache Aktion
ppe der EC2 Instance als Quelle.	des ElastiCache Caches als Quelle.	

Aktuelle Konfiguration der ElastiCache Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	ElastiCache Aktion
Es gilt eine der folgenden Bedingungen: • Dem ElastiCache Cache ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster entspricht <code>telasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . • Dem ElastiCache Cache sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht <code>telasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . ElastiCache Kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der EC2 Instance verwenden . ElastiCache kann keine Sicherheitsgruppe verwenden, die keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle hat. ElastiCache kann auch keine Sicherheitsgruppe verwenden, die geändert wurde. Beispiele für Änderungen sind das	Es gilt eine der folgenden Bedingungen: • Der EC2 Instanz ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster entspricht <code>tec2-elasticache-\${ec2InstanceId}:\${cacheId}</code> . • Der EC2 Instanz sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht <code>tec2-elasticache-\${ec2InstanceId}:\${cacheId}</code> . ElastiCache Kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem ElastiCache Cache verwenden. ElastiCache kann keine Sicherheitsgruppe verwenden, die keine ausgehende Regel mit der VPC-Sicherheitsgruppe des ElastiCache Caches als Quelle hat. ElastiCache kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	ELC action: create new security groups

Aktuelle Konfiguration der ElastiCache Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	ElastiCache Aktion
Hinzufügen einer Regel oder das Ändern des Ports einer vorhandenen Regel.		
Dem ElastiCache Cache sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle.	Der EC2 Instanz sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht <code>t.elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . ElastiCache Kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem ElastiCache Cache verwenden. ElastiCache kann keine Sicherheitsgruppe verwenden, die keine ausgehende Regel mit der VPC-Sicherheitsgruppe des ElastiCache Caches als Quelle hat. ElastiCache kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	ELC action: create new security groups

Aktuelle Konfiguration der ElastiCache Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	ElastiCache Aktion
Dem ElastiCache Cache sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle.	Eine gültige EC2 Sicherheitsgruppe für die Verbindung ist vorhanden, aber sie ist der EC2 Instance nicht zugeordnet. Die Sicherheitsgruppe trägt einen Namen, der dem Muster <code>ec2-elasticache-\${ec2InstanceId}:\${cacheId}</code> entspricht. Sie wurde nicht geändert. Es gibt nur eine ausgehende Regel mit der VPC-Sicherheitsgruppe des ElastiCache Caches als Quelle.	ELC action: associate EC2 security group

Aktuelle Konfiguration der ElastiCache Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	ElastiCache Aktion
Es gilt eine der folgenden Bedingungen: • Dem ElastiCache Cache ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster entspricht. <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> • Dem ElastiCache Cache sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht <code>telasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . ElastiCache Kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der EC2 Instance verwenden . ElastiCache kann keine Sicherheitsgruppe verwenden, die keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle hat. ElastiCache kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Der EC2 Instanz sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht <code>ec2-elasticache-\${ec2InstanceId}:\${cacheId}</code> . Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine ausgehende Regel mit der VPC-Sicherheitsgruppe des ElastiCache Caches als Quelle.	ELC action: create new security groups

ElastiCacheAktion: neue Sicherheitsgruppen erstellen

ElastiCache ergreift die folgenden Aktionen:

- Erstellt eine neue Sicherheitsgruppe, die dem Muster `elasticache-ec2-${cacheId}:${ec2InstanceId}` entspricht. Diese Sicherheitsgruppe hat eine eingehende Regel mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle. Diese Sicherheitsgruppe ist dem ElastiCache Cache zugeordnet und ermöglicht der EC2 Instance den Zugriff darauf.
- Erstellt eine neue Sicherheitsgruppe, die dem Muster `elasticache-ec2-${cacheId}:${ec2InstanceId}` entspricht. Diese Sicherheitsgruppe hat eine ausgehende Regel mit der VPC-Sicherheitsgruppe des ElastiCache Caches als Ziel. Diese Sicherheitsgruppe ist der EC2 Instance zugeordnet und ermöglicht es der EC2 Instance, Traffic an den ElastiCache Cache zu senden.

ElastiCache Aktion: EC2 Sicherheitsgruppe zuordnen

ElastiCache ordnet der EC2 Instanz die gültige, bestehende EC2 Sicherheitsgruppe zu. Diese Sicherheitsgruppe ermöglicht es der EC2 Instance, Traffic an den ElastiCache Cache zu senden.

Anzeigen verbundener Rechenressourcen

Sie können die verwendete AWS-Managementkonsole, um die Rechenressourcen anzuzeigen, die mit einem ElastiCache Cache verbunden sind. Zu den angezeigten Ressourcen gehören Rechenressourcenverbindungen, die automatisch eingerichtet wurden. Sie können beispielsweise einer Rechenressource den manuellen Zugriff auf einen Cache ermöglichen, indem Sie der mit dem Cache verknüpften VPC-Sicherheitsgruppe eine Regel hinzufügen. Diese Ressourcen werden nicht in der Liste der verbundenen Rechenressourcen angezeigt.

Damit eine Rechenressource aufgeführt wird, müssen dieselben Bedingungen gelten wie beim automatischen Verbinden einer EC2 Instanz und eines ElastiCache Caches.

Um Rechenressourcen anzuzeigen, die mit einem ElastiCache Cache verbunden sind

1. Melden Sie sich bei der AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole
2. Wählen Sie im Navigationsbereich Caches und dann einen Valkey- oder Redis OSS-Cache aus.
3. Sehen Sie sich auf der Registerkarte Konnektivität und Sicherheit unter Rechenverbindung einrichten die Rechenressourcen an.

Connected compute resources (1) Set up compute connection

Connections to compute resources that were automatically created by ElastiCache for your cache are shown here. Connections to compute resources that were manually created are not displayed.

< 1 >

⚙

Compute resource name	Status	Resource type	Availability zones	Security group
i-0bdb4ea9f9bb36aec	Running	t2.micro	eu-west-3c	ec2-elasticache-i-0bdb4ea9f9bb36aec:cache-foo

Skalierung ElastiCache

Sie können Ihren ElastiCache Cache an Ihre Bedürfnisse anpassen. Serverlose Caches und knotenbasierte Cluster bieten verschiedene Skalierungsoptionen.

ElastiCache Serverlose Skalierung

ElastiCache Serverless passt sich automatisch Ihrem Workload-Verkehr an, wenn er steigt oder fällt. Verfolgt für jeden ElastiCache serverlosen Cache ElastiCache kontinuierlich die Auslastung von Ressourcen wie CPU, Arbeitsspeicher und Netzwerk. Wenn eine dieser Ressourcen eingeschränkt ist, skaliert ElastiCache Serverless, indem es einen neuen Shard hinzufügt und die Daten auf den neuen Shard umverteilt, ohne dass es zu Ausfallzeiten für Ihre Anwendung kommt. Sie können die Ressourcen überwachen, die von Ihrem Cache-In CloudWatch verbraucht werden, indem Sie die `BytesUsedForCache` Metrik für den Cache-Datenspeicher und `ElastiCacheProcessingUnits` (ECPU) für die Computernutzung überwachen.

Festlegen von Skalierungslimits zur Kostenkontrolle

Sie können wählen, ob Sie eine maximale Nutzung sowohl für den Cache-Datenspeicher als auch ECPU/second für Ihren Cache konfigurieren möchten, um die Cache-Kosten zu kontrollieren. Dadurch wird sichergestellt, dass Ihre Cache-Nutzung das konfigurierte Maximum nicht überschreitet.

Wenn Sie ein Skalierungsmaximum festlegen, kann es bei Ihrer Anwendung zu einer verringerten Cache-Leistung kommen, wenn der Cache das Maximum erreicht. Wenn Sie ein Maximum für den Cache-Datenspeicher festlegen und Ihr Cache-Datenspeicher das Maximum erreicht, ElastiCache beginnt das Löschen von Daten in Ihrem Cache, für die ein Time-To-Live (TTL) festgelegt ist. Dabei wird die LRU-Logik verwendet. Wenn es keine Daten gibt, die bereinigt werden können, wird bei Anfragen zum Schreiben zusätzlicher Daten eine Out-of-Memory-Fehlermeldung (OOM) angezeigt. Wenn Sie ein ECPU/second Maximum festlegen und die Rechenauslastung Ihrer Arbeitslast diesen Wert überschreitet, ElastiCache beginnt die Drosselung von Anfragen.

Wenn Sie ein Höchstlimit für BytesUsedForCache oder einrichtenElastiCacheProcessingUnits, empfehlen wir dringend, einen CloudWatch Alarm bei einem niedrigeren Wert als dem Höchstwert einzurichten, damit Sie benachrichtigt werden, wenn Ihr Cache in der Nähe dieser Grenzwerte arbeitet. Wir empfehlen, einen Alarm bei 75 % des von Ihnen festgelegten maximalen Limits einzustellen. Informationen zum Einrichten von CloudWatch Alarmen finden Sie in der Dokumentation.

Vorskalierung mit Serverless ElastiCache

ElastiCache Serverlose Vorskalierung

Mit der Vorskalierung, auch Pre-Warming genannt, können Sie die unterstützten Mindestgrenzen für Ihren Cache festlegen. ElastiCache Sie können diese Mindestwerte für ElastiCache Verarbeitungseinheiten (ECPUs) pro Sekunde oder Datenspeicher festlegen. Dies kann bei der Vorbereitung auf erwartete Skalierungsereignisse nützlich sein. Wenn ein Spieleunternehmen beispielsweise innerhalb der ersten Minute, in der sein neues Spiel veröffentlicht wird, eine 5-fache Zunahme der Logins erwartet, kann es seinen Cache für diesen deutlichen Anstieg der Nutzung vorbereiten.

Sie können die Vorskalierung über die ElastiCache Konsole, CLI oder API durchführen. ElastiCache Serverless aktualisiert die im Cache verfügbaren ECPUs/second Daten innerhalb von 60 Minuten und sendet eine Ereignisbenachrichtigung, wenn die Aktualisierung des Mindestlimits abgeschlossen ist.

So funktioniert die Vorskalierung

Wenn das Mindestlimit für ECPUs/second den Datenspeicher über die Konsole, CLI oder API aktualisiert wird, ist dieses neue Limit innerhalb von 1 Stunde verfügbar. ElastiCache Serverless unterstützt 30 KB ECPUs/second bei leerem Cache und bis zu 90 KB ECPUs/sec bei Verwendung der Funktion „Aus Replikat lesen“. ElastiCache Serverless for Valkey 8.0 kann die Anzahl der unterstützten Anfragen pro Sekunde (RPS) alle 2 bis 3 Minuten verdoppeln, sodass 5 Millionen RPS pro Cache in weniger als 13 Minuten erreicht werden, und das bei gleichbleibender p50-Leselatenz von unter einer Millisekunde. Wenn Sie davon ausgehen, dass ein bevorstehendes Skalierungsereignis diese Rate überschreiten könnte, empfehlen wir, das Minimum auf den Höchstwert festzulegen, den ECPUs/sec Sie mindestens 60 Minuten vor ECPUs/second dem Spitzenereignis erwarten. Andernfalls kann es bei der Anwendung zu einer erhöhten Latenz und einer Drosselung von Anfragen kommen.

Sobald die Aktualisierung des Mindestlimits abgeschlossen ist, beginnt ElastiCache Serverless mit der Berechnung des neuen Mindestwerts ECPUs pro Sekunde oder des neuen Mindestspeichers.

Dies ist auch dann der Fall, wenn Ihre Anwendung keine Anforderungen im Cache ausführt oder wenn Ihre Datenspeichernutzung unter dem Mindestwert liegt. Wenn Sie den Mindestgrenzwert gegenüber der aktuellen Einstellung herabsetzen, erfolgt die Aktualisierung sofort, sodass ElastiCache Serverless sofort mit der Messung des neuen Mindestlimits beginnt.

Note

- Wenn Sie ein Mindestnutzungslimit festlegen, wird Ihnen dieses Limit in Rechnung gestellt, auch wenn Ihre tatsächliche Nutzung unter dem Mindestnutzungslimit liegt. Für die Nutzung von ECPU oder Datenspeicher, die das Mindestnutzungslimit überschreiten, wird der reguläre Tarif berechnet. Wenn Sie beispielsweise ein Mindestnutzungslimit von 100.000 ECPUs/second festlegen, werden Ihnen mindestens 1,224 USD pro Stunde berechnet (unter Verwendung der ECPU-Preise in US-East-1), auch wenn Ihre Nutzung unter dem festgelegten Mindestwert liegt.
- ElastiCache Serverless unterstützt den angeforderten Mindestmaßstab auf aggregierter Ebene im Cache. ElastiCache Serverless unterstützt außerdem maximal 30.000 ECPUs/second pro Steckplatz (90.000 ECPUs/second bei Verwendung von Read from Replica mit READONLY-Verbindungen). Als bewährte Methode sollte Ihre Anwendung sicherstellen, dass die Schlüsselverteilung zwischen den Valkey- oder Redis-OSS-Steckplätzen und der Datenverkehr zwischen den Schlüsseln so einheitlich wie möglich sind.

Skalierungsgrenzen mithilfe der Konsole festlegen und AWS CLI

Skalierungsgrenzen mithilfe der AWS Konsole festlegen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich die Engine aus, die auf dem Cache ausgeführt wird, den Sie ändern möchten.
3. Es wird eine Liste der Caches der ausgewählten Engine angezeigt.
4. Wählen Sie den zu ändernden Cache aus, indem Sie das Optionsfeld links neben dem Cache-Namen anklicken.
5. Wählen Sie Aktionen und dann Ändern.
6. Legen Sie unter Nutzungslimits die entsprechenden Speicher - oder Rechenlimits fest.
7. Klicken Sie auf Änderungen in der Vorschau anzeigen und dann auf Änderungen speichern.

Festlegen von Skalierungsgrenzen mit dem AWS CLI

Verwenden Sie die `modify-serverless-cache` API, um Skalierungsgrenzen mithilfe der CLI zu ändern.

Linux:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Entfernen von Skalierungslimits mithilfe der CLI

Um Skalierungsgrenzen mithilfe der CLI zu entfernen, setzen Sie die Parameter `Minimum` und `Maximum` auf 0.

Linux:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
ECPUPerSecond={Minimum=0,Maximum=0}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
ECPUPerSecond={Minimum=0,Maximum=0}'
```

Skalierung knotenbasierter Cluster

Die Datenmenge, die von einer Anwendung verarbeitet wird, ist selten statisch. Sie steigt und sinkt mit dem Unternehmenswachstum und unterliegt normalen Schwankungen im Bedarf. Wenn Sie Ihren Cache selbst verwalten, müssen Sie für Bedarfsspitzen ausreichend Hardware bereitstellen – dies kann kostspielig sein. Mit Amazon können ElastiCache Sie skalieren, um der aktuellen Nachfrage

gerecht zu werden, und zahlen nur für das, was Sie tatsächlich nutzen. ElastiCache ermöglicht es Ihnen, Ihren Cache an die Nachfrage anzupassen.

Note

Wenn ein Valkey- oder Redis-OSS-Cluster in einer oder mehreren Regionen repliziert wird, werden diese Regionen der Reihe nach skaliert. Bei der Skalierung werden zuerst sekundäre Regionen und dann die primäre Region skaliert. Beim Herunterskalieren steht die primäre Region an erster Stelle und dann folgen alle sekundären Regionen.

Bei der Aktualisierung der Engine-Version lautet die Reihenfolge sekundäre Region und dann primäre Region.

Themen

- [Skalierung auf Abruf für Memcached-Cluster](#)
- [Manuelle Skalierung für Memcached-Cluster](#)
- [Skalierung für Valkey- oder Redis OSS-Cluster \(Clustermodus deaktiviert\)](#)
- [Skalierung von Replikatknoten für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#)
- [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus aktiviert\)](#)

Skalierung auf Abruf für Memcached-Cluster

ElastiCache for Memcached bietet einen vollständig verwalteten In-Memory-Caching-Service, der Memcached in der Cloud bereitstellt, betreibt und vertikal skaliert. AWS

Vertikale Skalierung auf Abruf

Mit vertikaler Skalierung bietet Memcached ein leistungsstarkes, verteiltes Speicher-Caching-System, das häufig verwendet wird, um dynamische Anwendungen zu beschleunigen, indem die Datenbanklast verringert wird. ElastiCache Es speichert Daten und Objekte im RAM, sodass weniger Daten aus externen Datenquellen gelesen werden müssen.

Sie können vertikale Skalierung sowohl auf bestehende als auch auf neue knotenbasierte Cluster anwenden. Dies kann für Flexibilität bei der Ressourcenzuweisung sorgen, sodass Benutzer sich effizient an sich ändernde Workloads anpassen können, ohne die Clusterarchitektur zu ändern. Diese Skalierungsfähigkeit verbessert die Leistung, indem sie die Cache-Kapazität in Zeiten hoher Nachfrage erhöht und in Zeiten geringer Nachfrage nach unten skaliert, um

die Kosten zu optimieren. Dies vereinfacht den Betrieb, macht die Erstellung neuer Cluster für wechselnde Ressourcenanforderungen überflüssig und ermöglicht eine schnelle Reaktion auf Verkehrsschwankungen. Insgesamt kann die vertikale Skalierung für knotenbasierte Memcached-Cluster dazu beitragen, die Kosteneffizienz zu erhöhen, die Ressourcennutzung zu verbessern und es Benutzern sogar zu ermöglichen, ihren Memcached-Instance-Typ zu ändern. All das erleichtert es Benutzern, ihre Caching-Infrastruktur an den tatsächlichen Anwendungsanforderungen auszurichten.

Note

- Änderungen des Knotentyps sind nur für knotenbasierte Memcached-Cluster mit Engine-Versionen 1.5 oder höher verfügbar.
- Auto Discovery muss aktiviert sein, um die vertikale Skalierung nutzen zu können.

Einrichtung der vertikalen Skalierung auf Abruf für knotenbasierte Memcached-Cluster

Sie können die vertikale Skalierung auf Abruf für Memcached mit `konfigurierenscale-config`, die zwei Parameter enthält:

1. `ScaleIntervalMinutes`: Zeit (in Minuten) zwischen den Skalierungsstapeln während des Memcached-Upgrade-Vorgangs
2. `ScalePercentage`: Prozentsatz der Knoten, die während des Memcached-Upgrade-Vorgangs gleichzeitig skaliert werden

Konvertierung eines vorhandenen Memcached-Knotentyps in einen Cache, der über die CLI vertikal skaliert werden kann

Um einen vorhandenen knotenbasierten Memcached-Cluster in einen Cache umzuwandeln, der vertikal skaliert werden kann, können Sie dies über die CLI verwenden `elasticache modify-cache-cluster`.

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id <your-cluster-id> \  
  --cache-node-type <new-node-type> \  
  --scale-config <scale-config> \  
  --apply-immediately
```

Vertikale Skalierung mit der CLI einrichten

Um die vertikale Skalierung für einen knotenbasierten Memcached-Cluster über die CLI einzurichten, verwenden Sie `elasticache modify-cache-cluster with scale-config` und seine Parameter und. `ScalePercentage` `ScaleIntervalMinutes`

- `scale-interval-minutes`: Dies definiert die Zeit (in Minuten) zwischen den Skalierungsstapeln. Diese Einstellung kann zwischen 2 und 30 Minuten liegen. Wenn kein Wert angegeben ist, wird der Standardwert von 5 Minuten angewendet.
- `Skalierungsprozentsatz`: Dies gibt den Prozentsatz der Knoten an, die in jedem Stapel gleichzeitig skaliert werden sollen. Diese Einstellung kann zwischen 10 und 100 liegen. Die Einstellung wird beim Dividieren aufgerundet. Wenn das Ergebnis beispielsweise 49,5 wäre, wird die Einstellung 50 angewendet. Wenn kein Wert angegeben ist, wird der Standardwert 20 angewendet.

Diese Konfigurationsoptionen ermöglichen es Ihnen, den Skalierungsprozess an Ihre spezifischen Bedürfnisse anzupassen und dabei ein Gleichgewicht zwischen der Minimierung von Clusterunterbrechungen und der Optimierung der Skalierungsgeschwindigkeit zu finden. Der Parameter `scale-config` gilt nur für Memcached-Engine-Typen und wird für andere Cache-Engines ignoriert, sodass die Abwärtskompatibilität mit der bestehenden API-Nutzung für andere Cluster gewährleistet ist.

API-Aufruf

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id <your-cluster-id> \  
  --cache-node-type <new-node-type> \  
  --scale-config '{  
    "ScalePercentage": 30,  
    "ScaleIntervalMinutes": 2  
  }'  
  --apply-immediately
```

Ergebnis:

Gibt die Cluster-ID und die ausstehende Änderung zurück.

```
{  
  "CacheCluster": {  
    "CacheNodeType": "old_instance_type",  
    ...  
    ...  
  }  
}
```



```
    "PendingModifiedValues": {  
      "CacheNodeType": "new_instance_type"  
    },  
  }  
}
```

Listet Ihre vertikale Skalierungseinstellung für den Memcached-Cache auf

Sie können die Skalierungsoptionen für Ihre Memcache-Caches abrufen und sehen, welche aktuellen Optionen für die vertikale Skalierung verfügbar sind.

API-Aufruf

```
aws elasticache list-allowed-node-type-modifications --cache-cluster-id <your-cluster-id>
```

Ergebnis:

```
{  
  "ScaleUpModifications": [  
    "cache.x.xxxx",  
    "cache.x.xxxx"  
  ],  
  "ScaleDownModifications": [  
    "cache.x.xxxx",  
    "cache.x.xxxx",  
    "cache.x.xxxx"  
  ]  
}
```

Vertikale Skalierung für Memcached mit dem AWS-Managementkonsole

Gehen Sie wie folgt vor, um mit dem einen AWS-Managementkonsole knotenbasierten Memcached-Cluster in einen vertikal skalierbaren Cluster umzuwandeln.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie den zu konvertierenden Memcached-Cluster aus.
3. Wählen Sie die Registerkarte Ändern aus.
4. Gehen Sie zum Abschnitt Cache-Einstellungen und wählen Sie den gewünschten Knotentyp aus.

5. Wählen Sie „Änderungen in der Vorschau anzeigen“ und überprüfen Sie die Änderungen.
6. Wählen Sie Modify aus.

Automatisierte horizontale Skalierung für Memcached

ElastiCache lässt sich jetzt in den Service AWS Application Auto Scaling (AAS) integrieren, um automatisierte horizontale Skalierung für Memcached-Cluster zu ermöglichen. Sie können Skalierungsrichtlinien über den AWS Application Auto Scaling Scaling-Dienst definieren und die Anzahl der Knoten in Memcached-Clustern automatisch nach Bedarf anpassen, basierend auf vordefinierten Metriken oder Zeitplänen.

Note

Automatisierte horizontale Skalierung ist derzeit in den Regionen Peking und Ningxia nicht verfügbar.

Dies sind die verfügbaren Methoden für die automatische horizontale Skalierung Ihrer knotenbasierten Cluster.

- **Geplante Skalierung:** Die Skalierung auf der Grundlage eines Zeitplans ermöglicht es Ihnen, Ihren eigenen Skalierungsplan für vorhersehbare Laständerungen festzulegen. Angenommen, die Auslastung Ihrer Webanwendung steigt am Mittwoch, bleibt am Donnerstag auf diesem hohen Niveau und fällt am Freitag ab. Sie können Auto Scaling so konfigurieren, dass die Kapazität am Mittwoch erhöht und die Kapazität am Freitag verringert wird.
- **Zielverfolgung:** Bei Skalierungsrichtlinien für die Zielverfolgung wählen Sie eine Skalierungsmetrik und legen einen Zielwert fest. Application Auto Scaling erstellt und verwaltet die CloudWatch Alarmer, die die Skalierungsrichtlinie auslösen, und berechnet die Skalierungsanpassung auf der Grundlage der Metrik und des Zielwerts. Durch die Skalierungsrichtlinie wird so viel Kapazität wie erforderlich hinzugefügt oder entfernt, damit die Metrik auf oder nahe an dem Zielwert gehalten wird.

So richten Sie die horizontale Skalierung für einen knotenbasierten Memcached-Cluster über die CLI ein

Bei der horizontalen Skalierung eines knotenbasierten Memcached-Clusters können Sie eine Ziel-Tracking-Richtlinie, eine geplante Richtlinie oder beides verwenden.

1. Registrieren Sie eine Ressource als skalierbares Ziel

Rufen Sie die `RegisterScalableTarget` API in AWS Application Auto Scaling auf, um das Ziel für die skalierbare Dimension zu registrieren `elasticache:cache-cluster:Nodes`.

API: `ApplicationAutoScaling.RegisterScalableTarget`

Eingabe:

```
{
  "ScalableDimension": "elasticache:cache-cluster:Nodes",
  "ResourceId": "cache-cluster/test-cluster-1",
  "ServiceNamespace": "elasticache",
  "MinCapacity": 20,
  "MaxCapacity": 50
}
```

2. Erstellen Sie eine Skalierungsrichtlinie für Target-Tracking

Als Nächstes können Sie eine Skalierungsrichtlinie für die Zielverfolgung für die Ressource erstellen, indem Sie die API für die Skalierungsrichtlinie aufrufen.

3. Vordefinierte Metrik

Im Folgenden finden Sie eine Richtlinie, die entlang der Cache-Node-Dimension skaliert und dabei die vordefinierte Metrik verwendet `ElastiCacheCPUUtilization`, die für Cluster `test-cluster-1` auf 50 festgelegt ist. Beim Löschen von Knoten für das Skalieren werden die letzten `n` Knoten entfernt.

API: `ApplicationAutoScaling.PutScalingPolicy`

Eingabe:

```
{
  "PolicyName": "cpu50-target-tracking-scaling-policy",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "TargetValue": 50,
    "PredefinedMetricSpecification": {
      "PredefinedMetricType": "ElastiCacheCPUUtilization"
    },
    "ScaleOutCooldown": 600,
    "ScaleInCooldown": 600
  }
}
```

```

    },
    "ServiceNamespace": "elasticache",
    "ScalableDimension": "elasticache:cache-cluster:Nodes",
    "ResourceId": "cache-cluster/test-cluster-1"
}

```

Ausgabe:

```

{
  "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/elasticache/cache-cluster/test-cluster-1:policyName/cpu50-target-tracking-scaling-policy",
  "Alarms": [
    {
      "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca",
      "AlarmName": "TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca"
    },
    {
      "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d",
      "AlarmName": "TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
    }
  ]
}

```

4. Benutzerdefinierte Metrik

Sie können auch eine Skalierungsrichtlinie für die Dimension festlegen, indem Sie einen benutzerdefinierten Prozentsatz verwenden, der auf der Cloudwatch-Metrik basiert.

Eingabe:

```

{
  "PolicyName": "cpu50-target-tracking-scaling-policy",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "CustomizedMetricSpecification": {
      "Dimensions": [

```

```
{
  "Name": "MyMetricDimension",
  "Value": "DimensionValue"
},
{
  "MetricName": "MyCustomMetric",
  "Namespace": "MyNamespace",
  "Statistic": "Average",
  "Unit": "Percent"
},
{
  "TargetValue": 40,
  "ScaleOutCooldown": 600,
  "ScaleInCooldown": 600
},
{
  "ServiceNamespace": "elasticache",
  "ScalableDimension": "elasticache:cache-cluster:Nodes",
  "ResourceId": "cache-cluster/test-cluster-1"
}
```

5. Geplante Aktionen

Wenn Sie für ein bestimmtes Ereignis die Skalierung vornehmen und nach dem Ereignis wieder skalieren müssen, können Sie zwei geplante Aktionen erstellen, indem Sie die `PutScheduledAction` API aufrufen.

Richtlinie 1: Skalierung

Der `at` Befehl in `--schedule` plant, dass die Aktion einmal an einem bestimmten Datum und zu einer bestimmten Uhrzeit in der future ausgeführt wird. Das Zeitplanfeld unterstützt auch Rate (Minute, Stunde, Tag usw.) und Cron (für Cron-Ausdruck).

Zum angegebenen Datum und zur angegebenen Uhrzeit aktualisiert Application Auto Scaling die Werte `MinCapacity` und `MaxCapacity`. Application Auto Scaling skaliert `MinCapacity` auf bis zu 70 Cache-Knoten.

API: `ApplicationAutoScaling.PutScheduledAction`

Eingabe:

```
{
  "ResourceId": "elasticache:ache-cluster:test-cluster-1",
  "ScalableDimension": "elasticache:cache-cluster:Nodes",
```

```
"ScalableTargetAction": {
  "MaxCapacity": 100,
  "MinCapacity": 70
},
"Schedule": "at(2020-05-20T17:05:00)",
"ScheduledActionName": "ScalingOutScheduledAction",
"ServiceNamespace": "elasticache",
}
```

Richtlinie 2: Skalierung

Am angegebenen Datum und zur angegebenen Uhrzeit aktualisiert Application Auto Scaling die Werte `MinCapacity` und und skaliert auf `MaxCapacity`, sodass die Cache-Knoten wieder auf 60 zurückgesetzt werden.

API: `ApplicationAutoScaling.PutScheduledAction`

Eingabe:

```
{
  "ResourceId": "elasticache:cache-cluster:test-cluster-1",
  "ScalableDimension": "elasticache:cache-cluster:Nodes",
  "ScalableTargetAction": {
    "MaxCapacity": 60,
    "MinCapacity": 40
  },
  "Schedule": "at(2020-05-21T17:05:00)",
  "ScheduledActionName": "ScalingInScheduledAction",
  "ServiceNamespace": "elasticache",
}
```

6. Sehen Sie sich die Skalierungsaktivitäten an

Sie können die Skalierungsaktivitäten mithilfe der `DescribeScalingActivities` API anzeigen.

API: `ApplicationAutoScaling.DescribeScalingActivities`

Ausgabe:

```
{
  "ScalingActivities": [
```

```
{
  "ScalableDimension": "elasticache:elasticache:DesiredCount",
  "Description": "Setting desired count to 30.",
  "ResourceId": "elasticache/cache-cluster/test-cluster-1",
  "ActivityId": "4d759079-a31f-4d0c-8468-504c56e2eecf",
  "StartTime": 1462574194.658,
  "elasticacheNamespace": "elasticache",
  "EndTime": 1462574276.686,
  "Cause": "monitor alarm TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca in state ALARM triggered policy cpu50-target-tracking-scaling-policy",
  "StatusMessage": "Failed to set desired count to 30",
  "StatusCode": "Failed"
},
{
  "ScalableDimension": "elasticache:elasticache:DesiredCount",
  "Description": "Setting desired count to 25.",
  "ResourceId": "elasticache/cache-cluster/test-cluster-1",
  "ActivityId": "90aff0eb-dd6a-443c-889b-b809e78061c1",
  "StartTime": 1462574254.223,
  "elasticacheNamespace": "elasticache",
  "EndTime": 1462574333.492,
  "Cause": "monitor alarm TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca in state ALARM triggered policy cpu50-target-tracking-scaling-policy",
  "StatusMessage": "Successfully set desired count to 25. Change successfully fulfilled by elasticache.",
  "StatusCode": "Successful"
}
]
```

7. Skalierungsrichtlinie bearbeiten/löschen

Sie können Richtlinien bearbeiten oder löschen, indem Sie die `PutScalingPolicy` API erneut aufrufen oder Action aufrufen `DeleteScalingPolicy`. `DeleteScheduled`

8. Registrierung skalierbarer Ziele aufheben

Sie können die Registrierung des skalierbaren Ziels über die `DeregisterScalableTarget` API aufheben. Durch die Deregistrierung eines skalierbaren Ziels werden die Skalierungsrichtlinien und die damit verbundenen geplanten Aktionen gelöscht.

API: `ApplicationAutoScaling DeregisterScalableTarget`

Eingabe:

```
{
  "ResourceId": "elasticache/cache-cluster/test-cluster-1",
  "ServiceNamespace": "elasticache",
  "ScalableDimension": "elasticache:cache-cluster:Nodes"
}
```

9. Skalierung der Richtlinienbereinigung

10. Mehrere Skalierungsrichtlinien

Sie können mehrere Skalierungsrichtlinien erstellen. Im Folgenden finden Sie wichtige Hinweise zum Verhalten von [Auto Scaling Target Tracking](#).

- Sie können mehrere Skalierungsrichtlinien für die Ziel-Nachverfolgung für ein skalierbares Ziel besitzen, vorausgesetzt, dass diese alle verschiedene Metriken verwenden.
- Die Absicht von Application Auto Scaling ist es, der Verfügbarkeit immer Vorrang einzuräumen. Daher unterscheidet sich das Verhalten von Application Auto Scaling, je nachdem, ob die Ziel-Tracking-Richtlinien für die Skalierung nach außen oder nach innen bereit sind. Sofern Richtlinien für die Ziel-Nachverfolgung für die horizontale Skalierung nach oben bereit sind, findet eine horizontale Skalierung des skalierbaren Ziels nach oben statt. Eine horizontale Skalierung nach unten wird jedoch nur vorgenommen, wenn alle Richtlinien für die Ziel-Nachverfolgung (mit aktivierter horizontaler Skalierung nach unten) zur horizontalen Skalierung nach unten bereit sind.
- Wenn mehrere Richtlinien das skalierbare Ziel gleichzeitig zum Skalieren nach außen oder nach innen anweisen, erfolgt Application Auto Scaling auf der Grundlage der Richtlinie, die die größte Kapazität sowohl für die Skalierung nach innen als auch nach außen bietet. Dies bietet Ihnen eine größere Flexibilität für verschiedene Szenarien und stellt sicher, dass immer ausreichend Kapazität vorhanden ist, um Ihre Anwendungs-Workloads zu verarbeiten.

 Note

AWS Application Auto Scaling stellt Skalierungsrichtlinien nicht in eine Warteschlange. Application Auto Scaling wartet, bis die erste Skalierung abgeschlossen ist, kühlt sich dann ab und wiederholt dann den obigen Algorithmus.

Automatisches horizontales Skalieren eines knotenbasierten Memcached-Clusters über AWS-Managementkonsole

Gehen Sie wie folgt vor, um mit dem einen vorhandenen AWS-Managementkonsole knotenbasierten Memcache-Cluster in einen horizontal skalierbaren Cluster umzuwandeln.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie den zu konvertierenden Memcached-Cache aus.
3. Gehen Sie zur Registerkarte Autoscaling.
4. Wählen Sie die anzuwendende Skalierungsrichtlinie aus, indem Sie entweder Dynamische Skalierung hinzufügen oder Geplante Skalierung hinzufügen auswählen.
5. Geben Sie nach Bedarf die Details für die ausgewählte Richtlinie ein.
6. Klicken Sie auf Create.

Manuelle Skalierung für Memcached-Cluster

Die manuelle horizontale Skalierung eines Memcached-Clusters nach innen oder außen ist so einfach wie das Hinzufügen oder Entfernen von Knoten aus dem Cluster. Memcached-Cluster bestehen aus 1 bis 60 Knoten.

Da Sie in einem Memcached-Cluster Ihre Daten auf alle vorhandenen Knoten verteilen können, müssen Sie selten auf einen Knotentyp mit mehr Speicher hochskalieren. Da die Memcached-Engine jedoch keine Daten speichert, ist Ihr neuer Cluster bei der Skalierung auf einen anderen Knotentyp zunächst leer, es sei denn, Ihre Anwendung füllt ihn auf.

Um Ihren Memcached-Cluster manuell vertikal zu skalieren, müssen Sie einen neuen Cluster erstellen. Memcached-Cluster sind zunächst leer, bis sie von der Anwendung befüllt werden.

Manuelles Skalieren von Memcached-Clustern

Action	Thema
Ausskalieren	Hinzufügen von Knoten zu einem Cluster
Einskalieren	Löschen von Knoten aus einem Cluster

Action	Thema
Ändern von Knotentypen	Manuelles vertikales Skalieren knotenbasierter Memcached-Cluster

Themen

- [Manuelles horizontales Skalieren eines knotenbasierten Memcached-Clusters](#)
- [Manuelles vertikales Skalieren knotenbasierter Memcached-Cluster](#)

Manuelles horizontales Skalieren eines knotenbasierten Memcached-Clusters

Die Memcached-Engine unterstützt die Partitionierung Ihrer Daten über mehrere Knoten hinweg. Dadurch lassen sich Memcached-Cluster einfach horizontal skalieren. Ein Memcached-Cluster wird einfach durch Hinzufügen oder Entfernen von Knoten horizontal skaliert.

In den folgenden Themen wird erläutert, wie Sie Memcached-Cluster durch Hinzufügen oder Entfernen von Knoten ein- oder ausskalieren.

- [Hinzufügen von Knoten zu einem Cluster](#)
- [Löschen von Knoten aus Ihrem Cluster](#)

Jedes Mal, wenn Sie die Anzahl der Knoten in Ihrem Memcached-Cluster ändern, müssen Sie zumindest einen Teil Ihres Keyspace neu zuordnen, sodass er dem richtigen Knoten zugeordnet ist. Detaillierte Informationen zum Lastausgleich Ihres Memcached-Clusters finden Sie unter [Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich \(Memcached\)](#).

Wenn Sie für Ihr Memcached-Cluster Auto Discovery verwenden, müssen Sie die Endpunkte in Ihrer Anwendung nicht ändern, wenn Sie Knoten hinzufügen oder entfernen. Weitere Informationen zu Auto Discovery finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#). Wenn Sie Auto Discovery nicht verwenden, müssen Sie bei jeder Änderung der Anzahl der Knoten in Ihrem Memcached-Cluster die Endpunkte in Ihrer Anwendung aktualisieren.

Manuelles vertikales Skalieren knotenbasierter Memcached-Cluster

Wenn Sie Ihren Memcached-Cluster manuell nach oben oder unten skalieren, müssen Sie einen neuen Cluster erstellen. Memcached-Cluster sind zunächst leer, bis sie von der Anwendung befüllt werden.

 Important

Vergewissern Sie sich beim Skalieren zu einem kleineren Knotentyp davon, dass der Speicher des kleineren Knotentyps für Ihre Daten und den Overhead ausreicht. Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Themen

- [Vertikales Skalieren eines knotenbasierten Memcached-Clusters \(Konsole\)](#)
- [Vertikales Skalieren eines knotenbasierten Memcached-Clusters \(AWS CLI\)](#)
- [Vertikale Skalierung eines knotenbasierten Memcached-Clusters \(API\) ElastiCache](#)

Vertikales Skalieren eines knotenbasierten Memcached-Clusters (Konsole)

Das folgende Verfahren führt Sie durch die vertikale Skalierung eines knotenbasierten Memcached-Clusters mithilfe von AWS-Managementkonsole

1. Erstellen Sie einen neuen Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).
2. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Die Endpunkte eines Clusters finden \(Konsole\) \(Memcached\)](#).
3. Löschen Sie den alten Cluster. Weitere Informationen finden Sie unter [Löschen eines neuen Knotens in Memcached](#).

Vertikales Skalieren eines knotenbasierten Memcached-Clusters (AWS CLI)

Das folgende Verfahren führt Sie durch die vertikale Skalierung eines knotenbasierten Memcached-Clusters mithilfe von AWS CLI

1. Erstellen Sie einen neuen Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Erstellen eines Clusters \(AWS CLI\)](#).
2. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Suchen von Endpunkten \(AWS CLI\)](#).
3. Löschen Sie den alten Cluster. Weitere Informationen finden Sie unter [Verwenden Sie den AWS CLI, um einen ElastiCache Cluster zu löschen](#).

Vertikale Skalierung eines knotenbasierten Memcached-Clusters (API) ElastiCache

Das folgende Verfahren führt Sie durch die vertikale Skalierung eines knotenbasierten Memcache-Clusters mithilfe der API. ElastiCache

1. Erstellen Sie einen neuen Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Einen Cluster für Memcached \(API\) erstellen ElastiCache](#).
2. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Endpunkte finden \(API\) ElastiCache](#).
3. Löschen Sie den alten Cluster. Weitere Informationen finden Sie unter [Verwenden der ElastiCache API](#).

Skalierung für Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert)

Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) können ein Einzelknoten-Cluster mit 0 Shards oder Mehrknoten-Cluster mit 1 Shard sein. Cluster mit nur einem Knoten verwenden einen Knoten sowohl für Lese- als auch für Schreibvorgänge. Cluster mit mehreren Knoten haben immer einen Knoten als read/write primären Knoten mit 0 bis 5 schreibgeschützten Replikatknoten.

Themen

- [Skalierung für Valkey- oder Redis OSS-Cluster \(Clustermodus deaktiviert\)](#)

Skalierung von Valkey- oder Redis OSS-Clustern

Action	Valkey oder Redis OSS (Clustermodus deaktiviert)	Valkey oder Redis OSS (Clustermodus aktiviert)
Einskalieren	Knoten aus einem ElastiCache Cluster entfernen	Skalierung von Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert)
Ausskalieren	Hinzufügen von Knoten zu einem Cluster	Online-Resharding für Valkey oder Redis OSS (Clustermodus aktiviert)
Ändern von Knotentypen	Zu einem größeren Knotentyp: • Skalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten • Skalierung von Valkey- oder Redis OSS-Clustern mit Replikaten Zu einem kleineren Knotentyp: • Herunterskalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten	Vertikales Online-Skalieren durch Ändern des Knotentyps

Action	Valkey oder Redis OSS (Clustermodus deaktiviert)	Valkey oder Redis OSS (Clustermodus aktiviert)
	• Herunterskalierung von Valkey- oder Redis OSS-Clustern mit Replikaten	
Ändern der Anzahl der Knotengruppen	Wird für Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) nicht unterstützt	Skalierung von Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert)

Inhalt

- [Skalierung für Valkey- oder Redis OSS-Cluster \(Clustermodus deaktiviert\)](#)
- [Skalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten](#)
 - [Skalierung von Valkey- oder Redis OSS-Clustern \(Clustermodus deaktiviert\) \(Konsole\) mit einem Knoten](#)
 - [Skalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten \(\)AWS CLI](#)
 - [Skalierung von Valkey- oder Redis OSS-Clustern \(API\) mit einem Knoten ElastiCache](#)
- [Herunterskalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten](#)
 - [Herunterskalierung eines Valkey- oder Redis OSS-Clusters mit einem Knoten \(Konsole\)](#)
 - [Herunterskalierung der Valkey- oder Redis OSS-Cluster mit einem Knoten \(\)AWS CLI](#)
 - [Herunterskalierung der Valkey- oder Redis OSS-Cluster \(API\) mit einem Knoten ElastiCache](#)

Skalierung für Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert)

Valkey- oder Redis OSS-Knoten (Clustermodus deaktiviert) müssen groß genug sein, um alle Cache-Daten plus den Valkey- oder Redis-OSS-Overhead aufzunehmen. Um die Datenkapazität Ihres Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) zu ändern, müssen Sie vertikal skalieren. Sie müssen auf einen größeren Knotentyp skalieren, um die Datenkapazität zu erhöhen, oder auf einen kleineren Knotentyp herunterskalieren, um die Datenkapazität zu reduzieren.

Der ElastiCache Skalierungsprozess ist darauf ausgelegt, Ihre vorhandenen Daten bestmöglich beizubehalten, und erfordert eine erfolgreiche Valkey- oder Redis-OSS-Replikation. Für Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) empfehlen wir, Valkey oder Redis OSS ausreichend Arbeitsspeicher zur Verfügung zu stellen.

Sie können Ihre Daten nicht auf mehrere Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) partitionieren. Wenn Sie jedoch nur die Lesekapazität Ihres Clusters erhöhen oder verringern müssen, können Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) mit Replikatknoten erstellen und Read Replicas hinzufügen oder entfernen. Informationen zum Erstellen eines Valkey- oder Redis OSS-Clusters (Clustermodus deaktiviert) mit Replikatknoten, die Ihren Valkey- oder Redis OSS-Cluster mit einem Knoten als primärem Cluster verwenden, finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)

Nachdem Sie den Cluster mit Replikaten erstellt haben, können Sie die Lesekapazität durch Hinzufügen von Read Replicas erweitern. Später kann die Lesekapazität durch Entfernen von Read Replicas wieder reduziert werden. Für weitere Informationen siehe [Erhöhen der Lesekapazität](#) oder [Verringern der Lesekapazität](#).

Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) mit Replikaten bieten nicht nur die Möglichkeit, die Lesekapazität zu skalieren, sondern bieten auch weitere Geschäftsvorteile. Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

Important

Wenn Ihre Parametergruppe Speicher für Valkey- oder Redis-OSS-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. `reserved-memory` Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

Themen

- [Skalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten](#)
- [Herunterskalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten](#)

Skalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten

Wenn Sie einen Valkey- oder Redis OSS-Cluster mit einem Knoten skalieren, ElastiCache führt der folgende Prozess durch, unabhängig davon, ob Sie die ElastiCache Konsole, die oder die API verwenden. AWS CLI ElastiCache

1. Ein neuer Cluster mit dem neuen Knotentyp wird in derselben Availability Zone wie der vorhandene Cluster eingerichtet.
2. Die Cache-Daten im vorhandenen Cluster werden in den neuen Cluster kopiert. Wie lange dieser Vorgang dauert, hängt von Ihrem Knotentyp und davon ab, wie viele Daten sich im Cluster befinden.
3. Lese- und Schreibvorgänge werden jetzt über den neuen Cluster bedient. Da die Endpunkte des neuen Clusters dieselben sind wie die des alten Clusters, müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Sie werden eine kurze Unterbrechung (einige Sekunden) der Lese- und Schreiboperationen des Primärknotens feststellen, während der DNS-Eintrag aktualisiert wird.
4. ElastiCache löscht den alten Cluster. Sie werden eine kurze Unterbrechung (einige Sekunden) der Lese- und Schreiboperationen des alten Knotens feststellen, da die Verbindungen zum alten Knoten getrennt werden.

Note

Bei Clustern, auf denen der R6gd-Knotentyp ausgeführt wird, können Sie nur auf Knotengrößen innerhalb der R6gd-Knotenfamilie skalieren.

Wie in der folgenden Tabelle dargestellt, ist Ihr Valkey- oder Redis-OSS-Scale-Up-Vorgang blockiert, wenn Sie für das nächste Wartungsfenster ein Engine-Upgrade geplant haben. Weitere Informationen zu Wartungsfenstern finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

Blockierte Valkey- oder Redis OSS-Operationen

Ausstehende Operationen	Blockierte Operationen
Aufwärtsskalierung	Unmittelbares Engine-Upgrade
Engine-Upgrade	Unmittelbares Aufwärtsskalieren

Ausstehende Operationen	Blockierte Operationen
Aufwärtsskalierung und Engine-Upgrade	Unmittelbares Aufwärtsskalieren
	Unmittelbares Engine-Upgrade

Wenn Sie durch ausstehende Operationen blockiert werden, haben Sie folgende Möglichkeiten.

- Planen Sie Ihren Valkey- oder Redis OSS-Scale-up-Vorgang für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Sofort anwenden deaktivieren (CLI use: `--no-apply-immediately`, API use: `ApplyImmediately=false`)
- Warten Sie bis zu Ihrem nächsten Wartungsfenster (oder danach), um Ihren Valkey- oder Redis OSS-Skalierungsvorgang durchzuführen.
- Fügen Sie das Valkey- oder Redis OSS-Engine-Upgrade zu dieser Cluster-Änderung hinzu, wobei das Kontrollkästchen Sofort anwenden aktiviert ist (CLI use: `--apply-immediately`, API use: `ApplyImmediately=true`). Dadurch wird die Skalierung freigegeben, da das Engine-Upgrade sofort ausgeführt wird.

Sie können einen Valkey- oder Redis OSS-Cluster mit einem Knoten (Clustermodus deaktiviert) mithilfe der ElastiCache Konsole, der oder der API hochskalieren. AWS CLI ElastiCache

Important

Wenn Ihre Parametergruppe Speicher für Valkey- oder Redis OSS-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. `reserved-memory` Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

Skalierung von Valkey- oder Redis OSS-Clustern (Clustermodus deaktiviert) (Konsole) mit einem Knoten

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder Redis OSS-Cluster mit einem Knoten mithilfe der Management Console skalieren. ElastiCache Während dieses Vorgangs bearbeitet Ihr Valkey- oder Redis OSS-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

So skalieren Sie einen Valkey- oder Redis OSS-Cluster mit einem Knoten (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey- oder Redis OSS-Cluster aus.
3. Wählen Sie aus der Clusterliste den Cluster aus, den Sie skalieren möchten (auf ihm muss die Valkey- oder Redis OSS-Engine ausgeführt werden, nicht die geclusterte Valkey- oder Redis OSS-Engine).
4. Wählen Sie Ändern aus.
5. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
 - a. Wählen Sie in der Liste Node type den Knotentyp aus, auf den Sie skalieren möchten.
 - b. Wenn Sie zur Speicherverwaltung `reserved-memory` verwenden, wählen Sie in der Liste Parameter Group die benutzerdefinierte Parametergruppe aus, die die korrekte Speichermenge für den neuen Knotentyp reserviert.
6. Wenn Sie die Skalierung sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately. Wenn Sie das Kontrollkästchen für Apply immediately nicht aktivieren, wird die Skalierung während des nächsten Wartungsfensters des Clusters durchgeführt.
7. Wählen Sie Ändern aus.

Wenn Sie im vorherigen Schritt Apply immediately aktiviert haben, ändert sich der Cluster-Status zu `modifying`. Wenn der Status zu `available` wechselt, ist die Änderung abgeschlossen und Sie können den neuen Cluster verwenden.

Skalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten (AWS CLI)

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder Redis-OSS-Cluster mit einem Knoten mithilfe von skalieren.AWS CLI Während dieses Vorgangs bearbeitet Ihr Valkey- oder Redis OSS-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Um einen Valkey- oder Redis OSS-Cluster mit einem Knoten zu skalieren ()AWS CLI

1. Ermitteln Sie die Knotentypen, auf die Sie skalieren können, indem Sie den `aws elasticache list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.

- `--cache-cluster-id`

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --cache-cluster-id my-cache-cluster-id
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --cache-cluster-id my-cache-cluster-id
```

Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ],  
  "ScaleDownModifications": [  
    "cache.t2.micro",  
    "cache.t2.small",  
    "cache.t2.medium",  
    "cache.t1.small",  
  ],  
}
```

```
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-modifications](#) in der AWS CLI-Referenz.

2. Ändern Sie Ihren vorhandenen Cluster, indem Sie den zu skalierenden Cluster und den neuen, größeren Knotentyp angeben. Verwenden Sie dazu den `AWS CLI modify-cache-cluster` Befehl und die folgenden Parameter.
 - `--cache-cluster-id`— Der Name des Clusters, den Sie skalieren möchten.
 - `--cache-node-type`— Der neue Knotentyp, für den Sie den Cluster skalieren möchten. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Befehl `list-allowed-node-type-modifications` zurückgegeben wurden.
 - `--cache-parameter-group-name` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
 - `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie den Parameter `--no-apply-immediately`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cache-cluster \  
  --cache-node-type cache.m3.xlarge \  
  --cache-parameter-group-name redis32-m2-xl \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-redis-cache-cluster ^  
  --cache-node-type cache.m3.xlarge ^  
  --cache-parameter-group-name redis32-m2-xl ^  
  --apply-immediately
```

Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format).

```
{
  "CacheCluster": {
    "Engine": "redis",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "CacheParameterGroupName": "default.redis6.x",
      "ParameterApplyStatus": "in-sync"
    },
    "SnapshotRetentionLimit": 1,
    "CacheClusterId": "my-redis-cache-cluster",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1,
    "SnapshotWindow": "00:00-01:00",
    "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "PreferredAvailabilityZone": "us-west-2a",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "6.0",
    "PendingModifiedValues": {
      "CacheNodeType": "cache.m3.2xlarge"
    },
    "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled"
  }
}
```

Weitere Informationen finden Sie unter [modify-cache-cluster](#) in der AWS CLI-Referenz.

3. Wenn Sie den verwendet haben `--apply-immediately`, überprüfen Sie den Status des neuen Clusters mithilfe des `AWS CLI describe-cache-clusters` Befehls mit dem folgenden Parameter. Wenn sich der Status auf verfügbar ändert, können Sie mit der Verwendung des neuen, größeren Clusters beginnen.
 - `--cache-cluster-id`— Der Name Ihres Valkey- oder Redis OSS-Clusters mit einem Knoten. Verwenden Sie diesen Parameter, um einen bestimmten Cluster und nicht alle Cluster zu beschreiben.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Weitere Informationen finden Sie unter [describe-cache-clusters](#) in der AWS CLI-Referenz.

Skalierung von Valkey- oder Redis OSS-Clustern (API) mit einem Knoten ElastiCache

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder Redis-OSS-Cluster mit einem Knoten mithilfe der API skalieren. ElastiCache Während dieses Vorgangs bearbeitet Ihr Valkey- oder Redis OSS-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Um einen Valkey- oder Redis OSS-Cluster (API) mit einem Knoten zu skalieren ElastiCache

1. Ermitteln Sie die Knotentypen, auf die Sie skalieren können, indem Sie die ElastiCache `ListAllowedNodeTypeModifications` API-Aktion mit dem folgenden Parameter ausführen.
 - `CacheClusterId`— Der Name des Valkey- oder Redis-OSS-Clusters mit einem Knoten, den Sie skalieren möchten.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&CacheClusterId=MyRedisCacheCluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [ListAllowedNodeTypeModifications](#) in der Amazon ElastiCache API-Referenz.

2. Ändern Sie Ihren vorhandenen Cluster und geben Sie den zu skalierenden Cluster und den neuen, größeren Knotentyp an. Verwenden Sie dazu die `ModifyCacheCluster` ElastiCache API-Aktion und die folgenden Parameter.
 - `CacheClusterId`— Der Name des Clusters, den Sie skalieren möchten.
 - `CacheNodeType`— Der neue, größere Knotentyp, auf den Sie den Cluster skalieren möchten. Dieser Wert muss einer der Knotentypen sein, die von der

ListAllowedNodeTypeModifications Aktion im vorherigen Schritt zurückgegeben wurden.

- **CacheParameterGroupName** – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
- **ApplyImmediately** Legen Sie `true` fest, um die Skalierung sofort auszuführen. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=true  
  &CacheClusterId=MyRedisCacheCluster  
  &CacheNodeType=cache.m3.xlarge  
  &CacheParameterGroupName=redis32-m2-xl  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [ModifyCacheCluster](#) in der Amazon ElastiCache API-Referenz.

3. Falls Sie dies verwendet haben `ApplyImmediately=true`, überprüfen Sie den Status des neuen Clusters mithilfe der ElastiCache `DescribeCacheClusters` API-Aktion mit dem folgenden Parameter. Wenn sich der Status auf `available` ändert, können Sie mit der Verwendung des neuen, größeren Clusters beginnen.
- **CacheClusterId**— Der Name Ihres Valkey- oder Redis OSS-Clusters mit einem Knoten. Verwenden Sie diesen Parameter, um einen bestimmten Cluster und nicht alle Cluster zu beschreiben.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=MyRedisCacheCluster  
  &Version=2015-02-02
```

```
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [DescribeCacheClusters](#) in der Amazon ElastiCache API-Referenz.

Herunterskalierung von Valkey- oder Redis OSS-Clustern mit einem Knoten

In den folgenden Abschnitten erfahren Sie, wie Sie einen Valkey- oder Redis OSS-Cluster mit einem Knoten auf einen kleineren Knotentyp herunterskalieren können. Es ist wichtig für den langfristigen Erfolg Ihres neuen Valkey- oder Redis OSS-Clusters sicherzustellen, dass der neue, kleinere Knotentyp groß genug ist, um alle Daten und den OSS-Overhead von Valkey oder Redis aufzunehmen. Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#).

Note

Bei Clustern, auf denen der R6gd-Knotentyp ausgeführt wird, können Sie nur auf Knotengrößen innerhalb der R6gd-Knotenfamilie skalieren.

Themen

- [Herunterskalierung eines Valkey- oder Redis OSS-Clusters mit einem Knoten \(Konsole\)](#)
- [Herunterskalierung der Valkey- oder Redis OSS-Cluster mit einem Knoten \(AWS CLI\)](#)
- [Herunterskalierung der Valkey- oder Redis OSS-Cluster \(API\) mit einem Knoten ElastiCache](#)

Herunterskalierung eines Valkey- oder Redis OSS-Clusters mit einem Knoten (Konsole)

Das folgende Verfahren führt Sie durch die Skalierung Ihres Valkey- oder Redis OSS-Clusters mit einem Knoten auf einen kleineren Knotentyp mithilfe der Konsole. ElastiCache

Important

Wenn Ihre Parametergruppe Speicher für Valkey- oder Redis-OSS-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. `reserved-memory` Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

Um Ihren Valkey- oder Redis OSS-Cluster (Konsole) mit einem Knoten zu verkleinern

1. Vergewissern Sie sich, dass der Speicher des kleineren Knotentyps für Ihre Daten und den Overhead ausreicht.
2. Wenn Ihre Parametergruppe Speicher für Valkey- oder Redis-OSS-Overhead reserviert, stellen Sie sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, um die richtige Speichermenge für Ihren neuen Knotentyp bereitzustellen. `reserved-memory`

Alternativ können Sie eine benutzerdefinierte Parametergruppe für die Verwendung von `reserved-memory-percent` anpassen. Weitere Informationen finden Sie unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

3. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
4. Wählen Sie aus der Liste der Cluster den Cluster aus, den Sie nach unten skalieren möchten. Auf diesem Cluster muss die Valkey- oder Redis OSS-Engine ausgeführt werden und nicht die geclusterte Valkey- oder Redis OSS-Engine.
5. Wählen Sie Ändern aus.
6. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
 - a. Wählen Sie in der Liste Node type (Knotentyp) den Knotentyp aus, auf den Sie herunterskalieren möchten.
 - b. Wenn Sie zur Speicherverwaltung `reserved-memory` verwenden, wählen Sie in der Liste Parameter Group die benutzerdefinierte Parametergruppe aus, die die korrekte Speichermenge für den neuen Knotentyp reserviert.
7. Wenn Sie die Skalierung nach unten sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately (Sofort anwenden). Wenn Sie das Kontrollkästchen für Apply immediately (Sofort anwenden) nicht aktivieren, wird die Skalierung nach unten während des nächsten Wartungsfensters des Clusters durchgeführt.
8. Wählen Sie Ändern aus.
9. Wenn der Status des Clusters von modifying zu available wechselt, wurde die Skalierung auf den neuen Knotentyp erfolgreich abgeschlossen. Die Endpunkte müssen nicht in der Anwendung aktualisiert werden.

Herunterskalierung der Valkey- oder Redis OSS-Cluster mit einem Knoten ()AWS CLI

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder Redis-OSS-Cluster mit einem Knoten mithilfe von herunterskalieren.AWS CLI

So verkleinern Sie einen Valkey- oder Redis OSS-Cluster mit einem Knoten ()AWS CLI

1. Ermitteln Sie die Knotentypen, auf die Sie herunterskalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.

- `--cache-cluster-id`

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
  --cache-cluster-id my-cache-cluster-id
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
  --cache-cluster-id my-cache-cluster-id
```

Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format).

```
{
  "ScaleUpModifications": [
    "cache.m3.2xlarge",
    "cache.m3.large",
    "cache.m3.xlarge",
    "cache.m4.10xlarge",
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
    "cache.r3.large",
    "cache.r3.xlarge"
  ]
}
```

```
    "ScaleDownModifications": [  
        "cache.t2.micro",  
        "cache.t2.small ",  
        "cache.t2.medium ",  
        "cache.t1.small ",  
    ],  
  
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-modifications](#) in der AWS CLI-Referenz.

2. Ändern Sie Ihren vorhandenen Cluster, indem Sie den Cluster, der herunterskaliert werden soll, und den neuen, kleineren Knotentyp angeben. Verwenden Sie dazu den `aws elasticache modify-cache-cluster` Befehl und die folgenden Parameter.
 - `--cache-cluster-id`— Der Name des Clusters, den Sie herunterskalieren.
 - `--cache-node-type`— Der neue Knotentyp, für den Sie den Cluster skalieren möchten. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Befehl `list-allowed-node-type-modifications` zurückgegeben wurden.
 - `--cache-parameter-group-name` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
 - `--apply-immediately` – Bewirkt, dass der Verkleinerungsprozess sofort angewendet wird. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie den Parameter `--no-apply-immediately`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
    --cache-cluster-id my-redis-cache-cluster \  
    --cache-node-type cache.m3.xlarge \  
    --cache-parameter-group-name redis32-m2-xl \  
    --apply-immediately
```

Für Windows:

```
aws elasticache modify-cache-cluster ^
--cache-cluster-id my-redis-cache-cluster ^
--cache-node-type cache.m3.xlarge ^
--cache-parameter-group-name redis32-m2-xl ^
--apply-immediately
```

Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format).

```
{
  "CacheCluster": {
    "Engine": "redis",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "CacheParameterGroupName": "default.redis6.x",
      "ParameterApplyStatus": "in-sync"
    },
    "SnapshotRetentionLimit": 1,
    "CacheClusterId": "my-redis-cache-cluster",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1,
    "SnapshotWindow": "00:00-01:00",
    "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "PreferredAvailabilityZone": "us-west-2a",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "6.0",
    "PendingModifiedValues": {
      "CacheNodeType": "cache.m3.2xlarge"
    },
    "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled"
  }
}
```

Weitere Informationen finden Sie unter [modify-cache-cluster](#) in der AWS CLI-Referenz.

3. Wenn Sie den verwendet haben `--apply-immediately`, überprüfen Sie den Status des neuen Clusters mithilfe des `AWS CLIdescribe-cache-clusters` Befehls mit dem folgenden

Parameter. Wenn sich der Status auf verfügbar ändert, können Sie mit der Verwendung des neuen, größeren Clusters beginnen.

- `--cache-cluster-id`— Der Name Ihres Valkey- oder Redis OSS-Clusters mit einem Knoten. Verwenden Sie diesen Parameter, um einen bestimmten Cluster und nicht alle Cluster zu beschreiben.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Weitere Informationen finden Sie unter [describe-cache-clusters](#) in der AWS CLI-Referenz.

Herunterskalierung der Valkey- oder Redis OSS-Cluster (API) mit einem Knoten ElastiCache

Das folgende Verfahren beschreibt, wie ein Valkey- oder Redis-OSS-Cluster mit einem Knoten mithilfe der API nach oben und unten skaliert wird. ElastiCache

So verkleinern Sie einen Valkey- oder Redis OSS-Cluster (API) mit einem Knoten ElastiCache

1. Ermitteln Sie die Knotentypen, auf die Sie herunterskalieren können, indem Sie die ElastiCache `ListAllowedNodeTypeModifications` API-Aktion mit dem folgenden Parameter ausführen.
 - `CacheClusterId`— Der Name des Valkey- oder Redis-OSS-Clusters mit einem Knoten, den Sie herunterskalieren möchten.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&CacheClusterId=MyRedisCacheCluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [ListAllowedNodeTypeModifications](#) in der Amazon ElastiCache API-Referenz.

2. Ändern Sie Ihren vorhandenen Cluster und geben Sie den zu skalierenden Cluster und den neuen, größeren Knotentyp an. Verwenden Sie dazu die `ModifyCacheCluster` ElastiCache API-Aktion und die folgenden Parameter.
 - `CacheClusterId`— Der Name des Clusters, den Sie herunterskalieren.
 - `CacheNodeType`— Der neue, kleinere Knotentyp, auf den Sie den Cluster herunterskalieren möchten. Dieser Wert muss einer der Knotentypen sein, die von der `ListAllowedNodeTypeModifications` Aktion im vorherigen Schritt zurückgegeben wurden.
 - `CacheParameterGroupName` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
 - `ApplyImmediately` – Legen Sie `true` fest, um die Skalierung sofort auszuführen. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&ApplyImmediately=true  
&CacheClusterId=MyRedisCacheCluster  
&CacheNodeType=cache.m3.xlarge  
&CacheParameterGroupName redis32-m2-xl  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [ModifyCacheCluster](#) in der Amazon ElastiCache API-Referenz.

3. Falls Sie dies verwendet haben `ApplyImmediately=true`, überprüfen Sie den Status des neuen Clusters mithilfe der ElastiCache `DescribeCacheClusters` API-Aktion mit dem folgenden Parameter. Wenn sich der Status auf `available` ändert, können Sie mit der Verwendung des neuen, kleineren Clusters beginnen.

- **CacheClusterId**— Der Name Ihres Valkey- oder Redis OSS-Clusters mit einem Knoten. Verwenden Sie diesen Parameter, um einen bestimmten Cluster und nicht alle Cluster zu beschreiben.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=MyRedisCacheCluster  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [DescribeCacheClusters](#) in der Amazon ElastiCache API-Referenz.

Skalierung von Replikatknoten für Valkey oder Redis OSS (Clustermodus deaktiviert)

Ein Valkey- oder Redis-OSS-Cluster mit Replikatknoten (in der API/CLI als Replikationsgruppe bezeichnet) bietet Hochverfügbarkeit über eine Replikation, bei der Multi-AZ mit aktiviertem automatischem Failover aktiviert ist. Ein Cluster mit Replikatknoten ist eine logische Sammlung von bis zu sechs Valkey- oder Redis-OSS-Knoten, wobei ein Knoten, der primäre, sowohl Lese- als auch Schreib Anforderungen bearbeiten kann. Alle anderen Knoten im Cluster sind reine Read Replicas des primären Clusters. Daten, die im primären Cluster geschrieben werden, werden asynchron auf die Read Replicas im Cluster repliziert. Da Valkey oder Redis OSS (Clustermodus deaktiviert) die Partitionierung Ihrer Daten auf mehrere Cluster nicht unterstützen, enthält jeder Knoten in einer Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) den gesamten Cache-Datensatz. Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) unterstützen die Partitionierung Ihrer Daten auf bis zu 500 Shards.

Um die Datenkapazität Ihres Clusters zu ändern, müssen Sie es auf einen größeren Knotentyp hoch- oder auf einen kleineren Knotentyp herabskalieren.

Um die Lesekapazität Ihres Clusters zu ändern, fügen Sie bis auf maximal 5 weitere Lesereplikate hinzu oder entfernen Sie Lesereplikate.

Der ElastiCache Skalierungsprozess ist darauf ausgelegt, Ihre vorhandenen Daten bestmöglich beizubehalten, und erfordert eine erfolgreiche Valkey- oder Redis-OSS-Replikation. Für Valkey- oder Redis OSS-Cluster mit Replikaten empfehlen wir, Valkey oder Redis OSS ausreichend Speicher zur Verfügung zu stellen.

Themen

- [Skalierung von Valkey- oder Redis OSS-Clustern mit Replikaten](#)
- [Herunterskalierung von Valkey- oder Redis OSS-Clustern mit Replikaten](#)
- [Erhöhen der Lesekapazität](#)
- [Verringern der Lesekapazität](#)

Verwandte Themen

- [Hohe Verfügbarkeit mit Replikationsgruppen](#)
- [Replikation: Valkey- und Redis OSS-Clustermodus deaktiviert oder aktiviert](#)
- [Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS](#)

- [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#)

Themen

- [Skalierung von Valkey- oder Redis OSS-Clustern mit Replikaten](#)
- [Herunterskalierung von Valkey- oder Redis OSS-Clustern mit Replikaten](#)
- [Erhöhen der Lesekapazität](#)
- [Verringern der Lesekapazität](#)

Skalierung von Valkey- oder Redis OSS-Clustern mit Replikaten

Amazon ElastiCache bietet Konsolen-, CLI- und API-Unterstützung für die Skalierung Ihrer Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus deaktiviert).

Wenn der Scale-Up-Prozess eingeleitet wird, ElastiCache geht Folgendes vor:

1. Es wird eine Replikationsgruppe unter Verwendung des neuen Knotentyps gestartet.
2. Alle Daten aus dem derzeitigen primären Knoten werden in den neuen primären Knoten kopiert.
3. Die neuen Read Replicas werden mit dem neuen primären Knoten synchronisiert.
4. Die DNS-Einträge werden aktualisiert und verweisen nun auf die neuen Knoten. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei Redis OSS Version 4.0.10 und niedriger stellen Sie möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der DNS-Eintrag aktualisiert wird.
5. Die alten Knoten (CLI/API: Replikationsgruppe) werden gelöscht. Sie werden eine kurze Unterbrechung (ein paar Sekunden) der Lese- und Schreiboperationen der alten Knoten feststellen, da die Verbindungen zu den alten Knoten getrennt werden.

Die Dauer dieses Prozess ist abhängig vom Knotentyp und der Datenmenge im Cluster.

Wie in der folgenden Tabelle dargestellt, ist Ihr Valkey- oder Redis OSS-Scale-up-Vorgang blockiert, wenn Sie ein Engine-Upgrade für das nächste Wartungsfenster des Clusters geplant haben.

Blockierte Valkey- oder Redis OSS-Operationen

Ausstehende Operationen	Blockierte Operationen
Aufwärtsskalierung	Unmittelbares Engine-Upgrade
Engine-Upgrade	Unmittelbares Aufwärtsskalieren
Aufwärtsskalierung und Engine-Upgrade	Unmittelbares Aufwärtsskalieren
	Unmittelbares Engine-Upgrade

Wenn Sie durch ausstehende Operationen blockiert werden, haben Sie folgende Möglichkeiten.

- Planen Sie Ihren Valkey- oder Redis OSS-Scale-up-Vorgang für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Sofort anwenden deaktivieren (CLI use: `--no-apply-immediately`, API use: `ApplyImmediately=false`).
- Warten Sie bis zu Ihrem nächsten Wartungsfenster (oder danach), um Ihren Valkey- oder Redis OSS-Scale-up-Vorgang durchzuführen.
- Fügen Sie das Valkey- oder Redis OSS-Engine-Upgrade zu dieser Cluster-Änderung hinzu, wobei das Kontrollkästchen Sofort anwenden aktiviert ist (CLI use: `--apply-immediately`, API use: `ApplyImmediately=true`). Dadurch wird die Skalierung freigegeben, da das Engine-Upgrade sofort ausgeführt wird.

In den folgenden Abschnitten wird beschrieben, wie Sie Ihren Valkey- oder Redis-OSS-Cluster mithilfe der ElastiCache Konsole, der und der API skalieren, wobei Replikate aktiviert werden. AWS CLI ElastiCache

Important

Wenn Ihre Parametergruppe Speicher für `reserved-memory` den Valkey- oder Redis-OSS-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

Skalierung eines Valkey- oder Redis OSS-Clusters mit Replikaten (Konsole)

Wie lange die Aufwärtsskalierung auf einen größeren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cluster ab.

Der folgende Prozess skaliert Ihren Cluster mit Replikaten mithilfe der Konsole vom aktuellen Knotentyp auf einen neuen, größeren Knotentyp. ElastiCache Während dieses Vorgangs kann für andere Versionen eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten auftreten, während der DNS-Eintrag aktualisiert wird. Für Knoten, die auf 5.0.6-Versionen ausgeführt

werden, tritt möglicherweise eine Ausfallzeit unter 1 Sekunde und bei älteren Versionen von wenigen Sekunden auf.

So skalieren Sie den Valkey- oder Redis OSS-Cluster mit Replikaten (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus
3. Wählen Sie aus der Liste der Cluster den Cluster aus, den Sie skalieren möchten. Auf diesem Cluster muss die Valkey- oder Redis OSS-Engine ausgeführt werden und nicht die geclusterte Valkey- oder Redis OSS-Engine.
4. Wählen Sie Ändern aus.
5. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
 - a. Wählen Sie in der Liste Node type den Knotentyp aus, auf den Sie skalieren möchten. Beachten Sie, dass nicht alle Knotentypen für das Herunterskalieren zur Verfügung stehen.
 - b. Wenn Sie zur Speicherverwaltung `reserved-memory` verwenden, wählen Sie in der Liste Parameter Group die benutzerdefinierte Parametergruppe aus, die die korrekte Speichermenge für den neuen Knotentyp reserviert.
6. Wenn Sie die Skalierung sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately. Wenn Sie das Kontrollkästchen für Apply immediately nicht aktivieren, wird die Skalierung während des nächsten Wartungsfensters des Clusters durchgeführt.
7. Wählen Sie Ändern aus.
8. Wenn der Status des Clusters von `modifying` zu `available` wechselt, wurde die Skalierung auf den neuen Knotentyp erfolgreich abgeschlossen. Die Endpunkte müssen nicht in der Anwendung aktualisiert werden.

Skalierung einer Valkey- oder Redis OSS-Replikationsgruppe (AWS CLI)

Im nachfolgenden Verfahren wird eine Replikationsgruppe mithilfe der AWS CLI vom derzeitigen Knotentyp auf einen neuen, größeren Knotentyp skaliert. Während dieses Vorgangs werden die DNS-Einträge ElastiCache aktualisiert, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei Version 4.0.10 und niedriger stellen Sie bei

vorherigen Versionen möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten fest, während der DNS-Eintrag aktualisiert wird.

Die Zeit, die für die Skalierung auf einen größeren Knotentyp benötigt wird, hängt von Ihrem Knotentyp und der Datenmenge in Ihrem aktuellen Cluster ab.

So skalieren Sie eine Valkey- oder Redis OSS-Replikationsgruppe ()AWS CLI

1. Ermitteln Sie, auf welche Knotentypen Sie skalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.
 - `--replication-group-id` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
  --replication-group-id my-repl-group
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
  --replication-group-id my-repl-group
```

Die Ausgabe dieser Operation sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
  "ScaleUpModifications": [
    "cache.m3.2xlarge",
    "cache.m3.large",
    "cache.m3.xlarge",
    "cache.m4.10xlarge",
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
```

```
        "cache.r3.large",  
        "cache.r3.xlarge"  
    ]  
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-modifications](#) in der AWS CLI-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe des `AWS CLI modify-replication-group` Befehls mit den folgenden Parametern auf den neuen Knotentyp.
 - `--replication-group-id` der Name der Replikationsgruppe.
 - `--cache-node-type`— der neue, größere Knotentyp der Cluster in dieser Replikationsgruppe. Dieser Wert muss einer der Instance-Typen sein, die vom `list-allowed-node-type-modifications` Befehl im vorherigen Schritt zurückgegeben wurden.
 - `--cache-parameter-group-name` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
 - `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `--no-apply-immediately`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
    --replication-group-id my-repl-group \  
    --cache-node-type cache.m3.xlarge \  
    --cache-parameter-group-name redis32-m3-2x1 \  
    --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
    --replication-group-id my-repl-group ^  
    --cache-node-type cache.m3.xlarge ^  
    --cache-parameter-group-name redis32-m3-2x1 \  
    --apply-immediately
```

```
--apply-immediately
```

Die Ausgabe dieses Befehls sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
  "ReplicationGroup": {
    "Status": "available",
    "Description": "Some description",
    "NodeGroups": [{
      "Status": "available",
      "NodeGroupMembers": [{
        "CurrentRole": "primary",
        "PreferredAvailabilityZone": "us-west-2b",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "my-repl-group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
        }
      }],
      "CacheClusterId": "my-repl-group-001"
    },
    {
      "CurrentRole": "replica",
      "PreferredAvailabilityZone": "us-west-2c",
      "CacheNodeId": "0001",
      "ReadEndpoint": {
        "Port": 6379,
        "Address": "my-repl-group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
      }
    },
    "CacheClusterId": "my-repl-group-002"
  ]
},
  "NodeGroupId": "0001",
  "PrimaryEndpoint": {
    "Port": 6379,
    "Address": "my-repl-group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
  }
},
  "ReplicationGroupId": "my-repl-group",
  "SnapshotRetentionLimit": 1,
  "AutomaticFailover": "disabled",
  "SnapshotWindow": "12:00-13:00",
  "SnapshottingClusterId": "my-repl-group-002",
```



```
"MemberClusters": [  
  "my-repl-group-001",  
  "my-repl-group-002"  
],  
"PendingModifiedValues": {}  
}  
}
```

Weitere Informationen finden Sie unter [modify-replication-group](#) in der AWS CLI-Referenz.

3. Wenn Sie den `--apply-immediately` Parameter verwendet haben, überwachen Sie den Status der Replikationsgruppe mithilfe des `AWS CLI describe-replication-group` Befehls mit dem folgenden Parameter. Wenn der Status noch wird geändert lautet, tritt möglicherweise eine Ausfallzeit unter 1 Sekunde für Knoten auf, die auf 5.0.6-Versionen ausgeführt werden, und für ältere Versionen eine kurze Unterbrechung von Lese- und Schreibvorgängen vom Primärknoten, während der DNS-Eintrag aktualisiert wird.
- `--replication-group-id` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

Für Linux, macOS oder Unix:

```
aws elasticache describe-replication-groups \  
  --replication-group-id my-replication-group
```

Für Windows:

```
aws elasticache describe-replication-groups ^  
  --replication-group-id my-replication-group
```

Weitere Informationen finden Sie unter [describe-replication-groups](#) in der AWS CLI-Referenz.

Skalierung einer Valkey- oder Redis OSS-Replikationsgruppe (API) ElastiCache

Der folgende Prozess skaliert Ihre Replikationsgruppe mithilfe der API von ihrem aktuellen Knotentyp auf einen neuen, größeren Knotentyp. ElastiCache Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei Version Redis OSS 4.0.10 und niedriger stellen Sie

möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der DNS-Eintrag aktualisiert wird.

Die Zeit, die für die Skalierung auf einen größeren Knotentyp benötigt wird, hängt von Ihrem Knotentyp und der Datenmenge in Ihrem aktuellen Cluster ab.

So skalieren Sie eine Valkey- oder Redis OSS-Replikationsgruppe (API) ElastiCache

1. Ermitteln Sie mit dem folgenden Parameter, auf welche Knotentypen Sie mithilfe der ElastiCache `ListAllowedNodeTypeModifications` API-Aktion skalieren können.
 - `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&ReplicationGroupId=MyReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [ListAllowedNodeTypeModifications](#) in der Amazon ElastiCache API-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe der `ModifyReplicationGroup` ElastiCache API-Aktion und mit den folgenden Parametern auf den neuen Knotentyp.
 - `ReplicationGroupId` der Name der Replikationsgruppe.
 - `CacheNodeType`— der neue, größere Knotentyp der Cluster in dieser Replikationsgruppe. Dieser Wert muss einer der Instanztypen sein, die von der `ListAllowedNodeTypeModifications` Aktion im vorherigen Schritt zurückgegeben wurden.
 - `CacheParameterGroupName` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen

Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.

- `ApplyImmediately` Legen Sie `true` fest, um die Skalierung sofort auszuführen. Wenn Sie die Skalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup  
  &ApplyImmediately=true  
  &CacheNodeType=cache.m3.2xlarge  
  &CacheParameterGroupName=redis32-m3-2x1  
  &ReplicationGroupId=myReplGroup  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie [ModifyReplicationGroup](#) in der Amazon ElastiCache API-Referenz.

3. Falls Sie dies verwendet haben `ApplyImmediately=true`, überwachen Sie den Status der Replikationsgruppe mithilfe der `ElastiCache DescribeReplicationGroups` API-Aktion mit den folgenden Parametern. Wenn der Status von `modifying` zu `available` wechselt, können Sie die neue, aufskalierte Replikationsgruppe verwenden.
- `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeReplicationGroups  
  &ReplicationGroupId=MyReplGroup  
  &Version=2015-02-02
```

```
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [DescribeReplicationGroups](#) in der Amazon ElastiCache API-Referenz.

Herunterskalierung von Valkey- oder Redis OSS-Clustern mit Replikaten

In den folgenden Abschnitten erfahren Sie, wie Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) mit Replikatknoten auf einen kleineren Knotentyp herunterskalieren. Es ist für den Erfolg wichtig sicherzustellen, dass der neue, kleinere Knotentyp für die Datenmenge und den Overhead ausreicht. Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#).

Note

Bei Clustern, auf denen der R6gd-Knotentyp ausgeführt wird, können Sie nur auf Knotengrößen innerhalb der R6gd-Knotenfamilie skalieren.

Important

Wenn Ihre Parametergruppe Speicher für Valkey- oder Redis OSS-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. `reserved-memory` Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

Herunterskalierung eines Valkey- oder Redis OSS-Clusters mit Replikaten (Konsole)

Der folgende Prozess skaliert Ihren Valkey- oder Redis OSS-Cluster mit Replikatknoten mithilfe der Konsole auf einen kleineren Knotentyp. ElastiCache

So verkleinern Sie einen Valkey- oder Redis OSS-Cluster mit Replikatknoten (Konsole)

1. Vergewissern Sie sich, dass der Speicher des kleineren Knotentyps für Ihre Daten und den Overhead ausreicht.
2. Wenn Ihre Parametergruppe Speicher für Valkey- oder Redis-OSS-Overhead reserviert, stellen Sie sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, um die richtige Speichermenge für Ihren neuen Knotentyp bereitzustellen. `reserved-memory`

Alternativ können Sie eine benutzerdefinierte Parametergruppe für die Verwendung von `reserved-memory-percent` anpassen. Weitere Informationen finden Sie unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

3. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
4. Wählen Sie aus der Liste der Cluster den Cluster aus, den Sie nach unten skalieren möchten. Auf diesem Cluster muss die Valkey- oder Redis OSS-Engine ausgeführt werden und nicht die geclusterte Valkey- oder Redis OSS-Engine.
5. Wählen Sie Ändern aus.
6. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
 - a. Wählen Sie in der Liste Node type (Knotentyp) den Knotentyp aus, auf den Sie herunterskalieren möchten.
 - b. Wenn Sie zur Speicherverwaltung `reserved-memory` verwenden, wählen Sie in der Liste Parameter Group die benutzerdefinierte Parametergruppe aus, die die korrekte Speichermenge für den neuen Knotentyp reserviert.
7. Wenn Sie die Skalierung nach unten sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately (Sofort anwenden). Wenn Sie das Kontrollkästchen für Apply immediately (Sofort anwenden) nicht aktivieren, wird die Skalierung nach unten während des nächsten Wartungsfensters des Clusters durchgeführt.
8. Wählen Sie Ändern aus.
9. Wenn der Status des Clusters von `modifying` zu `available` wechselt, wurde die Skalierung auf den neuen Knotentyp erfolgreich abgeschlossen. Die Endpunkte müssen nicht in der Anwendung aktualisiert werden.

Herunterskalierung einer Valkey- oder Redis OSS-Replikationsgruppe ()AWS CLI

Im nachfolgenden Verfahren wird eine Replikationsgruppe mithilfe der AWS CLI vom derzeitigen Knotentyp auf einen neuen, kleineren Knotentyp skaliert. Während dieses Vorgangs werden die DNS-Einträge ElastiCache aktualisiert, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei Version 4.0.10 und niedriger stellen Sie bei vorherigen Versionen möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten fest, während der DNS-Eintrag aktualisiert wird.

Die Lesevorgänge aus den Read Replica-Clustern werden jedoch ohne Unterbrechung fortgesetzt.

Die Zeit, die benötigt wird, um auf einen kleineren Knotentyp herunterzuskalieren, hängt von Ihrem Knotentyp und der Datenmenge in Ihrem aktuellen Cluster ab.

So verkleinern Sie eine Valkey- oder Redis OSS-Replikationsgruppe ()AWS CLI

1. Ermitteln Sie, auf welche Knotentypen Sie herunterskalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.

- `--replication-group-id` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
  --replication-group-id my-repl-group
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
  --replication-group-id my-repl-group
```

Die Ausgabe dieser Operation sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
  "ScaleDownModifications": [
    "cache.m3.2xlarge",
    "cache.m3.large",
    "cache.m3.xlarge",
    "cache.m4.10xlarge",
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
```

```
        "cache.r3.large",  
        "cache.r3.xlarge"  
    ]  
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-modifications](#) in der AWS CLI-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe des `AWS CLI modify-replication-group` Befehls mit den folgenden Parametern auf den neuen Knotentyp.
 - `--replication-group-id` der Name der Replikationsgruppe.
 - `--cache-node-type`— der neue, kleinere Knotentyp der Cluster in dieser Replikationsgruppe. Dieser Wert muss einer der Instance-Typen sein, die vom `list-allowed-node-type-modifications` Befehl im vorherigen Schritt zurückgegeben wurden.
 - `--cache-parameter-group-name` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
 - `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `--no-apply-immediately`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
    --replication-group-id my-repl-group \  
    --cache-node-type cache.t2.small \  
    --cache-parameter-group-name redis32-m3-2x1 \  
    --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
    --replication-group-id my-repl-group ^  
    --cache-node-type cache.t2.small ^  
    --cache-parameter-group-name redis32-m3-2x1 \  
    --apply-immediately
```



```
--apply-immediately
```

Die Ausgabe dieses Befehls sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
  "ReplicationGroup": {
    "Status": "available",
    "Description": "Some description",
    "NodeGroups": [
      {
        "Status": "available",
        "NodeGroupMembers": [
          {
            "CurrentRole": "primary",
            "PreferredAvailabilityZone": "us-west-2b",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
              "Port": 6379,
              "Address": "my-repl-
group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
            },
            "CacheClusterId": "my-repl-group-001"
          },
          {
            "CurrentRole": "replica",
            "PreferredAvailabilityZone": "us-west-2c",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
              "Port": 6379,
              "Address": "my-repl-
group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
            },
            "CacheClusterId": "my-repl-group-002"
          }
        ],
        "NodeGroupId": "0001",
        "PrimaryEndpoint": {
          "Port": 6379,
          "Address": "my-repl-
group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
        }
      }
    ],
  },
}
```

```

    "ReplicationGroupId": "my-repl-group",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "disabled",
    "SnapshotWindow": "12:00-13:00",
    "SnapshottingClusterId": "my-repl-group-002",
    "MemberClusters": [
        "my-repl-group-001",
        "my-repl-group-002",
    ],
    "PendingModifiedValues": {}
}

```

Weitere Informationen finden Sie unter [modify-replication-group](#) in der AWS CLI-Referenz.

3. Wenn Sie den `--apply-immediately` Parameter verwendet haben, überwachen Sie den Status der Replikationsgruppe mithilfe des `AWS CLI describe-replication-group` Befehls mit dem folgenden Parameter. Wenn der Status von In Bearbeitung zu Verfügbar wechselt, können Sie die neue, abwärts skalierte Replikationsgruppe verwenden.
 - `--replication-group-id` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

Für Linux, macOS oder Unix:

```

aws elasticache describe-replication-group \
    --replication-group-id my-replication-group

```

Für Windows:

```

aws elasticache describe-replication-groups ^
    --replication-group-id my-replication-group

```

Weitere Informationen finden Sie unter [describe-replication-groups](#) in der AWS CLI-Referenz.

Herunterskalierung einer Valkey- oder Redis OSS-Replikationsgruppe (API) ElastiCache

Der folgende Prozess skaliert Ihre Replikationsgruppe mithilfe der API von ihrem aktuellen Knotentyp auf einen neuen, kleineren Knotentyp. ElastiCache Während dieses Vorgangs werden die DNS-

Einträge ElastiCache aktualisiert, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei Redis OSS Version 4.0.10 und niedriger stellen Sie möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der DNS-Eintrag aktualisiert wird. Die Lesevorgänge aus den Read Replica-Clustern werden jedoch ohne Unterbrechung fortgesetzt.

Die Zeit, die benötigt wird, um auf einen kleineren Knotentyp herunterzuskalieren, hängt von Ihrem Knotentyp und der Datenmenge in Ihrem aktuellen Cluster ab.

So verkleinern Sie eine Valkey- oder Redis OSS-Replikationsgruppe (API) ElastiCache

1. Ermitteln Sie, auf welche Knotentypen Sie mithilfe der ElastiCache `ListAllowedNodeTypeModifications` API-Aktion mit dem folgenden Parameter herunterskalieren können.
 - `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&ReplicationGroupId=MyReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [ListAllowedNodeTypeModifications](#) in der Amazon ElastiCache API-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe der `ModifyReplicationGroup` ElastiCache API-Aktion und mit den folgenden Parametern auf den neuen Knotentyp.
 - `ReplicationGroupId` der Name der Replikationsgruppe.
 - `CacheNodeType`— der neue, kleinere Knotentyp der Cluster in dieser Replikationsgruppe. Dieser Wert muss einer der Instance-Typen sein, die von der

ListAllowedNodeTypeModifications Aktion im vorherigen Schritt zurückgegeben wurden.

- **CacheParameterGroupName** – [Optional) Verwenden Sie diesen Parameter, wenn Sie **reserved-memory** verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie **reserved-memory-percent** verwenden, können Sie diesen Parameter weglassen.
- **ApplyImmediately** Legen Sie **true** fest, um die Skalierung sofort auszuführen. Wenn Sie die Abwärtsskalierung für das nächste Wartungsfenster planen möchten, verwenden Sie **ApplyImmediately=false**.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup  
  &ApplyImmediately=true  
  &CacheNodeType=cache.m3.2xlarge  
  &CacheParameterGroupName=redis32-m3-2x1  
  &ReplicationGroupId=myReplGroup  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie [ModifyReplicationGroup](#) in der Amazon ElastiCache API-Referenz.

3. Falls Sie dies verwendet haben **ApplyImmediately=true**, überwachen Sie den Status der Replikationsgruppe mithilfe der **ElastiCache DescribeReplicationGroups** API-Aktion mit den folgenden Parametern. Wenn der Status von **In Bearbeitung** zu **Verfügbar** wechselt, können Sie die neue, abwärts skalierte Replikationsgruppe verwenden.
- **ReplicationGroupId** der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeReplicationGroups  
  &ReplicationGroupId=MyReplGroup  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [DescribeReplicationGroups](#) in der Amazon ElastiCache API-Referenz.

Erhöhen der Lesekapazität

Um die Lesekapazität zu erhöhen, fügen Sie Read Replicas (bis zu maximal fünf) zu Ihrer Valkey- oder Redis OSS-Replikationsgruppe hinzu.

Sie können die Lesekapazität Ihres Valkey- oder Redis OSS-Clusters mithilfe der ElastiCache Konsole, der oder der API skalieren. AWS CLI ElastiCache Weitere Informationen finden Sie unter [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

Verringern der Lesekapazität

Um die Lesekapazität zu verringern, löschen Sie eine oder mehrere Read Replicas aus Ihrem Valkey- oder Redis OSS-Cluster mit Replikaten (in der API/CLI als Replikationsgruppe bezeichnet). Wenn es sich bei dem Cluster um Multi-AZ mit aktiviertem automatischem Failover handelt, können Sie das letzte Read Replica nicht löschen, ohne zuvor Multi-AZ zu deaktivieren. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

Weitere Informationen finden Sie unter [Löschen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

Skalierung von Valkey- oder Redis OSS-Clustern (Clustermodus aktiviert)

Wenn sich die Nachfrage nach Ihren Clustern ändert, können Sie entscheiden, die Leistung zu verbessern oder die Kosten zu senken, indem Sie die Anzahl der Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) ändern. Wir empfehlen, dazu die horizontale Online-Skalierung zu verwenden, da Ihr Cluster während des Skalierungsprozesses weiterhin Anforderungen bedienen kann.

Zu den Bedingungen, unter denen Sie den Cluster möglicherweise neu skalieren, gehören folgende:

- Speicherbelastung

Wenn die Knoten in Ihrem Cluster einer hohen Speicherbelastung ausgesetzt sind, können Sie sich für eine Skalierung entscheiden, sodass Sie über mehr Ressourcen verfügen, um Daten besser speichern und Anforderungen verarbeiten zu können.

Sie können feststellen, ob Ihre Knoten unter Speicherauslastung stehen, indem Sie die folgenden Messwerte überwachen: `FreeableMemory`, und `SwapUsage`. `BytesUsedForCache`

- CPU- oder Netzwerkengpässe:

Wenn latency/throughput Probleme in Ihrem Cluster auftreten, müssen Sie möglicherweise eine Skalierung vornehmen, um die Probleme zu lösen.

Sie können Ihre Latenz und Ihren Durchsatz überwachen, indem Sie die folgenden Messwerte überwachen: `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOutCurrConnections`, und `NewConnections`

- Ihr Cluster ist übermäßig skaliert:

Der aktuelle Bedarf an Ihrem Cluster ist so hoch, dass eine Skalierung die Leistung nicht beeinträchtigt und Ihre Kosten reduziert.

Sie können die Nutzung Ihres Clusters überwachen, um anhand der folgenden Metriken festzustellen, ob Sie sicher skalieren können oder nicht: `FreeableMemorySwapUsageBytesUsedForCache`, `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, und `NewConnections`.

Leistungsbeeinträchtigung durch Skalierung

Wenn Sie den Offline-Prozess skalieren, ist Ihr Cluster für einen erheblichen Teil des Prozesses offline und kann daher keine Anforderungen erfüllen. Wenn Sie mithilfe der Onlinemethode skalieren, da die Skalierung eine rechenintensive Operation ist, gibt es eine Leistungseinbuße. Ihr Cluster führt jedoch weiterhin während der Skalierungsoperation Anforderungen aus. Wie stark die Verschlechterung ist, hängt von Ihrer normalen CPU-Auslastung und Ihren Daten ab.

Es gibt zwei Möglichkeiten, Ihren Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) zu skalieren: horizontale und vertikale Skalierung.

- Mit horizontaler Skalierung ändern Sie die Anzahl der Knotengruppen (Shards) in der Replikationsgruppe, indem Sie Knotengruppen hinzufügen oder entfernen. Der Online-Resharding-Prozess ermöglicht die Skalierung, in/out während der Cluster weiterhin eingehende Anfragen bearbeitet.

Konfigurieren Sie die Slots in Ihrem neuen Cluster anders als im alten Cluster. Nur Offline-Methode.

- Vertikale Skalierung – Ändern Sie den Knotentyp, um die Größe des Clusters anzupassen. Die vertikale Online-Skalierung ermöglicht die Skalierung up/down, während der Cluster weiterhin eingehende Anfragen bearbeitet.

Wenn Sie die Größe und Speicherkapazität des Clusters reduzieren, indem Sie entweder ein- oder herunterskalieren, stellen Sie sicher, dass die neue Konfiguration über ausreichend Speicher für Ihre Daten und den OSS-Overhead von Valkey oder Redis verfügt.

Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Inhalt

- [Offline-Resharding für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
- [Online-Resharding für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
 - [Hinzufügen von Shards mit Online-Resharding](#)
 - [Entfernen von Shards mit Online-Resharding](#)
 - [Entfernen von Shards \(Konsole\)](#)
 - [Entfernen von Shards \(AWS CLI\)](#)
 - [Shards entfernen \(API\) ElastiCache](#)
 - [Online-Shard-Rebalancing](#)
 - [Online-Shard-Rebalancing \(Console\)](#)

- [Online-Shard-Rebalancing \(AWS CLI\)](#)
- [Online-Shard-Rebalancing \(API\) ElastiCache](#)
- [Vertikales Online-Skalieren durch Ändern des Knotentyps](#)
 - [Online-aufwärtsskalieren](#)
 - [Skalierung von Valkey- oder Redis OSS-Clustern \(Konsole\)](#)
 - [Skalierung von Valkey- oder Redis OSS-Clustern \(\)AWS CLI](#)
 - [Skalierung von Valkey- oder Redis OSS-Clustern \(API\) ElastiCache](#)
 - [Online-abwärtsskalieren](#)
 - [Herunterskalierung von Valkey- oder Redis OSS-Clustern \(Konsole\)](#)
 - [Herunterskalieren von Valkey- oder Redis OSS-Clustern \(\)AWS CLI](#)
 - [Herunterskalierung der Valkey- oder Redis OSS-Cluster \(API\) ElastiCache](#)

Offline-Resharding für Valkey oder Redis OSS (Clustermodus aktiviert)

Der Hauptvorteil der Offline-Shard-Rekonfiguration besteht darin, dass Sie mehr als nur das Hinzufügen oder Entfernen von Shards aus Ihrer Replikationsgruppe tun können. Wenn Sie Resharding und Rebalancing offline durchführen, können Sie nicht nur die Anzahl der Shards in Ihrer Replikationsgruppe ändern, sondern auch Folgendes tun:

 Note

Offline-Resharding wird auf Valkey- oder Redis-OSS-Clustern mit aktiviertem Data Tiering nicht unterstützt. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

- Ändern Sie den Knotentyp Ihrer Replikationsgruppe.
- Geben Sie die Availability Zone für jeden Knoten in der Replikationsgruppe an.
- Upgrade auf eine neuere Engine-Version.
- Geben Sie die Anzahl der Replikatknoten in jeder Shard unabhängig voneinander an.
- Geben Sie den Keyspace für jede Shard an.

Der Hauptnachteil der Offline-Shard-Neukonfiguration besteht darin, dass Ihr Cluster offline ist und mit dem Wiederherstellungsabschnitt des Prozesses beginnt und so lange fortfährt, bis Sie die

Endpunkte in Ihrer Anwendung aktualisieren. Die Dauer, in der Ihr Cluster offline ist, hängt von der Datenmenge in Ihrem Cluster ab.

Um Ihre Shards Valkey oder Redis OSS-Cluster (Clustermodus aktiviert) offline neu zu konfigurieren

1. Erstellen Sie ein manuelles Backup Ihres vorhandenen Valkey- oder Redis OSS-Clusters. Weitere Informationen finden Sie unter [Erstellen manueller Backups](#).
2. Erstellen Sie einen neuen Cluster, indem Sie sie aus der Sicherung wiederherstellen. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).
3. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).

Online-Resharding für Valkey oder Redis OSS (Clustermodus aktiviert)

Mithilfe von Online-Resharding und Shard-Rebalancing mit ElastiCache Valkey 7.2 oder neuer oder Redis OSS Version 3.2.10 oder neuer können Sie Ihren Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) dynamisch und ohne Ausfallzeiten skalieren. Dieser Ansatz bedeutet, dass Ihr Cluster weiterhin Anfragen bearbeiten kann, selbst wenn die Skalierung oder das Rebalancing in Bearbeitung ist.

Sie haben die folgenden Möglichkeiten:

- Skalierung — Erhöhen Sie die Lese- und Schreibkapazität, indem Sie Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) (Replikationsgruppe) Shards (Knotengruppen) hinzufügen.

Wenn Sie Ihrer Replikationsgruppe einen oder mehrere Shards hinzufügen, entspricht die Anzahl der Knoten in jedem neuen Shard der Anzahl der Knoten in der kleinsten der vorhandenen Shards.

- Skalierung — Reduzieren Sie die Lese- und Schreibkapazität und damit die Kosten, indem Sie Shards aus Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) entfernen.
- Rebalance — Verschieben Sie die Keyspaces zwischen den Shards in Ihrem Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert), sodass sie so gleichmäßig wie möglich auf die Shards verteilt sind.

Sie haben nicht die folgenden Möglichkeiten:

- Konfigurieren Sie Shards unabhängig:

Sie können den Keyspace für Shards nicht unabhängig angeben. Um dies zu tun, müssen Sie den Offline-Prozess verwenden.

Derzeit gelten die folgenden Einschränkungen für Online-Resharding und Rebalancing: ElastiCache

- Diese Prozesse erfordern Valkey 7.2 und neuer oder Redis OSS 3.2.10 oder neuer. Informationen zum Aktualisieren der Engine-Version finden Sie unter [Versionsverwaltung für ElastiCache](#).
- Es gibt Einschränkungen bei Slots oder Keyspaces und großen Elementen:

Wenn ein Schlüssel in einem Shard ein großes Element enthält, wird dieser Schlüssel beim horizontalen Skalieren oder Neuausrichten nicht in eine neue Shard migriert. Diese Funktionalität kann zu unsymmetrischen Shards führen.

Wenn einer der Schlüssel in einer Shard ein großes Element enthält (Elemente größer als 256 MB nach der Serialisierung), wird diese Shard beim Skalieren nicht gelöscht. Diese Funktionalität kann dazu führen, dass einige Shards nicht gelöscht werden.

- Beim Ausskalieren entspricht die Anzahl der Knoten in neuen Shards der Anzahl der Knoten in der kleinsten vorhandenen Shard.
- Beim Ausskalieren werden alle Tags, die allen vorhandenen Shards gemeinsam sind, in die neuen Shards kopiert.
- Beim Skalieren eines Global Data Store-Clusters ElastiCache werden Funktionen nicht automatisch von einem der vorhandenen Knoten auf die neuen Knoten repliziert. Wir empfehlen, Ihre Funktionen nach der Aufskalierung Ihres Clusters in den (die) neuen Shard(s) zu laden, sodass alle Shards dieselben Funktionen aufweisen.

Note

ElastiCache Für Valkey 7.2 und höher und ElastiCache für Redis OSS Version 7 und höher: Wenn Sie Ihren Cluster skalieren, repliziert er ElastiCache automatisch die Funktionen, die in einem der vorhandenen Knoten geladen sind (zufällig ausgewählt), auf die neuen Knoten. Wenn Ihre Anwendung [Functions](#) verwendet, empfehlen wir, alle Ihre Funktionen vor dem Skalieren auf alle Shards zu laden, damit Ihr Cluster nicht mit unterschiedlichen Funktionsdefinitionen auf verschiedenen Shards endet.

Weitere Informationen finden Sie unter [Online-Größenanpassung von Clustern](#).

Sie können Ihre Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mithilfe der API AWS-Managementkonsole, der und der API horizontal skalieren oder neu ausbalancieren AWS CLI. ElastiCache

Hinzufügen von Shards mit Online-Resharding

Sie können Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mithilfe der API, oder Shards hinzufügen. AWS-Managementkonsole AWS CLI ElastiCache Wenn Sie Shards zu einem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) hinzufügen, werden alle Tags auf den vorhandenen Shards auf die neuen Shards kopiert.

Hinzufügen von Shards (Konsole)

Sie können den verwenden AWS-Managementkonsole, um Ihrem Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) einen oder mehrere Shards hinzuzufügen. Das folgende Verfahren beschreibt den Prozess.

Um Shards zu Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) hinzuzufügen

1. Öffnen Sie die Konsole unter ElastiCache . <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.
3. Suchen Sie den Namen des Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert), dem Sie Shards hinzufügen möchten, und wählen Sie ihn aus, nicht das Feld links neben dem Clusternamen.



Tip

Valkey oder Redis OSS (Clustermodus aktiviert) zeigen Clustered Valkey oder Clustered Redis OSS in der Modus-Spalte

4. Wählen Sie Shard hinzufügen aus.
 - a. Wählen Sie für Anzahl der hinzuzufügenden Shards die Anzahl der Shards, die Sie diesem Cluster hinzufügen möchten.
 - b. Für Availability zone(s), entweder No preference oder Specify availability zones angeben.

- c. Wenn Sie Availability Zones angeben für jeden Knoten angeben möchten, wählen Sie für jeden Knoten in jeder Shard die Availability Zone des Knotens aus der Liste der Availability Zones.
- d. Wählen Sie Hinzufügen aus.

Shards hinzufügen (AWS CLI)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mit dem hinzufügen.AWS CLI

Verwenden Sie die folgenden Parameter mit `modify-replication-group-shard-configuration`.

Parameters

- `--apply-immediately` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.
- `--replication-group-id` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `--node-group-count` – Erforderlich. Gibt die Anzahl der Shards (Knotengruppen) an, die nach Abschluss des Vorgangs vorhanden sein müssen. Beim Hinzufügen von Shards muss der Wert von `--node-group-count` größer als die aktuelle Anzahl von Shards sein.

Optional können Sie die Availability Zone für jeden Knoten in der Replikationsgruppe mit `--resharding-configuration` angeben.

- `--resharding-configuration` – Optional. Eine Liste der bevorzugten Availability Zones für jeden Knoten in jedem Shard in der Replikationsgruppe. Verwenden Sie diesen Parameter nur, wenn der Wert von `--node-group-count` größer ist als die aktuelle Anzahl von Shards. Wenn dieser Parameter beim Hinzufügen von Shards weggelassen wird, ElastiCache wählt Amazon die Availability Zones für die neuen Knoten aus.

Im folgenden Beispiel werden die Keyspaces für vier Shards in einem Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) mit dem Namen neu konfiguriert. `my-cluster` Das Beispiel gibt auch die Availability Zone für jeden Knoten in jedem Shard an. Die Operation beginnt sofort.

Example- Shards hinzufügen

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id my-cluster \  
  --node-group-count 4 \  
  --resharding-configuration \  
    "PreferredAvailabilityZones=us-east-2a,us-east-2c" \  
    "PreferredAvailabilityZones=us-east-2b,us-east-2a" \  
    "PreferredAvailabilityZones=us-east-2c,us-east-2d" \  
    "PreferredAvailabilityZones=us-east-2d,us-east-2c" \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group-shard-configuration ^  
  --replication-group-id my-cluster ^  
  --node-group-count 4 ^  
  --resharding-configuration ^  
    "PreferredAvailabilityZones=us-east-2a,us-east-2c" ^  
    "PreferredAvailabilityZones=us-east-2b,us-east-2a" ^  
    "PreferredAvailabilityZones=us-east-2c,us-east-2d" ^  
    "PreferredAvailabilityZones=us-east-2d,us-east-2c" ^  
  --apply-immediately
```

[Weitere Informationen finden Sie unter -configuration in der Dokumentationmodify-replication-group-shard.](#) AWS CLI

Hinzufügen von Shards (API) ElastiCache

Sie können die ElastiCache API verwenden, um die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) online neu zu konfigurieren, indem Sie den Vorgang verwenden. `ModifyReplicationGroupShardConfiguration`

Verwenden Sie die folgenden Parameter mit `ModifyReplicationGroupShardConfiguration`.

Parameters

- `ApplyImmediately=true` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.
- `ReplicationGroupId` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.

- **NodeGroupCount** – Erforderlich. Gibt die Anzahl der Shards (Knotengruppen) an, die nach Abschluss des Vorgangs vorhanden sein müssen. Beim Hinzufügen von Shards muss der Wert von **NodeGroupCount** größer als die aktuelle Anzahl von Shards sein.

Optional können Sie die Availability Zone für jeden Knoten in der Replikationsgruppe mit **ReshardingConfiguration** angeben.

- **ReshardingConfiguration** – Optional. Eine Liste der bevorzugten Availability Zones für jeden Knoten in jedem Shard in der Replikationsgruppe. Verwenden Sie diesen Parameter nur, wenn der Wert von **NodeGroupCount** größer ist als die aktuelle Anzahl von Shards. Wenn dieser Parameter beim Hinzufügen von Shards weggelassen wird, ElastiCache wählt Amazon die Availability Zones für die neuen Knoten aus.

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Cluster-Modus aktiviert) neu konfigurieren, indem Sie Shards mithilfe der API hinzufügen.

ElastiCache

Example- Shards hinzufügen

Im folgenden Beispiel werden dem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) Knotengruppen hinzugefügt `my-cluster`, sodass nach Abschluss des Vorgangs insgesamt vier Knotengruppen vorhanden sind. Das Beispiel gibt auch die Availability Zone für jeden Knoten in jedem Shard an. Die Operation beginnt sofort.

```
https://elasticache.us-east-2.amazonaws.com/
  ?Action=ModifyReplicationGroupShardConfiguration
  &ApplyImmediately=true
  &NodeGroupCount=4
  &ReplicationGroupId=my-cluster

  &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone
east-2a

  &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone
east-2c

  &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone
east-2b

  &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone
east-2a
```



```
&ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone
east-2c
```

```
&ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone
east-2d
```

```
&ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone
east-2d
```

```
&ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone
east-2c
```

```
&Version=2015-02-02
```

```
&SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256
```

```
&Timestamp=20171002T192317Z
```

```
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [ModifyReplicationGroupShardConfiguration](#) in der ElastiCache API-Referenz.

Entfernen von Shards mit Online-Resharding

Sie können Shards mithilfe der API, oder aus Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) entfernen. [AWS-Managementkonsole](#) [AWS CLI](#) [ElastiCache](#)

Themen

- [Entfernen von Shards \(Konsole\)](#)
- [Entfernen von Shards \(AWS CLI\)](#)
- [Shards entfernen \(API\) ElastiCache](#)

Entfernen von Shards (Konsole)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mit dem entfernen. [AWS-Managementkonsole](#)

Bevor Sie Knotengruppen (Shards) aus Ihrer Replikationsgruppe entfernen, stellen Sie sicher, ElastiCache dass alle Ihre Daten in die verbleibenden Shards passen. Wenn die Daten passen, werden die angegebenen Shards wie angefordert aus der Replikationsgruppe gelöscht. Wenn

die Daten nicht in die verbleibenden Knotengruppen passen, wird der Prozess beendet und die Replikationsgruppe hat dieselbe Knotengruppenkonfiguration wie vor dem Zeitpunkt, an dem die Anforderung erstellt wurde.

Sie können den verwenden AWS-Managementkonsole, um einen oder mehrere Shards aus Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) zu entfernen. Sie können nicht alle Shards in einer Replikationsgruppe entfernen. Stattdessen müssen Sie die Replikationsgruppe löschen. Weitere Informationen finden Sie unter [Löschen einer Replikationsgruppe](#). Das folgende Verfahren beschreibt den Vorgang zum Löschen eines oder mehrerer Shards.

Um Shards aus Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) zu entfernen

1. Öffnen Sie die Konsole unter ElastiCache . <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.
3. Suchen Sie den Namen des Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert), aus dem Sie Shards entfernen möchten, und wählen Sie ihn aus, nicht das Feld links neben dem Clusternamen.

 Tip

Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) haben in der Spalte Shards einen Wert von 1 oder höher.

4. Wählen Sie in der Liste der Shards das Feld links von den Namen aller Shards, die Sie löschen möchten.
5. Wählen Sie Delete Shard.

Entfernen von Shards (AWS CLI)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mit dem entfernen.AWS CLI

 Important

Bevor Sie Knotengruppen (Shards) aus Ihrer Replikationsgruppe entfernen, stellen Sie sicher, ElastiCache dass alle Ihre Daten in die verbleibenden Shards passen. Wenn die Daten passen, werden die angegebenen Shards (`--node-groups-to-remove`) wie angefordert aus der Replikationsgruppe gelöscht und ihre Keyspaces den verbleibenden

Shards zugeordnet. Wenn die Daten nicht in die verbleibenden Knotengruppen passen, wird der Prozess beendet und die Replikationsgruppe hat dieselbe Knotengruppenkonfiguration wie vor dem Zeitpunkt, an dem die Anforderung erstellt wurde.

Sie können den verwenden AWS CLI, um einen oder mehrere Shards aus Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) zu entfernen. Sie können nicht alle Shards in einer Replikationsgruppe entfernen. Stattdessen müssen Sie die Replikationsgruppe löschen. Weitere Informationen finden Sie unter [Löschen einer Replikationsgruppe](#).

Verwenden Sie die folgenden Parameter mit `modify-replication-group-shard-configuration`.

Parameters

- `--apply-immediately` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.
- `--replication-group-id` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `--node-group-count` – Erforderlich. Gibt die Anzahl der Shards (Knotengruppen) an, die nach Abschluss des Vorgangs vorhanden sein müssen. Beim Entfernen von Shards muss der Wert von `--node-group-count` kleiner als die aktuelle Anzahl von Shards sein.
- `--node-groups-to-remove` Erforderlich, wenn `--node-group-count` kleiner als die aktuelle Anzahl der Knotengruppen (Shards) ist. Eine Liste von Shards (Knotengruppe), die aus der Replikationsgruppe entfernt IDs werden sollen.

Das folgende Verfahren beschreibt den Vorgang zum Löschen eines oder mehrerer Shards.

Example- Entfernen von Shards

Im folgenden Beispiel werden zwei Knotengruppen aus dem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) entfernt `my-cluster`, sodass nach Abschluss des Vorgangs insgesamt zwei Knotengruppen vorhanden sind. Die Keyspaces der entfernten Shards werden gleichmäßig auf die verbleibenden Shards verteilt.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group-shard-configuration \
```

```
--replication-group-id my-cluster \  
--node-group-count 2 \  
--node-groups-to-remove "0002" "0003" \  
--apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group-shard-configuration ^  
--replication-group-id my-cluster ^  
--node-group-count 2 ^  
--node-groups-to-remove "0002" "0003" ^  
--apply-immediately
```

Shards entfernen (API) ElastiCache

Sie können die ElastiCache API verwenden, um die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) online neu zu konfigurieren, indem Sie den Vorgang verwenden. `ModifyReplicationGroupShardConfiguration`

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mithilfe der API entfernen. ElastiCache

Important

Bevor Sie Knotengruppen (Shards) aus Ihrer Replikationsgruppe entfernen, stellen Sie sicher, ElastiCache dass alle Ihre Daten in die verbleibenden Shards passen. Wenn die Daten passen, werden die angegebenen Shards (`NodeGroupsToRemove`) wie angefordert aus der Replikationsgruppe gelöscht und ihre Keyspaces den verbleibenden Shards zugeordnet. Wenn die Daten nicht in die verbleibenden Knotengruppen passen, wird der Prozess beendet und die Replikationsgruppe hat dieselbe Knotengruppenkonfiguration wie vor dem Zeitpunkt, an dem die Anforderung erstellt wurde.

Sie können die ElastiCache API verwenden, um einen oder mehrere Shards aus Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) zu entfernen. Sie können nicht alle Shards in einer Replikationsgruppe entfernen. Stattdessen müssen Sie die Replikationsgruppe löschen. Weitere Informationen finden Sie unter [Löschen einer Replikationsgruppe](#).

Verwenden Sie die folgenden Parameter mit `ModifyReplicationGroupShardConfiguration`.

Parameters

- `ApplyImmediately=true` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.
- `ReplicationGroupId` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `NodeGroupCount` – Erforderlich. Gibt die Anzahl der Shards (Knotengruppen) an, die nach Abschluss des Vorgangs vorhanden sein müssen. Beim Entfernen von Shards muss der Wert von `NodeGroupCount` kleiner als die aktuelle Anzahl von Shards sein.
- `NodeGroupsToRemove` Erforderlich, wenn `--node-group-count` kleiner als die aktuelle Anzahl der Knotengruppen (Shards) ist. Eine Liste von Shards (Knotengruppe), die aus der Replikationsgruppe entfernt IDs werden sollen.

Das folgende Verfahren beschreibt den Vorgang zum Löschen eines oder mehrerer Shards.

Example- Entfernen von Shards

Im folgenden Beispiel werden zwei Knotengruppen aus dem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) entfernt `my-cluster`, sodass nach Abschluss des Vorgangs insgesamt zwei Knotengruppen vorhanden sind. Die Keyspaces der entfernten Shards werden gleichmäßig auf die verbleibenden Shards verteilt.

```
https://elasticache.us-east-2.amazonaws.com/  
?Action=ModifyReplicationGroupShardConfiguration  
&ApplyImmediately=true  
&NodeGroupCount=2  
&ReplicationGroupId=my-cluster  
&NodeGroupsToRemove.member.1=0002  
&NodeGroupsToRemove.member.2=0003  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20171002T192317Z  
&X-Amz-Credential=<credential>
```

Online-Shard-Rebalancing

Sie können Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mithilfe der API, oder neu verteilen. AWS-Managementkonsole AWS CLI ElastiCache

Themen

- [Online-Shard-Rebalancing \(Console\)](#)
- [Online-Shard-Rebalancing \(AWS CLI\)](#)
- [Online-Shard-Rebalancing \(API\) ElastiCache](#)

Online-Shard-Rebalancing (Console)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu konfigurieren, indem Sie die Shards mithilfe von neu verteilen.AWS-Managementkonsole

Um die Schlüsselräume zwischen den Shards auf Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu auszurichten

1. Öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.
3. Wählen Sie den Namen des Valkey- oder Redis OSS-Clusters (Clustermodus aktiviert), den Sie neu verteilen möchten, und nicht das Feld links neben dem Namen.

Tip

Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) haben in der Spalte Shards einen Wert von 1 oder höher.

4. Wählen Sie Rebalance aus.
5. Wählen Sie nach Aufforderung die Option Rebalance aus. Möglicherweise wird eine Meldung ähnlich der folgenden angezeigt: *Slots in the replication group are uniformly distributed. Nothing to do. (Service: AmazonElastiCache; Status Code: 400; Error Code: InvalidReplicationGroupState; Request ID: 2246cebd-9721-11e7-8d5b-e1b0f086c8cf)* Wählen Sie Cancel wenn Sie das tun.

Online-Shard-Rebalancing (AWS CLI)

Verwenden Sie die folgenden Parameter mit `modify-replication-group-shard-configuration`.

Parameters

- `--apply-immediately` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.
- `--replication-group-id` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `--node-group-count` – Erforderlich. Um die Keyspaces über alle Shards im Cluster neu zu verteilen, muss dieser Wert der aktuellen Anzahl von Shards entsprechen.

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu konfigurieren, indem Sie die Shards mithilfe von neu verteilen.AWS CLI

Example- Ausgleichen der Shards in einem Cluster

Im folgenden Beispiel werden die Steckplätze im Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu verteilt, sodass die Steckplätze `my-cluster` so gleichmäßig wie möglich verteilt werden. Der Wert von `--node-group-count` (4) ist die Anzahl der derzeit im Cluster befindlichen Shards.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id my-cluster \  
  --node-group-count 4 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group-shard-configuration ^  
  --replication-group-id my-cluster ^  
  --node-group-count 4 ^  
  --apply-immediately
```

Online-Shard-Rebalancing (API) ElastiCache

Sie können die ElastiCache API verwenden, um die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) online neu zu konfigurieren, indem Sie den Vorgang verwenden. `ModifyReplicationGroupShardConfiguration`

Verwenden Sie die folgenden Parameter mit `ModifyReplicationGroupShardConfiguration`.

Parameters

- `ApplyImmediately=true` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.
- `ReplicationGroupId` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `NodeGroupCount` – Erforderlich. Um die Keyspaces über alle Shards im Cluster neu zu verteilen, muss dieser Wert der aktuellen Anzahl von Shards entsprechen.

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu konfigurieren, indem Sie die Shards mithilfe der API neu verteilen.
ElastiCache

Example- Cluster-Ausgleich

Im folgenden Beispiel werden die Steckplätze im Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) neu verteilt, sodass die Steckplätze `my-cluster` so gleichmäßig wie möglich verteilt werden. Der Wert von `NodeGroupCount` (4) ist die Anzahl der derzeit im Cluster befindlichen Shards.

```
https://elasticache.us-east-2.amazonaws.com/  
?Action=ModifyReplicationGroupShardConfiguration  
&ApplyImmediately=true  
&NodeGroupCount=4  
&ReplicationGroupId=my-cluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20171002T192317Z  
&X-Amz-Credential=<credential>
```

Vertikales Online-Skalieren durch Ändern des Knotentyps

Durch die Verwendung der vertikalen Online-Skalierung mit Valkey Version 7.2 oder neuer oder Redis OSS Version 3.2.10 oder neuer können Sie Ihre Valkey- oder Redis OSS-Cluster dynamisch mit minimalen Ausfallzeiten skalieren. Dadurch kann Ihr Valkey- oder Redis OSS-Cluster Anfragen auch während der Skalierung bearbeiten.

 Note

Die Skalierung zwischen einem Cluster mit Daten-Tiering (z. B. ein Cluster, der einen R6gd-Knotentyp verwendet) und einem Cluster ohne Daten-Tiering (z. B. ein Cluster, der einen R6g-Knotentyp verwendet) wird nicht unterstützt. Weitere Informationen finden Sie unter [Datenzuweisung ElastiCache](#).

Sie haben die folgenden Möglichkeiten:

- Skalieren — Erhöhen Sie die Lese- und Schreibkapazität, indem Sie den Knotentyp Ihres Valkey- oder Redis OSS-Clusters so anpassen, dass ein größerer Knotentyp verwendet wird.

ElastiCache passt die Größe Ihres Clusters dynamisch an, bleibt aber online und bearbeitet Anfragen.

- Herunterskalierung – Verringern Sie die Lese- und Schreibkapazität, indem Sie den Knotentyp auf die Verwendung eines kleineren Knotens anpassen. Auch hier wird die Größe Ihres Clusters ElastiCache dynamisch angepasst, während er online bleibt und Anfragen bearbeitet. In diesem Fall verringern Sie die Kosten durch die Verkleinerung des Knotens.

 Note

Aufwärts- und Abwärtsskalieren basiert auf der Erstellung von neu ausgewählten Knotentypen und der Synchronisierung der neuen Knoten mit den vorherigen. Gehen Sie wie folgt vor, um einen reibungslosen up/down Ablauf der Skalierung sicherzustellen:

- Stellen Sie sicher, dass die Elastic Network-Schnittstelle (ENI) über ausreichend Kapazität verfügt. Vergewissern Sie sich beim Abwärtsskalieren, dass der kleinere Knoten über ausreichend Speicher für erwarteten Datenverkehr verfügt.

Erfahren Sie mehr über bewährte Methoden für die Speicherverwaltung unter [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#).

- Während die vertikale Skalierung ausgelegt ist, vollständig online zu bleiben, basiert sie auf der Synchronisierung von Daten zwischen dem alten und dem neuen Knoten. Wir empfehlen, dass Sie Abwärts-/Aufwärtsskalieren zu einem Zeitpunkt durchführen, an dem der Datenverkehr am geringsten ist.

- Testen Sie das Verhalten Ihrer Anwendung während der Skalierung möglichst in einer Staging-Umgebung.

Inhalt

- [Online-aufwärtsskalieren](#)
 - [Skalierung von Valkey- oder Redis OSS-Clustern \(Konsole\)](#)
 - [Skalierung von Valkey- oder Redis OSS-Clustern \(\)AWS CLI](#)
 - [Skalierung von Valkey- oder Redis OSS-Clustern \(API\) ElastiCache](#)
- [Online-abwärtsskalieren](#)
 - [Herunterskalierung von Valkey- oder Redis OSS-Clustern \(Konsole\)](#)
 - [Herunterskalieren von Valkey- oder Redis OSS-Clustern \(\)AWS CLI](#)
 - [Herunterskalierung der Valkey- oder Redis OSS-Cluster \(API\) ElastiCache](#)

Online-aufwärtsskalieren

Themen

- [Skalierung von Valkey- oder Redis OSS-Clustern \(Konsole\)](#)
- [Skalierung von Valkey- oder Redis OSS-Clustern \(\)AWS CLI](#)
- [Skalierung von Valkey- oder Redis OSS-Clustern \(API\) ElastiCache](#)

Skalierung von Valkey- oder Redis OSS-Clustern (Konsole)

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder Redis-OSS-Cluster mithilfe der ElastiCache Management Console skalieren. Während dieses Vorgangs bearbeitet Ihr Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Um einen Valkey- oder Redis OSS-Cluster (Konsole) zu skalieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.
3. Wählen Sie aus der Liste der Cluster den Cluster aus.
4. Wählen Sie Ändern aus.

5. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:

- Wählen Sie in der Liste Node type den Knotentyp aus, auf den Sie skalieren möchten. Wählen Sie zur Aufwärtsskalierung einen Knotentyp, der größer als Ihr bestehender Knoten ist.

6. Wenn Sie die Aufwärtsskalierung sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately. Wenn Sie das Kontrollkästchen für Apply immediately nicht aktivieren, wird die Skalierung während des nächsten Wartungsfensters des Clusters durchgeführt.

7. Wählen Sie Ändern aus.

Wenn Sie im vorherigen Schritt Apply immediately aktiviert haben, ändert sich der Cluster-Status zu modifying. Wenn der Status zu available wechselt, ist die Änderung abgeschlossen und Sie können den neuen Cluster verwenden.

Skalierung von Valkey- oder Redis OSS-Clustern (AWS CLI)

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder Redis-OSS-Cluster mithilfe von AWS CLI skalieren. Während dieses Vorgangs bearbeitet Ihr Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Um einen Valkey- oder Redis OSS-Cluster zu skalieren (AWS CLI)

1. Ermitteln Sie die Knotentypen, auf die Sie skalieren können, indem Sie den `aws elasticache list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
  --replication-group-id my-replication-group-id
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
  --replication-group-id my-replication-group-id
```

Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format).

```
{
  "ScaleUpModifications": [
```

```
    "cache.m3.2xlarge",
    "cache.m3.large",
    "cache.m3.xlarge",
    "cache.m4.10xlarge",
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
    "cache.r3.large",
    "cache.r3.xlarge"
  ]
  "ScaleDownModifications": [
    "cache.t2.micro",
    "cache.t2.small ",
    "cache.t2.medium",
    "cache.t1.small "
  ],
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-modifications](#) in der AWS CLI-Referenz.

2. Ändern Sie Ihre Replikationsgruppe mithilfe des AWS CLI `modify-replication-group` Befehls und der folgenden Parameter, sodass sie auf den neuen, größeren Knotentyp skaliert werden kann.
 - `--replication-group-id` Der Name der zu erstellenden Replikationsgruppe, zu der Sie aufwärtsskalieren.
 - `--cache-node-type`— Der neue Knotentyp, für den Sie den Cluster skalieren möchten. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Befehl `list-allowed-node-type-modifications` zurückgegeben wurden.
 - `--cache-parameter-group-name` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.

- `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie den Parameter `--no-apply-immediately`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-cluster \  
  --cache-node-type cache.m3.xlarge \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-redis-cluster ^  
  --cache-node-type cache.m3.xlarge ^  
  --apply-immediately
```

Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format).

```
{  
  "ReplicationGroup": {  
    "Status": "modifying",  
    "Description": "my-redis-cluster",  
    "NodeGroups": [  
      {  
        "Status": "modifying",  
        "Slots": "0-16383",  
        "NodeGroupId": "0001",  
        "NodeGroupMembers": [  
          {  
            "PreferredAvailabilityZone": "us-east-1f",  
            "CacheNodeId": "0001",  
            "CacheClusterId": "my-redis-cluster-0001-001"  
          },  
          {  
            "PreferredAvailabilityZone": "us-east-1d",  
            "CacheNodeId": "0001",  
            "CacheClusterId": "my-redis-cluster-0001-002"  
          }  
        ]  
      }  
    ]  
  }  
}
```

```

        }
    ]
}
],
"ConfigurationEndpoint": {
    "Port": 6379,
    "Address": "my-redis-
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
},
"ClusterEnabled": true,
"ReplicationGroupId": "my-redis-cluster",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"SnapshotWindow": "07:30-08:30",
"MemberClusters": [
    "my-redis-cluster-0001-001",
    "my-redis-cluster-0001-002"
],
"CacheNodeType": "cache.m3.xlarge",
"DataTiering": "disabled"
"PendingModifiedValues": {}
}
}

```

Weitere Informationen finden Sie unter [modify-replication-group](#) in der AWS CLI-Referenz.

3. Wenn Sie den verwendet haben `--apply-immediately`, überprüfen Sie den Status des Clusters mithilfe des AWS CLI `describe-cache-clusters` Befehls mit dem folgenden Parameter. Wenn sich der Status auf `available` ändert, können Sie mit der Verwendung des neuen, größeren Clusterknotens beginnen.

Skalierung von Valkey- oder Redis OSS-Clustern (API) ElastiCache

Der folgende Prozess skaliert Ihren Cluster mithilfe der API von seinem aktuellen Knotentyp auf einen neuen, größeren Knotentyp. ElastiCache Während dieses Vorgangs werden die DNS-Einträge ElastiCache aktualisiert, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher, Redis OSS 5.0.5 und höher, können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei Version Redis OSS 4.0.10 und niedriger stellen Sie möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der DNS-Eintrag aktualisiert wird.

Die Zeit, die für die Skalierung auf einen größeren Knotentyp benötigt wird, hängt von Ihrem Knotentyp und der Datenmenge in Ihrem aktuellen Cluster ab.

So skalieren Sie einen Valkey- oder Redis OSS-Cache-Cluster (API) ElastiCache

1. Ermitteln Sie, auf welche Knotentypen Sie mithilfe der ElastiCache `ListAllowedNodeTypeModifications` API-Aktion mit dem folgenden Parameter skalieren können.

- `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&ReplicationGroupId=MyReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [ListAllowedNodeTypeModifications](#) in der Amazon ElastiCache API-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe der `ModifyReplicationGroup` ElastiCache API-Aktion und mit den folgenden Parametern auf den neuen Knotentyp.
 - `ReplicationGroupId` der Name der Replikationsgruppe.
 - `CacheNodeType`— der neue, größere Knotentyp der Cluster in dieser Replikationsgruppe. Dieser Wert muss einer der Instance-Typen sein, die von der `ListAllowedNodeTypeModifications` Aktion im vorherigen Schritt zurückgegeben wurden.
 - `CacheParameterGroupName` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.

- `ApplyImmediately` Legen Sie `true` fest, um die Skalierung sofort auszuführen. Wenn Sie die Skalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup  
  &ApplyImmediately=true  
  &CacheNodeType=cache.m3.2xlarge  
  &CacheParameterGroupName=redis32-m3-2x1  
  &ReplicationGroupId=myReplGroup  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie [ModifyReplicationGroup](#) in der Amazon ElastiCache API-Referenz.

3. Falls Sie dies verwendet haben `ApplyImmediately=true`, überwachen Sie den Status der Replikationsgruppe mithilfe der ElastiCache `DescribeReplicationGroups` API-Aktion mit den folgenden Parametern. Wenn der Status von `modifying` zu `available` wechselt, können Sie die neue, aufskalierte Replikationsgruppe verwenden.
- `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeReplicationGroups  
  &ReplicationGroupId=MyReplGroup  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256
```



```
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [DescribeReplicationGroups](#) in der Amazon ElastiCache API-Referenz.

Online-abwärtsskalieren

Themen

- [Herunterskalierung von Valkey- oder Redis OSS-Clustern \(Konsole\)](#)
- [Herunterskalieren von Valkey- oder Redis OSS-Clustern \(\)AWS CLI](#)
- [Herunterskalierung der Valkey- oder Redis OSS-Cluster \(API\) ElastiCache](#)

Herunterskalierung von Valkey- oder Redis OSS-Clustern (Konsole)

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder Redis-OSS-Cluster mithilfe der ElastiCache Management Console herunterskalieren. Während dieses Vorgangs bearbeitet Ihr Valkey- oder Redis OSS-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Um einen Valkey- oder Redis OSS-Cluster herunterzuskalieren (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.
3. Wählen Sie aus der Liste der Cluster Ihren bevorzugten Cluster.
4. Wählen Sie Ändern aus.
5. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
 - Wählen Sie in der Liste Node type den Knotentyp aus, auf den Sie skalieren möchten. Wählen Sie zur Abwärtsskalierung einen Knotentyp, der kleiner als Ihr bestehender Knoten ist. Beachten Sie, dass nicht alle Knotentypen für das Herunterskalieren zur Verfügung stehen.
6. Wenn Sie die Abwärtsskalierung sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately. Wenn Sie das Kontrollkästchen für Apply immediately nicht aktivieren, wird die Abwärtsskalierung während des nächsten Wartungsfensters des Clusters durchgeführt.
7. Wählen Sie Ändern aus.

Wenn Sie im vorherigen Schritt Apply immediately aktiviert haben, ändert sich der Cluster-Status zu modifying. Wenn der Status zu available wechselt, ist die Änderung abgeschlossen und Sie können den neuen Cluster verwenden.

Herunterskalieren von Valkey- oder Redis OSS-Clustern (AWS CLI)

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder Redis-OSS-Cluster mithilfe von herunterskalieren.AWS CLI Während dieses Vorgangs bearbeitet Ihr Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Um einen Valkey- oder Redis OSS-Cluster herunterzuskalieren (AWS CLI)

1. Ermitteln Sie die Knotentypen, auf die Sie herunterskalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
  --replication-group-id my-replication-group-id
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
  --replication-group-id my-replication-group-id
```

Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format).

```
{
  "ScaleUpModifications": [
    "cache.m3.2xlarge",
    "cache.m3.large",
    "cache.m3.xlarge",
    "cache.m4.10xlarge",
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
```

```
        "cache.r3.4xlarge",
        "cache.r3.8xlarge",
        "cache.r3.large",
        "cache.r3.xlarge"
    ]

    "ScaleDownModifications": [
        "cache.t2.micro",
        "cache.t2.small ",
        "cache.t2.medium ",
        "cache.t1.small"
    ]
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-modifications](#) in der AWS CLI-Referenz.

2. Ändern Sie Ihre Replikationsgruppe mithilfe des AWS CLI `modify-replication-group` Befehls und der folgenden Parameter, sodass sie auf den neuen, kleineren Knotentyp herunterskaliert wird.
 - `--replication-group-id` Der Name der Replikationsgruppe, zu der Sie abwärtsskalieren.
 - `--cache-node-type`— Der neue Knotentyp, für den Sie den Cluster skalieren möchten. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Befehl `list-allowed-node-type-modifications` zurückgegeben wurden.
 - `--cache-parameter-group-name` – [Optional) Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
 - `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie den Parameter `--no-apply-immediately`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
    --replication-group-id my-redis-cluster \
    --cache-node-type cache.t2.micro \
```

```
--apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-redis-cluster ^  
  --cache-node-type cache.t2.micro ^  
  --apply-immediately
```

Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format).

```
{  
  "ReplicationGroup": {  
    "Status": "modifying",  
    "Description": "my-redis-cluster",  
    "NodeGroups": [  
      {  
        "Status": "modifying",  
        "Slots": "0-16383",  
        "NodeGroupId": "0001",  
        "NodeGroupMembers": [  
          {  
            "PreferredAvailabilityZone": "us-east-1f",  
            "CacheNodeId": "0001",  
            "CacheClusterId": "my-redis-cluster-0001-001"  
          },  
          {  
            "PreferredAvailabilityZone": "us-east-1d",  
            "CacheNodeId": "0001",  
            "CacheClusterId": "my-redis-cluster-0001-002"  
          }  
        ]  
      }  
    ],  
    "ConfigurationEndpoint": {  
      "Port": 6379,  
      "Address": "my-redis-cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"  
    },  
    "ClusterEnabled": true,  
    "ReplicationGroupId": "my-redis-cluster",  
  }  
}
```

```
    "SnapshotRetentionLimit": 1,  
    "AutomaticFailover": "enabled",  
    "SnapshotWindow": "07:30-08:30",  
    "MemberClusters": [  
        "my-redis-cluster-0001-001",  
        "my-redis-cluster-0001-002"  
    ],  
    "CacheNodeType": "cache.t2.micro",  
    "DataTiering": "disabled"  
    "PendingModifiedValues": {}  
}  
}
```

Weitere Informationen finden Sie unter [modify-replication-group](#) in der AWS CLI-Referenz.

3. Wenn Sie den verwendet haben `--apply-immediately`, überprüfen Sie den Status des Clusters mithilfe des AWS CLI `describe-cache-clusters` Befehls mit dem folgenden Parameter. Wenn sich der Status auf verfügbar ändert, können Sie mit der Verwendung des neuen, kleineren Clusterknotens beginnen.

Herunterskalierung der Valkey- oder Redis OSS-Cluster (API) ElastiCache

Mit dem folgenden Prozess wird Ihre Replikationsgruppe mithilfe der API vom aktuellen Knotentyp auf einen neuen, kleineren Knotentyp skaliert. ElastiCache Während dieses Vorgangs bearbeitet Ihr Valkey- oder Redis OSS-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Die Zeit, die benötigt wird, um auf einen kleineren Knotentyp herunterzuskalieren, hängt von Ihrem Knotentyp und der Datenmenge in Ihrem aktuellen Cluster ab.

Herunterskalierung (ElastiCache API)

1. Ermitteln Sie, auf welche Knotentypen Sie mithilfe der ElastiCache `ListAllowedNodeTypeModifications` API-Aktion mit dem folgenden Parameter herunterskalieren können.
 - `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=ListAllowedNodeTypeModifications
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [ListAllowedNodeTypeModifications](#) in der Amazon ElastiCache API-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe der `ModifyReplicationGroup` ElastiCache API-Aktion und mit den folgenden Parametern auf den neuen Knotentyp herunter.
 - `ReplicationGroupId` der Name der Replikationsgruppe.
 - `CacheNodeType`— der neue, kleinere Knotentyp der Cluster in dieser Replikationsgruppe. Dieser Wert muss einer der Instance-Typen sein, die von der `ListAllowedNodeTypeModifications` Aktion im vorherigen Schritt zurückgegeben wurden.
 - `CacheParameterGroupName` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
 - `ApplyImmediately` – Legen Sie `true` fest, um die Abwärtsskalierung sofort auszuführen. Wenn Sie die Abwärtsskalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&ApplyImmediately=true
&CacheNodeType=cache.t2.micro
&CacheParameterGroupName=redis32-m3-2x1
&ReplicationGroupId=myReplGroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
```

```
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie [ModifyReplicationGroup](#) in der Amazon ElastiCache API-Referenz.

Erste Schritte mit Bloom-Filtern

ElastiCache unterstützt die Bloom-Filter-Datenstruktur, die eine platzsparende probabilistische Datenstruktur bietet, mit der überprüft werden kann, ob ein Element Mitglied einer Menge ist. Bei der Verwendung von Bloom-Filtern sind falsch positive Ergebnisse möglich — ein Filter kann fälschlicherweise angeben, dass ein Element vorhanden ist, obwohl dieses Element dem Satz nicht hinzugefügt wurde. Die Verwendung von Bloom-Filtern verhindert jedoch falsch negative Ergebnisse — falsche Hinweise darauf, dass ein Element nicht existiert, obwohl dieses Element dem Satz hinzugefügt wurde.

Sie können den Prozentsatz potenzieller falsch positiver Ergebnisse auf eine für Ihre Arbeitslast bevorzugte Rate festlegen, indem Sie die FP-Rate anpassen. Sie können auch die Kapazität (die Anzahl der Elemente, die ein Bloom-Filter aufnehmen kann), Eigenschaften für Skalierung und Nicht-Skalierung und vieles mehr konfigurieren.

Nachdem Sie einen Cluster mit einer unterstützten Engine-Version erstellt haben, sind der Bloom-Datentyp und die zugehörigen Befehle automatisch verfügbar. Der bloom Datentyp ist API-kompatibel mit der Bloom-Filter-Befehlssyntax der offiziellen Valkey-Clientbibliotheken `valkey-py`, einschließlich `valkey-java`, und `valkey-go`. Sie können bestehende Bloom-basierte Valkey- und Redis-OSS-Anwendungen problemlos in diese migrieren. ElastiCache Eine vollständige Liste der Befehle finden Sie unter. [Bloom-Filter-Befehle](#)

Die Bloom-bezogenen Metriken `BloomFilterBasedCmds`, `BloomFilterBasedCmdsLatency`, und `BloomFilterBasedCmdsECPUs` sind integriert, CloudWatch um die Verwendung dieses Datentyps zu überwachen. Weitere Informationen finden Sie unter [Metriken für Valkey und Redis OSS](#).

 Note

- Um Bloom-Filter verwenden zu können, müssen Sie ElastiCache Valkey 8.1 und höher verwenden.
- Der Bloom-Datentyp ist nicht RDB-kompatibel mit anderen Bloom-Angeboten, die nicht auf Valkey basieren.

Übersicht über den Datentyp Bloom-Filter

Bloom-Filter sind eine platzsparende probabilistische Datenstruktur, mit der Elemente hinzugefügt und überprüft werden können, ob Elemente vorhanden sind. Falsch positive Ergebnisse sind möglich, wenn ein Filter fälschlicherweise anzeigt, dass ein Element existiert, obwohl es nicht hinzugefügt wurde. Bloom-Filter garantieren jedoch, dass falsch negative Ergebnisse (die fälschlicherweise darauf hinweisen, dass ein Element nicht existiert, obwohl es hinzugefügt wurde) nicht vorkommen.

Die Hauptdokumentationsquelle für Bloom-Filter finden Sie auf der valkey.io-Dokumentationsseite. Diese enthält die folgenden Informationen:

- [Häufige Anwendungsfälle für Bloom-Filter](#)
 - Deduplizierung von Werbung und Ereignissen
 - Betrugserkennung
 - Filterung schädlicher Inhalte /Spam
 - Einzigartige Benutzererkennung
- [Unterschiede zwischen skalierenden und nicht skalierenden Bloom-Filtern](#)
 - Wie entscheidet man sich zwischen skalierenden und nicht skalierenden Bloom-Filtern
- [Bloom-Eigenschaften](#)
 - Erfahren Sie mehr über die einstellbaren Eigenschaften von Bloom-Filtern. Dazu gehören die Falsch-Positiv-Rate, die Kapazität, die Skalierung und die Nicht-Skalierung und vieles mehr.
- [Leistung von Bloom-Befehlen](#)
- [Überwachung der gesamten Bloom-Filter-Statistiken](#)
- [Umgang mit großen Bloom-Filtern](#)

- Empfehlungen und Details, wie Sie überprüfen können, ob ein Bloom-Filter sein Speichernutzungslimit erreicht hat und ob er skaliert werden kann, um die gewünschte Kapazität zu erreichen.
- Mit dem Befehl [BF.INFO](#) können Sie gezielt überprüfen, wie viel Speicher von einem Bloom-Filter-Dokument belegt wird.

Größenbeschränkung für Bloom

Der Speicherverbrauch eines einzelnen Bloom-Filterobjekts ist auf 128 MB begrenzt. Mit dem `BF.INFO <key> SIZE` Befehl können Sie überprüfen, wie viel Speicher von einem Bloom-Filter verbraucht wird.

Bloom ACLs

Ähnlich wie bei den bestehenden Kategorien pro Datentyp (`@string`, `@hash` usw.) wurde eine neue Kategorie `@bloom` hinzugefügt, um die Verwaltung des Zugriffs auf Bloom-Befehle und -Daten zu vereinfachen. Keine anderen vorhandenen Valkey- oder Redis OSS-Befehle gehören zur Kategorie `@bloom`.

Es gibt 3 bestehende ACL-Kategorien, die aktualisiert wurden und nun die neuen Bloom-Befehle enthalten: `@read`, `@write` und `@fast`. Die folgende Tabelle zeigt die Zuordnung der Bloom-Befehle zu den entsprechenden Kategorien.

Befehl Bloom	@read	@write	@fast	@bloom
BF.HINZUFÜGEN		y	y	y
BF.CARD	y		y	y
BF.EXISTIERT	y		y	y
BF.INFO	y		y	y
BF.EINFÜGEN		y	y	y
BF.MADD		y	y	y

Befehl Bloom	@read	@write	@fast	@bloom
BF.M EXISTIERT	y		y	y
BF.RESERVE	y		y	y

Metriken im Zusammenhang mit Bloom-Filtern

Die folgenden CloudWatch Metriken im Zusammenhang mit Bloom-Datenstrukturen werden bereitgestellt:

CW-Metriken	Einheit	Serverlos/knotenbasiert	Description
BloomFilterBasedCommands	Anzahl	beides	Die Gesamtzahl der Bloom-Filterbefehle, einschließlich Lese- und Schreibbefehlen.
BloomFilterBasedCommandsLatency	Mikrosekunden	Selbstverwaltet	Latenz aller Bloom-Filterbefehle, einschließlich Lese- und Schreibbefehlen.
BloomFilterBasedCommandsECPUs	Anzahl	Serverless	ECPUs wird von allen Bloom-Filterbefehlen, einschließlich Lese- und Schreibbefehlen, verbraucht.

Bloom-Filter-Befehle

[Bloom-Filter-Befehle](#) sind auf der [Valkey.io-Website](#) dokumentiert. Jede Befehlsseite bietet einen umfassenden Überblick über die Bloom-Befehle, einschließlich ihrer Syntax, ihres Verhaltens, ihrer Rückgabewerte und möglicher Fehlerbedingungen.

Name	Description
BF.ADD	Fügt einem Bloom-Filter ein einzelnes Element hinzu. Wenn der Filter noch nicht existiert, wird er erstellt.
BF.CARD	Gibt die Kardinalität eines Bloom-Filters zurück.
BF.EXISTIERT	Ermittelt, ob der Bloom-Filter das angegebene Element enthält.
BF.INFO	Gibt Nutzungsinformationen und Eigenschaften eines bestimmten Bloom-Filters zurück.
BF.INSERT	Erstellt einen Bloom-Filter mit 0 oder mehr Elementen oder fügt Elemente zu einem vorhandenen Bloom-Filter hinzu.
BF.MADD	Fügt einem Bloom-Filter ein oder mehrere Elemente hinzu.
BF.M EXISTIERT	Ermittelt, ob der Bloom-Filter ein oder mehrere Elemente enthält.
BF.RESERVE	Erzeugt einen leeren Bloom-Filter mit den angegebenen Eigenschaften.

 Note

BF.LOAD wird von nicht unterstützt. ElastiCache Es ist nur für die Verwendung von AOF relevant, die ElastiCache nicht unterstützt wird.

Erste Schritte mit Watch in Serverless

ElastiCache unterstützt den WATCH Befehl, mit dem Sie Schlüssel auf Änderungen überwachen und bedingte [Transaktionen](#) ausführen können. Der WATCH Befehl ist besonders nützlich für Anwendungen, die eine optimistische Parallelitätssteuerung erfordern. Dadurch wird sichergestellt,

dass Transaktionen nur ausgeführt werden, wenn die überwachten Schlüssel nicht geändert wurden. Dazu gehören Änderungen, die von einem Client vorgenommen wurden, wie Schreibbefehle, und von Valkey selbst, wie Ablauf oder Räumung. Wenn Schlüssel seit ihrer Eingabe WATCH und bis zum Zeitpunkt ihres Eingangs geändert wurden, wird die gesamte Transaktion abgebrochen. EXEC

Für ElastiCache Serverless werden die folgenden Einschränkungen eingeführt:

ElastiCache Serverless WATCH ist auf einen einzelnen Hash-Slot beschränkt. Das bedeutet, dass nur Schlüssel, die demselben Hash-Slot zugeordnet sind, gleichzeitig über dieselbe Verbindung überwacht werden können und die Transaktion, die den Watch-Befehlen folgt, nur auf demselben Hash-Slot ausgeführt werden kann. Wenn eine Anwendung versucht, Schlüssel aus verschiedenen Hash-Slots zu überwachen oder Transaktionsbefehle auszuführen, die auf Schlüsseln ausgeführt werden, die einem anderen Hash-Slot als dem der überwachten Schlüssel zugeordnet sind, wird ein CROSSLT Fehler zurückgegeben. [Hashtags](#) können verwendet werden, um sicherzustellen, dass mehrere Schlüssel demselben Hash-Slot zugeordnet sind.

Darüber hinaus kann der SCAN Befehl nicht innerhalb einer Verbindung mit überwachten Schlüsseln ausgeführt werden und gibt einen command not supported during watch state Fehler zurück.

Die Transaktion wird abgebrochen (als ob überwachte Schlüssel berührt worden wären), wenn ElastiCache Serverless nicht sicher ist, ob ein Schlüssel geändert wurde. Zum Beispiel, wenn ein Slot migriert wurde und die überwachten Schlüssel nicht auf demselben Knoten gefunden werden können.

Codebeispiele

Tasten von verschiedenen Steckplätzen aus überwachen und bedienen

Im folgenden Beispiel sind der überwachte Schlüssel und der im SET Befehl angegebene Schlüssel verschiedenen Hash-Slots zugeordnet. Die Ausführung gibt a zurückCROSSLT ERROR.

```
> WATCH foo:{005119}
OK
> MULTI
OK
> SET bar:{011794} 1234
QUEUED
> EXEC
CROSSLT Keys in request don't hash to the same slot
```

Tasten vom selben Steckplatz aus beobachten und bedienen

Das folgende Beispiel zeigt eine erfolgreiche Transaktion, da der eingegebene Schlüssel WATCH nicht geändert wurde.

```
> WATCH foo:{005119}  
OK  
> MULTI  
OK  
> SET bar:{005119} 1234  
QUEUED  
> EXEC  
1) OK
```

Sehen Sie sich die Schlüssel von verschiedenen Spielautomaten an

Im folgenden Beispiel gibt ein Versuch, WATCH Schlüssel von verschiedenen Steckplätzen gleichzeitig innerhalb derselben Client-Verbindung zu verwenden, a zurückCROSSSLOT ERROR.

```
> WATCH foo:{005119}  
OK  
> WATCH bar:{123455}  
CROSSSLOT Keys in request don't hash to the same slot
```

Limit ansehen

Jede Client-Verbindung kann bis zu 1000 Schlüssel gleichzeitig überwachen.

Unterstützte Befehle im Zusammenhang mit Watch

Die Befehle [WATCH](#) und [UNWATCH](#) sind auf der [Valkey.io-Website](#) dokumentiert. Sie bietet einen umfassenden Überblick über die Befehle, einschließlich ihrer Syntax, ihres Verhaltens, ihrer Rückgabewerte und möglicher Fehlerbedingungen.

Erste Schritte mit Vector Search

Amazon ElastiCache for Valkey unterstützt die Vektorsuche, sodass Sie Milliarden von hochdimensionalen Vektoreinbettungen mit Latenzen von nur Mikrosekunden und einem Abruf von mehr als 99% im Speicher speichern, suchen und aktualisieren können. ElastiCache for Valkey bietet Funktionen zum Indizieren, Suchen und Aktualisieren von Milliarden von hochdimensionalen Vektoreinbettungen von bekannten Anbietern wie Amazon Bedrock, Amazon SageMaker, Anthropic oder OpenAI für schnelles Suchen und Abrufen. Die Vektorsuche für Amazon ElastiCache ist ideal für Anwendungsfälle, in denen Spitzenleistung und Skalierbarkeit die wichtigsten Auswahlkriterien sind. Dazu gehören semantisches Caching, Generierung mit erweitertem Abruf, Empfehlungen in Echtzeit, Personalisierung und Erkennung von Anomalien.

Die Vektorsuche kann in Verbindung mit anderen Funktionen verwendet werden, um Ihre Anwendungen zu verbessern. ElastiCache Die Vektorsuche nach ElastiCache ist in Valkey Version 8.2 auf knotenbasierten Clustern in allen [AWS Regionen](#) ohne zusätzliche Kosten verfügbar. Erstellen Sie zunächst einen neuen Valkey 8.2-Cluster mit dem SDK oder [AWS-Managementkonsole](#). AWS CLI Sie können die Vektorsuche auch in Ihrem vorhandenen Cluster verwenden, indem Sie mit [wenigen Klicks](#) und ohne Ausfallzeiten von einer beliebigen früheren Version von Valkey oder Redis OSS auf Valkey 8.2 aktualisieren.

Überblick über die Vektorsuche

ElastiCache for Valkey bietet Funktionen zum Indizieren, Suchen und Aktualisieren von Milliarden von hochdimensionalen Vektoreinbettungen. Die Vektorsuche ermöglicht es Ihnen, Sekundärindizes für eine effiziente und skalierbare Suche zu erstellen, zu verwalten und zu verwenden. Jeder Vektorsuchvorgang bezieht sich auf einen einzelnen Index. Indexoperationen gelten nur für den angegebenen Index. Eine beliebige Anzahl von Vorgängen kann jederzeit für jeden Index ausgeführt werden, mit Ausnahme von Vorgängen zum Erstellen und Löschen von Indizes. Auf Clusterebene können mehrere Operationen für mehrere Indizes gleichzeitig ausgeführt werden.

In diesem Dokument sind die Begriffe Schlüssel, Zeile und Datensatz identisch und werden synonym verwendet. In ähnlicher Weise werden auch die Begriffe Spalte, Feld, Pfad und Element synonym verwendet.

Der `FT.CREATE` Befehl kann verwendet werden, um einen Index für eine Teilmenge von Schlüsseln mit den angegebenen Indextypen zu erstellen. `FT.SEARCH` führt Abfragen an erstellten Indizes durch und `FT.DROPINDEX` entfernt einen vorhandenen Index und alle zugehörigen Daten. Es gibt keine speziellen Befehle zum Hinzufügen, Löschen oder Ändern indizierter Daten. Die vorhandenen JSON

Befehle `HASH` oder, die einen Schlüssel ändern, der sich in einem Index befindet, aktualisieren den Index automatisch.

Themen

- [Indizes und der Valkey OSS-Keyspace](#)
- [Index-Feldtypen](#)
- [Vektor-Index-Algorithmen](#)
- [Sicherheit bei der Vektorsuche](#)

Indizes und der Valkey OSS-Keyspace

Indizes werden für eine Teilmenge des Valkey-OSS-Schlüsselraums erstellt und verwaltet. Der Schlüsselraum für jeden Index wird durch eine Liste von Schlüsselpräfixen definiert, die bei der Erstellung des Indexes bereitgestellt werden. Die Liste der Präfixe ist optional, und wenn sie weggelassen wird, ist der gesamte Schlüsselraum Teil dieses Indexes. Bei mehreren Indizes können unzusammenhängende oder überlappende Teilmengen des Schlüsselraums ohne Einschränkung ausgewählt werden.

Indizes werden auch so eingegeben, dass sie nur Schlüssel abdecken, die einen übereinstimmenden Typ haben. Derzeit werden Indizes nur für Typen `string` und `json` unterstützt. Ein `HASH` Index indexiert nur `HASH` Schlüssel, die in seiner Präfixliste enthalten sind, und in ähnlicher Weise indexiert ein `JSON` Index nur `JSON` Schlüssel, die in seiner Präfixliste enthalten sind. Schlüssel in der Schlüsselraumpräfixliste eines Indexes, die nicht den angegebenen Typ haben, werden ignoriert und wirken sich nicht auf Suchvorgänge aus.

Ein Index wird aktualisiert, wenn ein Befehl eine Taste ändert, die sich innerhalb des Schlüsselraums des Indexes befindet. Valkey extrahiert automatisch die deklarierten Felder für jeden Index und aktualisiert den Index mit dem neuen Wert. Der Aktualisierungsprozess besteht aus drei Schritten. Im ersten Schritt wird der `HASH`- oder `JSON`-Schlüssel geändert und der anfragende Client wird blockiert. Der zweite Schritt wird im Hintergrund ausgeführt und aktualisiert jeden der Indizes, die den geänderten Schlüssel enthalten. Im dritten Schritt wird der Client entsperrt. Bei Abfrageoperationen, die auf derselben Verbindung wie eine Mutation ausgeführt werden, ist diese Änderung also sofort in den Suchergebnissen sichtbar.

Die Erstellung eines Indexes ist ein mehrstufiger Prozess. Der erste Schritt besteht darin, den Befehl `FT.CREATE` auszuführen, der den Index definiert. Bei erfolgreicher Ausführung einer Erstellung wird automatisch der zweite Schritt eingeleitet — das Backfilling. Der Backfill-Prozess läuft in einem

Hintergrund-Thread und durchsucht den Schlüsselbereich nach Schlüsseln, die sich in der Präfixliste des neuen Indexes befinden. Jeder gefundene Schlüssel wird dem Index hinzugefügt. Schließlich wird der gesamte Schlüsselraum gescannt, wodurch der Indexerstellungsprozess abgeschlossen ist. Beachten Sie, dass während der Ausführung des Backfill-Prozesses Mutationen von indizierten Schlüsseln zulässig sind, es keine Einschränkungen gibt und dass der Index-Backfill-Prozess erst abgeschlossen wird, wenn alle Schlüssel ordnungsgemäß indexiert sind. Abfrageoperationen, die versucht werden, während ein Index aufgefüllt wird, sind nicht zulässig und werden mit einem Fehler beendet. Der Befehl FT.INFO gibt den Status des Backfill-Vorgangs im Feld „backfill_status“ zurück.

Index-Feldtypen

Jeder Index hat einen bestimmten Typ, der zusammen mit der Position eines zu indizierenden Felds (Spalte) deklariert wird, wenn der Index erstellt wird. Bei HASH Schlüsseln ist die Position der Feldname innerhalb von. HASH Bei JSON Schlüsseln ist der Speicherort eine JSON-Pfadbeschreibung. Wenn ein Schlüssel geändert wird, werden die mit den deklarierten Feldern verknüpften Daten extrahiert, in den deklarierten Typ konvertiert und im Index gespeichert. Wenn die Daten fehlen oder nicht erfolgreich in den deklarierten Typ konvertiert werden können, wird dieses Feld aus dem Index weggelassen. Es gibt drei Arten von Feldern, die im Folgenden erklärt werden:

- Vektorfelder enthalten einen Zahlenvektor, der auch als Vektoreinbettung bezeichnet wird. Vektorfelder können verwendet werden, um Vektoren auf der Grundlage bestimmter Entfernungsmetriken zur Messung der Ähnlichkeit zu filtern. Bei HASH Indizes sollte das Feld den gesamten Vektor enthalten, der im Binärformat codiert ist (Little-Endian IEEE 754). Bei JSON Schlüsseln sollte der Pfad auf ein mit Zahlen gefülltes Array der richtigen Größe verweisen. Beachten Sie, dass bei der Verwendung eines JSON-Arrays als Vektorfeld die interne Darstellung des Arrays innerhalb des JSON-Schlüssels in das für den ausgewählten Algorithmus erforderliche Format konvertiert wird, wodurch der Speicherverbrauch und die Genauigkeit reduziert werden. Nachfolgende Lesevorgänge mit den JSON-Befehlen ergeben den reduzierten Genauigkeitswert.
- Zahlenfelder enthalten eine einzelne Zahl. Zahlenfelder können mit dem Bereichs-Suchoperator verwendet werden. Denn es wird erwartet, dass das Feld den ASCII-Text einer Zahl enthält, die im Standardformat von Fest- oder Gleitkommazahlen geschrieben ist. Bei JSON Feldern müssen die numerischen Regeln für JSON-Zahlen befolgt werden. Unabhängig von der Darstellung innerhalb des Schlüssels wird dieses Feld zur Speicherung im Index in eine 64-Bit-Fließkommazahl umgewandelt. Da die zugrunde liegenden Zahlen aufgrund ihrer Genauigkeitseinschränkungen in Fließkommazahlen gespeichert werden, gelten die üblichen Regeln für numerische Vergleiche für Fließkommazahlen.

- Tag-Felder enthalten null oder mehr Tag-Werte, die als einzelne UTF-8-Zeichenfolge codiert sind. Tag-Felder können verwendet werden, um Abfragen nach der Äquivalenz von Tag-Werten zu filtern, wobei entweder Groß- und Kleinschreibung beachtet wird oder nicht. Die Zeichenfolge wird mithilfe eines Trennzeichens (Standard ist ein Komma, kann aber überschrieben werden) in Tagwerte zerlegt, wobei führende und nachfolgende Leerzeichen entfernt werden. In einem einzigen Tag-Feld können beliebig viele Tag-Werte enthalten sein.

Vektor-Index-Algorithmen

Zwei Vektorindex-Algorithmen werden in Valkey unterstützt:

- Flat — Der Flat-Algorithmus ist eine lineare Brute-Force-Verarbeitung aller Vektoren im Index, die exakte Antworten innerhalb der Grenzen der Genauigkeit der Entfernungsberechnungen liefert. Aufgrund der linearen Verarbeitung des Index können die Laufzeiten für diesen Algorithmus bei großen Indizes sehr hoch sein. Flache Indizes unterstützen höhere Aufnahmegeschwindigkeiten.
- Hierarchical Navigable Small Worlds (HNSW) — Der HNSW-Algorithmus ist eine Alternative, bei der annähernd engste Vektorübereinstimmungen im Austausch für wesentlich kürzere Ausführungszeiten bereitgestellt werden. Der Algorithmus wird durch drei Parameter gesteuert, und. `M` `EF_CONSTRUCTION` `EF_RUNTIME` Die ersten beiden Parameter werden bei der Indexerstellung angegeben und können nicht geändert werden. Der `EF_RUNTIME` Parameter hat einen Standardwert, der bei der Indexerstellung angegeben wird, aber danach bei jedem einzelnen Abfragevorgang überschrieben werden kann. Diese drei Parameter wirken zusammen, um den Speicher- und CPU-Verbrauch bei Aufnahme- und Abfragevorgängen auszugleichen und die Qualität der Annäherung an eine exakte KNN-Suche (bekannt als Recall-Ratio) zu steuern.

In HNSW steuert der Parameter `M` die maximale Anzahl von Nachbarn, mit denen jeder Knoten eine Verbindung herstellen kann, wodurch die Indexdichte bestimmt wird. Ein höherer Wert von `M`, z. B. 32 und höher, führt zu einem stärker vernetzten Graphen, wodurch die Abruf- und Abfragegeschwindigkeit verbessert wird, da mehr Pfade existieren, um relevante Nachbarn zu erreichen. Es erhöht jedoch die Indexgröße und den Speicherverbrauch und verlangsamt die Indizierung. Ein niedrigeres `M`, z. B. 8 und weniger, ergibt einen kleineren faster-to-build Index mit geringerer Speicherbelegung, aber der Abruf nimmt ab und Abfragen können aufgrund weniger Verbindungen länger dauern.

Der Parameter `EF_Construction` bestimmt, wie viele mögliche Verbindungen bei der Indexerstellung ausgewertet werden. Ein höherer Wert von `EF_Construction`, z. B. 400 und höher, bedeutet, dass der Indexer mehr Pfade berücksichtigt, bevor er Nachbarn auswählt. Dies führt zu einem

Diagramm, das später sowohl die Abruf- als auch die Abfrageeffizienz verbessert, jedoch auf Kosten einer langsameren Indizierung und einer höheren CPU- und Speicherauslastung während der Konstruktion. Eine niedrige `EF_Construction`, z. B. 64-120, beschleunigt die Indizierung und reduziert den Ressourcenverbrauch, aber das resultierende Diagramm kann die Anzahl der Rückrufaktionen reduzieren und Abfragen verlangsamen, selbst wenn `EF_Runtime` auf einen hohen Wert eingestellt ist.

Schließlich bestimmt `EF_Runtime` den Umfang der Suche während der Abfrage und steuert, wie viele Nachbarkandidaten zur Laufzeit untersucht werden. Ein hoher Wert erhöht den Abruf und die Genauigkeit, geht jedoch auf Kosten der Abfragelatenz und der CPU-Auslastung. Ein niedriger Wert für `EF_Runtime` macht Abfragen schneller und einfacher, allerdings mit geringerem Abruf. Im Gegensatz zu `M` oder `EF_Construction` hat dieser Parameter keinen Einfluss auf die Indexgröße oder die Erstellungszeit, sodass er der Parameter ist, mit dem die Kompromisse zwischen Abruf und Latenz nach der Indexerstellung abgestimmt werden können.

Beide Vektorsuchalgorithmen (Flat und HNSW) unterstützen einen optionalen Parameter.

`INITIAL_CAP` Wenn dieser Parameter angegeben ist, weist er den Indizes vorab Speicher zu, was zu einem geringeren Speicherverwaltungsaufwand und höheren Vektoraufnahmeraten führt. Flache Indizes unterstützen bessere Aufnahmegeschwindigkeiten als HNSW.

Vektorsuchalgorithmen wie HNSW können das Löschen oder Überschreiben zuvor eingefügter Vektoren möglicherweise nicht effizient handhaben. Die Verwendung dieser Operationen kann zu einem übermäßigen Index-Speicherverbrauch und einer and/or Verschlechterung der Abrufqualität führen. Die Neuindizierung ist eine Methode zur Wiederherstellung einer optimalen Speicherauslastung beim Abrufen. and/or

Sicherheit bei der Vektorsuche

Die Sicherheitsmechanismen von [Valkey ACL \(Access Control Lists\)](#) sowohl für den Befehls- als auch für den Datenzugriff wurden erweitert, um die Suchfunktion zu kontrollieren. Die ACL-Steuerung einzelner Suchbefehle wird vollständig unterstützt. Eine neue ACL-Kategorie `@search`, wird bereitgestellt, und viele der vorhandenen Kategorien (`@fast@read`, `@write`, usw.) wurden aktualisiert, um die neuen Befehle aufzunehmen. Suchbefehle ändern keine Schlüsseldaten, was bedeutet, dass die bestehende ACL-Maschinerie für den Schreibzugriff erhalten bleibt. Die Zugriffsregeln für HASH und JSON Operationen werden durch das Vorhandensein eines Indexes nicht verändert; auf diese Befehle wird weiterhin die normale Zugriffskontrolle auf Schlüsselebene angewendet.

Der Zugriff auf Suchbefehle mit einem Index wird ebenfalls über ACL gesteuert. Zugriffsprüfungen werden auf der Ebene des gesamten Indexes durchgeführt, nicht auf der Ebene einzelner Schlüssel. Das bedeutet, dass einem Benutzer nur dann Zugriff auf einen Index gewährt wird, wenn dieser Benutzer berechtigt ist, auf alle möglichen Schlüssel in der Schlüsselraumpräfixliste dieses Indexes zuzugreifen. Mit anderen Worten, der tatsächliche Inhalt eines Indexes steuert den Zugriff nicht. Vielmehr ist es der theoretische Inhalt eines Indexes, wie er in der Präfixliste definiert ist, der für die Sicherheitsüberprüfung verwendet wird. Situationen, in denen ein Benutzer Lese- und and/ or Schreibzugriff auf einen Schlüssel hat, aber nicht auf einen Index zugreifen kann, der diesen Schlüssel enthält, sind möglich. Beachten Sie, dass nur Lesezugriff auf den Schlüsselraum erforderlich ist, um einen Index zu erstellen oder zu verwenden. Das Vorhandensein oder Fehlen von Schreibzugriff wird nicht berücksichtigt.

Funktionen und Grenzen der Vektorsuche

Verfügbarkeit der Vektorsuche

Die Vektorsuche für Amazon ElastiCache ist mit Valkey Version 8.2 auf knotenbasierten Clustern in allen AWS Regionen ohne zusätzliche Kosten verfügbar. [Sie können die Vektorsuche auch für Ihre vorhandenen Cluster verwenden, indem Sie mit wenigen Klicks und ohne Ausfallzeiten von einer beliebigen Version von Valkey oder Redis OSS auf Valkey 8.2 aktualisieren.](#)

Die Vektorsuche ist derzeit für alle ElastiCache Instance-Typen mit Ausnahme von Knoten mit Daten-Tiering verfügbar. Die Verwendung der Vektorsuche auf T2-, T3- und T4G-Instances erfordert eine Erhöhung der Speicherreserve auf mindestens 50% für Mikro- und 30% für kleine Instances. Weitere Informationen finden Sie auf [dieser Seite](#).

Parametrische Einschränkungen

Die folgende Tabelle zeigt Grenzwerte für verschiedene Vektor-Suchelemente:

Grenzwerte für die Vektorsuche

Item	Maximaler Wert
Anzahl der Dimensionen in einem Vektor	32768
Anzahl der Indizes, die erstellt werden können	10
Anzahl der Felder in einem Index	50

Item	Maximaler Wert
FT.SEARCH TIMEOUT-Klausel (Millisekunden)	60000
Maximal zulässige Anzahl von Präfixen pro Index	16
Maximale Länge eines Tag-Felds	10000
Maximale Länge eines numerischen Feldes	256
HNSW M-Parameter	2000000
HNSW EF_KONSTRUKTIONSPARAMETER	4096
HNSW EF_RUNTIME-Parameter	4096

Betriebseinschränkungen

Persistenz und Backfilling von Indizes

Der Aktualisierungsvorgang besteht aus drei Schritten. Im ersten Schritt wird der HASH- oder JSON-Schlüssel geändert und der anfragende Client wird blockiert. Der zweite Schritt wird im Hintergrund ausgeführt und aktualisiert jeden der Indizes, die den geänderten Schlüssel enthalten. Im dritten Schritt wird der Client entsperrt. Bei Abfrageoperationen, die auf derselben Verbindung wie eine Mutation ausgeführt werden, ist diese Änderung also sofort in den Suchergebnissen sichtbar. Das Einfügen oder Aktualisieren eines Schlüssels ist jedoch möglicherweise für einen kurzen Zeitraum nicht in den Suchergebnissen anderer Clients sichtbar. In Zeiten hoher Systemlast und and/or starker Mutation von Daten kann sich die Sichtbarkeitsverzögerung verlängern.

Die Vektorsuchfunktion behält die Definition von Indizes und den Inhalt der Indizes bei. Indizes für Vektorfelder werden gespeichert, aber die Indizes für TAGS und NUMERIC werden nicht gespeichert, was bedeutet, dass sie neu erstellt werden müssen, wenn sie extern geladen werden (vollständige Synchronisierung oder Neuladen). Das bedeutet, dass bei jeder Betriebsanfrage oder jedem Ereignis, das zum Starten oder Neustarten eines Knotens führt, die Indexdefinition und der Inhalt für Vektoren aus dem letzten Snapshot wiederhergestellt werden. Um dies zu initiieren, ist keine Benutzeraktion erforderlich. Bei TAGS- und NUMERIC-Indizes wird die Neuerstellung jedoch als Backfill-Vorgang durchgeführt, sobald die Daten wiederhergestellt sind. Dies entspricht funktionell der automatischen Ausführung eines FT.CREATE-Befehls durch das System für jeden definierten Index. Beachten Sie,

dass der Knoten für Anwendungsoperationen verfügbar ist, sobald die Daten wiederhergestellt sind, aber wahrscheinlich bevor das Auffüllen des Index abgeschlossen ist, was bedeutet, dass Backfill-Operationen wieder für Anwendungen sichtbar werden.

Der Abschluss des Index-Backfills wird nicht zwischen einem primären Replikat und einem Replikat synchronisiert. Dieser Mangel an Synchronisation kann für Anwendungen unerwartet sichtbar werden. Es wird daher empfohlen, dass Anwendungen den Abschluss des Backfills für Primärdateien und alle Replikate überprüfen, bevor sie Suchvorgänge einleiten.

Grenzen der Skalierung

Bei Skalierungsereignissen kann es vorkommen, dass der Index bei der Migration von Daten wieder aufgefüllt wird. Dies führt zu einem geringeren Abruf von Suchanfragen.

Snapshot import/export und Live-Migration

Die RDB-Dateien aus dem einen Cluster mit Suchindizes können in einen anderen ElastiCache Valkey-Cluster mit Version 8.2 oder höher importiert werden. Der neue Cluster wird den Indexinhalt beim Laden der RDB-Datei neu erstellen. Das Vorhandensein von Suchindizes in einer RDB-Datei schränkt jedoch die Kompatibilität dieser Daten mit früheren Versionen von Valkey ein. Das durch die Vektorsuchfunktion definierte Format der Suchindizes wird nur von einem anderen ElastiCache Cluster mit Valkey-Version 8.2 oder höher verstanden. RDB-Dateien, die keine Indizes enthalten, sind auf diese Weise jedoch nicht eingeschränkt.

Beim Auffüllen ist nicht genügend Arbeitsspeicher verfügbar

Ähnlich wie bei Valkey-OSS-Schreiboperationen unterliegt ein Index-Backfill Einschränkungen. out-of-memory Wenn der Engine-Speicher voll ist, während ein Backfill läuft, werden alle Backfills angehalten. Wenn Speicher verfügbar wird, wird der Backfill-Vorgang wieder aufgenommen. Es ist möglich, einen Index zu löschen, wenn das Auffüllen aufgrund von Speichermangel unterbrochen wird.

Transaktionen

Die Befehle `FT.CREATE`, `FT.DROPINDEX`, `FT.ALIASADD`, `FT.ALIASDEL`, und `FT.ALIASUPDATE` können nicht in einem Transaktionskontext ausgeführt werden, d. h. nicht innerhalb eines `MULTI/EXEC` Blocks oder innerhalb eines `LUA`- oder `FUNCTION`-Skripts.

Auswahl der geeigneten Konfiguration

In der Konsolenumgebung können ElastiCache Sie auf einfache Weise den richtigen Instance-Typ auswählen, der auf den Speicher- und CPU-Anforderungen Ihres Vektor-Workloads basiert.

Speicherverbrauch

Der Speicherverbrauch basiert auf der Anzahl der Vektoren, der Anzahl der Dimensionen, dem M-Wert und der Menge der Nicht-Vektordaten, z. B. Metadaten, die dem Vektor zugeordnet sind, oder auf anderen in der Instanz gespeicherten Daten. Der Gesamtspeicherbedarf ist eine Kombination aus dem für die eigentlichen Vektordaten benötigten Speicherplatz und dem für die Vektorindizes benötigten Speicherplatz. Der für Vektordaten benötigte Speicherplatz wird berechnet, indem die tatsächliche Kapazität gemessen wird, die für die Speicherung von Vektoren in unseren HASH JSON Datenstrukturen erforderlich ist, und der Overhead bis zu den nächstgelegenen Speicherplatten, um optimale Speicherzuweisungen zu erzielen. Jeder der Vektorindizes verwendet Verweise auf die in diesen Datenstrukturen gespeicherten Vektordaten sowie eine zusätzliche Kopie des Vektors im Index. Es wird empfohlen, diesen zusätzlichen Speicherverbrauch im Index einzuplanen.

Die Anzahl der Vektoren hängt davon ab, wie Sie Ihre Daten als Vektoren darstellen möchten. Sie können beispielsweise festlegen, dass ein einzelnes Dokument in mehreren Abschnitten dargestellt wird, wobei jeder Abschnitt einen Vektor darstellt. Sie können sich auch dafür entscheiden, das gesamte Dokument als einen einzigen Vektor darzustellen. Die Anzahl der Dimensionen Ihrer Vektoren hängt vom ausgewählten Einbettungsmodell ab. Wenn Sie sich beispielsweise für das AWS Titan-Einbettungsmodell entscheiden, beträgt die Anzahl der Dimensionen 1536. Beachten Sie, dass Sie den Instance-Typ testen sollten, um sicherzustellen, dass er Ihren Anforderungen entspricht.

Skalieren Sie Ihren Workload

Die Vektorsuche unterstützt alle drei Skalierungsmethoden: horizontal, vertikal und Replikate. Bei der Kapazitätsskalierung verhält sich die Vektorsuche genauso wie normales Valkey, d. h. die Erhöhung des Speichers einzelner Knoten (vertikale Skalierung) oder die Erhöhung der Anzahl der Knoten (horizontale Skalierung) erhöht die Gesamtkapazität. Im Clustermodus kann der `FT.CREATE` Befehl an jeden primären Knoten des Clusters gesendet werden, und das System verteilt die neue Indexdefinition automatisch an alle Clustermitglieder.

Aus Sicht der Leistung verhält sich die Vektorsuche jedoch ganz anders als das reguläre Valkey. Die Multithread-Implementierung der Vektorsuche bedeutet zusätzliche CPUs Erträge bis hin zu linearen Steigerungen sowohl des Abfrage- als auch des Aufnahmedurchsatzes. Die horizontale Skalierung

führt zu einer linearen Erhöhung des Aufnahmedurchsatzes, kann jedoch den Abfragedurchsatz verringern. Wenn zusätzlicher Abfragedurchsatz erforderlich ist, ist eine Skalierung durch Replikate oder zusätzliche CPUs Daten erforderlich.

Befehle für die Vektorsuche

Im Folgenden finden Sie eine Liste der unterstützten Befehle für die Vektorsuche.

Themen

- [FT.CREATE](#)
- [FT.SEARCH](#)
- [FT.DROPINDEX](#)
- [FT.INFO](#)
- [FT._LISTE](#)

FT.CREATE

Der `FT.CREATE` Befehl erstellt einen leeren Index und leitet den Backfill-Prozess ein. Jeder Index besteht aus einer Reihe von Felddefinitionen. Jede Felddefinition gibt einen Feldnamen, einen Feldtyp und einen Pfad innerhalb jedes indizierten Schlüssels an, um einen Wert des deklarierten Typs zu finden. Einige Feldtypdefinitionen haben zusätzliche Untertypbezeichner.

Bei Indizes für HASH-Schlüssel ist der Pfad derselbe wie der Name des Hash-Elements. Die optionale `AS` Klausel kann verwendet werden, um das Feld auf Wunsch umzubenennen. Das Umbenennen von Feldern ist besonders nützlich, wenn der Elementname Sonderzeichen enthält.

Bei Indizes für JSON-Schlüssel ist der Pfad ein JSON-Pfad zu den Daten des deklarierten Typs. Da der JSON-Pfad immer Sonderzeichen enthält, ist die `AS` Klausel erforderlich.

Syntax

```
FT.CREATE <index-name>
ON HASH | JSON
[PREFIX <count> <prefix1> [<prefix2>...]]
SCHEMA
(<field-identifier> [AS <alias>]
 | VECTOR [HNSW|FLAT] <attr_count> [<attribute_name> <attribute_value>])
 | TAG [SEPARATOR <sep>] [CASESENSITIVE]
```

```
| NUMERIC
)+
```

(erforderlich): `<index-name>` Dies ist der Name, den Sie Ihrem Index geben. Wenn bereits ein Index mit demselben Namen existiert, wird ein Fehler zurückgegeben.

ON HASH | JSON (optional): Nur Schlüssel, die dem angegebenen Typ entsprechen, sind in diesem Index enthalten. Wenn es weggelassen wird, wird HASH angenommen.

PREFIX (optional): `<prefix-count><prefix>` Wenn diese Klausel angegeben ist, werden nur Schlüssel, die mit denselben Bytes wie eines oder mehrere der angegebenen Präfixe beginnen, in diesen Index aufgenommen. Wenn diese Klausel weggelassen wird, werden alle Schlüssel des richtigen Typs eingeschlossen. Ein Präfix mit der Länge Null würde auch allen Schlüssel des richtigen Typs entsprechen.

Feldtypen:

- TAG: Ein Tag-Feld ist eine Zeichenfolge, die einen oder mehrere Tag-Werte enthält.
 - SEPARATOR `<sep>`(optional): Eines der Zeichen, die zur Abgrenzung einzelner Tags , . <> { } [] " ' : ; ! @ # \$ % ^ & * () - + = ~ verwendet werden. Wenn es weggelassen wird, ist , der Standardwert.
 - CASESENSITIVE (optional): Falls vorhanden, wird bei Tag-Vergleichen zwischen Groß- und Kleinschreibung unterschieden. Standardmäßig wird bei Tag-Vergleichen NICHT zwischen Groß- und Kleinschreibung unterschieden.
- NUMERISCH: Ein numerisches Feld enthält eine Zahl.
- VEKTOR: Ein Vektorfeld enthält einen Vektor. Derzeit werden zwei Algorithmen zur Vektorindizierung unterstützt: HNSW (Hierarchical Navigable Small World) und FLAT (Brute Force). Jeder Algorithmus hat eine Reihe zusätzlicher Attribute, von denen einige erforderlich und andere optional sind.
 - FLAT: Der Flat-Algorithmus liefert exakte Antworten, hat jedoch eine Laufzeit, die proportional zur Anzahl der indizierten Vektoren ist, und ist daher möglicherweise nicht für große Datensätze geeignet.
 - DIM `<number>`(erforderlich): Gibt die Anzahl der Dimensionen in einem Vektor an.
 - TYPE FLOAT32 (erforderlich): Datentyp, FLOAT32 wird derzeit nur unterstützt.
 - DISTANCE_METRIC [L2 | IP | COSINE] (erforderlich): Gibt den Entfernungsalgorithmus an.
 - `<size>INITIAL_CAP` (optional): Anfängliche Indexgröße.

- HNSW: Der HNSW-Algorithmus liefert ungefähre Antworten, arbeitet aber wesentlich schneller als FLAT.
- DIM <number>(erforderlich): Gibt die Anzahl der Dimensionen in einem Vektor an.
- TYPE FLOAT32 (erforderlich): Datentyp, FLOAT32 wird derzeit nur unterstützt.
- DISTANCE_METRIC [L2 | IP | COSINE] (erforderlich): Gibt den Entfernungsalgorithmus an.
- <size>INITIAL_CAP (optional): Anfängliche Indexgröße.
- M <number>(optional): Anzahl der maximal zulässigen ausgehenden Kanten für jeden Knoten im Diagramm in jeder Ebene. Auf Ebene Null beträgt die maximale Anzahl ausgehender Kanten $2 \cdot M$. Der Standardwert ist 16, das Maximum ist 512.
- EF_CONSTRUCTION <number>(optional): Steuert die Anzahl der Vektoren, die bei der Indexkonstruktion untersucht werden. Höhere Werte für diesen Parameter verbessern die Abruftrate auf Kosten längerer Indexerstellungszeiten. Der Standardwert ist 200. Der Höchstwert ist 4096.
- EF_RUNTIME <number>(optional): Steuert die Anzahl der Vektoren, die während eines Abfragevorgangs untersucht werden sollen. Der Standardwert ist 10 und der Höchstwert ist 4096. Sie können diesen Parameterwert für jede Abfrage festlegen, die Sie ausführen. Höhere Werte verlängern die Abfragezeit, verbessern jedoch den Abfrageabruf.

ANTWORT: OK oder Fehler.

FT.SEARCH

Führt eine Suche im angegebenen Index durch. Die Schlüssel, die dem Abfrageausdruck entsprechen, werden zurückgegeben.

```
FT.SEARCH <index-name> <query>
[NOCONTENT]
[RETURN <token_count> (<field-identifier> [AS <alias>])+]
[TIMEOUT timeout]
[PARAMS <count> <name> <value> [<name> <value>]]
[LIMIT <offset> <count>]
```

- <index>(erforderlich): Dieser Indexname, den Sie abfragen möchten.
- <query>(erforderlich): Die Abfragezeichenfolge, Einzelheiten finden Sie unten.
- NOCONTENT (optional): Falls vorhanden, werden nur die resultierenden Schlüsselnamen zurückgegeben, keine Schlüsselwerte sind enthalten.

- **TIMEOUT** <timeout>(optional): Ermöglicht das Festlegen eines Timeout-Werts für den Suchbefehl. Dies muss eine Ganzzahl in Millisekunden sein.
- <count><name1><value1><name2><value2>PARAMETER... (optional): count entspricht der Anzahl der Argumente, d. h. der doppelten Anzahl von Wertenamenpaaren. Einzelheiten zur Verwendung finden Sie in der Abfragezeichenfolge.
- **RÜCKKEHR** <count><field1><field2>... (optional): Anzahl ist die Anzahl der Felder, die zurückgegeben werden sollen. Gibt die Felder an, die Sie aus Ihren Dokumenten abrufen möchten, zusammen mit allen Aliasnamen für die zurückgegebenen Werte. Standardmäßig werden alle Felder zurückgegeben, es sei denn, die Option NOCONTENT ist gesetzt. In diesem Fall werden keine Felder zurückgegeben. Wenn count auf 0 gesetzt ist, verhält es sich genauso wie NOCONTENT.
- **LIMIT**: <offset><count>: Ermöglicht es Ihnen, einen Teil des Ergebnisses auszuwählen. Die ersten <offset>Schlüssel werden übersprungen und nur ein Maximum an <count>Schlüsseln ist enthalten. Die Standardeinstellung ist LIMIT 0 10, wodurch maximal 10 Schlüssel zurückgegeben werden.
- **PARAMS**: Zweimal so viele Schlüssel-Wert-Paare. Auf key/value Parameterpaare kann innerhalb des Abfrageausdrucks verwiesen werden. Weitere Informationen finden Sie unter [Abfrageausdruck für die Vektorsuche](#).
- **DIALEKT**: <dialect>(optional): Gibt Ihren Dialekt an. Der einzige unterstützte Dialekt ist 2.

ANTWORT

Der Befehl gibt bei Erfolg entweder ein Array oder einen Fehler zurück.

Bei Erfolg steht der erste Eintrag im Antwort-Array für die Anzahl der passenden Schlüssel, gefolgt von einem Array-Eintrag für jeden passenden Schlüssel. Beachten Sie, dass die angegebene LIMIT Option nur die Anzahl der zurückgegebenen Schlüssel steuert und den Wert des ersten Eintrags nicht beeinflusst.

Wenn NOCONTENT angegeben, enthält jeder Eintrag in der Antwort nur den passenden Schlüsselnamen. Andernfalls enthält jeder Eintrag den passenden Schlüsselnamen, gefolgt von einem Array der zurückgegebenen Felder. Die Ergebnisfelder für einen Schlüssel bestehen aus einer Reihe von name/value Paaren. Das erste name/value Paar steht für die berechnete Entfernung. Der Name dieses Paares wird aus dem Vektorfeldnamen gebildet, dem „_“ vorangestellt und „_score“ angehängt wird. Der Wert entspricht der berechneten Entfernung. Die verbleibenden name/value Paare sind die Elemente und Werte des Schlüssels, der durch die Klausel gesteuert wird. RETURN

Die Abfragezeichenfolge entspricht dieser Syntax:

```
<filtering>=>[ KNN <K> @<vector_field_name> $<vector_parameter_name> <query-modifiers> ]
```

Wobei Folgendes gilt:

- **<filtering>**: Ist entweder ein* oder ein Filterausdruck. Ein * bedeutet, dass keine Filterung erfolgt und somit alle Vektoren innerhalb des Index durchsucht werden. Ein Filterausdruck kann angegeben werden, um eine Teilmenge der zu durchsuchenden Vektoren zu bezeichnen.
- **<vector_field_name>**: Der Name eines Vektorfeldes innerhalb des angegebenen Index.
- **<K>**: Die Anzahl der Vektoren mit den nächsten Nachbarn, die zurückgegeben werden sollen.
- **<vector_parameter_name>**: Ein PARAM-Name, dessen entsprechender Wert den Abfragevektor für den KNN-Algorithmus bereitstellt. Beachten Sie, dass dieser Parameter als binäre 32-Bit-Gleitkommazahl nach IEEE 754 im Little-Endian-Format codiert werden muss.
- **<query-modifiers>**: (Optional) Eine Liste von keyword/value Paaren, die diese spezielle KNN-Suche modifizieren. Derzeit werden zwei Schlüsselwörter unterstützt:
 - **EF_RUNTIME**: Dieses Schlüsselwort wird von einem Integer-Wert begleitet, der den Standardwert von EF_RUNTIME überschreibt, der bei der Indexerstellung angegeben wurde.
 - **AS**: Dieses Schlüsselwort wird von einem Zeichenkettenwert begleitet, der im Ergebnis zum Namen des Bewertungsfeldes wird und den Standardalgorithmus zur Generierung von Score-Feldnamen überschreibt.

Ausdruck filtern

Ein Filterausdruck besteht aus einer logischen Kombination von Tag-Suchoperatoren und numerischen Suchoperatoren, die in Klammern stehen.

Markierung

Der Tag-Suchoperator wird mit einer oder mehreren Zeichenfolgen angegeben, die durch das Zeichen | getrennt sind. Ein Schlüssel entspricht dem Tag-Suchoperator, wenn das angegebene Feld eine der angegebenen Zeichenketten enthält.

```
@<field_name>:{<tag>}  
or  
@<field_name>:{<tag1> | <tag2>}  
or  
@<field_name>:{<tag1> | <tag2> | ...}
```

Die folgende Abfrage gibt beispielsweise Dokumente mit blauer ODER schwarzer ODER grüner Farbe zurück.

```
@color:{blue | black | green}
```

Als weiteres Beispiel gibt die folgende Abfrage Dokumente zurück, die „Hello World“ oder „Hello Universe“ enthalten.

```
@description:{hello world | hello universe}
```

Numerischer Bereich

Der numerische Bereichsoperator ermöglicht das Filtern von Abfragen, sodass nur Werte zurückgegeben werden, die zwischen einem bestimmten Start- und Endwert liegen. Sowohl inklusive als auch exklusive Bereichsabfragen werden unterstützt. Für einfache relationale Vergleiche kann `+inf`, `-inf` mit einer Bereichsabfrage verwendet werden. Die Syntax für einen Bereichssuchoperator lautet:

```
@<field_name>:[ [()] <bound> [()] <bound>]
```

... wobei `<bound>` entweder eine Zahl oder `+inf` oder `-inf` ist. Grenzen ohne eine führende offene Klammer sind inklusiv, wohingegen Grenzen mit der führenden offenen Klammer exklusiv sind.

Verwenden Sie die folgende Tabelle als Leitfaden für die Zuordnung mathematischer Ausdrücke zu Filterabfragen:

<code>min <= field <= max</code>	<code>@field:[min max]</code>
<code>min < field <= max</code>	<code>@field:[(min max]</code>
<code>min <= field < max</code>	<code>@field:[min (max]</code>
<code>min < field < max</code>	<code>@field:[(min (max]</code>
<code>field >= min</code>	<code>@field:[min +inf]</code>
<code>field > min</code>	<code>@field:[(min +inf]</code>
<code>field <= max</code>	<code>@field:[-inf max]</code>
<code>field < max</code>	<code>@field:[-inf (max]</code>
<code>field == val</code>	<code>@field:[val val]</code>

Logische Operatoren

Mehrere Tags und numerische Suchoperatoren können verwendet werden, um komplexe Abfragen mithilfe logischer Operatoren zu erstellen.

Logisches UND

Um ein logisches UND festzulegen, verwenden Sie ein Leerzeichen zwischen den Prädikaten.

Beispiel:

```
query1 query2 query3
```

Logisches ODER

Um ein logisches ODER festzulegen, verwenden Sie ein Leerzeichen zwischen den Prädikaten.

Beispiel:

```
query1 | query2 | query3
```

Logische Negation

Jede Abfrage kann negiert werden, indem das - Zeichen vor jeder Abfrage vorangestellt wird.

Negative Abfragen geben alle Einträge zurück, die nicht mit der Abfrage übereinstimmen. Dies schließt auch Schlüssel ein, die das Feld nicht enthalten.

Eine negative Abfrage auf `@genre: {comedy}` gibt beispielsweise alle Bücher zurück, die keine Komödie sind UND alle Bücher, die kein Genrefeld haben.

Die folgende Abfrage gibt alle Bücher des Genres „Komödie“ zurück, die nicht zwischen 2015 und 2024 veröffentlicht wurden oder die kein Jahresfeld haben: `@genre: [Komödie] - @year: [2015 2024]`

Rangfolge der Operatoren

Es gelten die typischen Regeln für die Rangfolge von Operatoren, d. h. logisches NEGATE hat die höchste Priorität, gefolgt von logischem UND dann logisches ODER mit der niedrigsten Priorität. Klammern können verwendet werden, um die Standardregeln für die Rangfolge außer Kraft zu setzen.

Beispiele für die Kombination logischer Operatoren

Logische Operatoren können zu komplexen Filterausdrücken kombiniert werden.

Die folgende Abfrage gibt alle Bücher mit den Genres „Komödie“ oder „Horror“ (AND) zurück, die zwischen 2015 und 2024 veröffentlicht wurden: `@genre: [comedy | horror] @year: [2015 2024]`

Die folgende Abfrage gibt alle Bücher mit den Genres „Komödie“ oder „Horror“ (OR) zurück, die zwischen 2015 und 2024 veröffentlicht wurden: `@genre: [comedy | horror] | @year: [2015 2024]`

Die folgende Abfrage gibt alle Bücher zurück, die entweder kein Genrefeld haben oder deren Genrefeld nicht „Komödie“ entspricht und die zwischen 2015 und 2024 veröffentlicht wurden: - @genre:[comedy] @year:[2015 2024]

FT.DROPINDEX

Syntax

```
FT.DROPINDEX <index-name>
```

Der angegebene Index wird gelöscht. Gibt OK oder einen Fehler zurück, wenn dieser Index nicht existiert.

- <index-name>(erforderlich): Der Name des zu löschenden Indexes.

FT.INFO

Syntax

```
FT.INFO <index-name>
```

Die Vektorsuche erweitert den Befehl [FT.INFO](#) um mehrere zusätzliche Abschnitte mit Statistiken und Zählern. Eine Anfrage zum Abrufen des Abschnitts SEARCH ruft alle der folgenden Statistiken ab:

Tastename	Werttyp	Description
index_name	Zeichenfolge	Name des Indexes
index_options	Zeichenfolge	Reserved Instances. Derzeit auf „0“ gesetzt
index_definition	Array	Im Folgenden finden Sie die Definition dieser Array-Elemente.
Attribute	Array von Attributinformationen	Ein Element in diesem Array für jedes definierte Attribut. Die Definition der Attributi

Tastename	Werttyp	Description
		Informationen finden Sie weiter unten.
num_docs	Ganzzahl	Anzahl der aktuell im Index enthaltenen Schlüssel
num_terms	Ganzzahl	Reserved Instances. Derzeit auf „0“ gesetzt.
record_count	Ganzzahl	Die Summe des Felds „Größe“ für jedes Attribut.
hash_indexing_failures	Ganzzahl	Gibt an, wie oft ein Attribut nicht in den deklarierten Attributtyp konvertiert werden konnte. Trotz des Namens gilt dies auch für JSON-Schlüssel.
backfill_in_progress	Ganzzahl	Wenn gerade ein Backfill im Gange ist, wird dies eine '1' sein, andernfalls ist es eine '0'
backfill_percent_complete	float	Schätzung des Abschlusses des Füllstands, eine Bruchzahl im Bereich [0.. 1]
mutation_queue_size	Ganzzahl	Anzahl der Schlüssel, die darauf warten, den Index zu aktualisieren.
recent_mutations_queue_delay	Ganzzahl	Schätzung der Verzögerung (in Sekunden) der Indexaktualisierung. 0, wenn keine Aktualisierungen im Gange sind.

Tastename	Werttyp	Description
state	Zeichenfolge	Backfill-Status: „Bereit“ gibt an, dass das Auffüllen erfolgreich abgeschlossen wurde. „backfill_in_progress“ gibt an, dass das Auffüllen fortgesetzt wird. „backfill_paused_by_oom“ bedeutet, dass das Backfilling aufgrund eines unzureichenden Speicherzustands unterbrochen wurde. Sobald der Speichermangel behoben ist, wird der Backfillvorgang fortgesetzt.

Die Struktur `index_definition` ist ein Array von key/value Paaren, die wie folgt definiert sind:

Tastename	Werttyp	Description
key_type	Zeichenfolge	Entweder die Zeichenfolge 'JSON' oder die Zeichenfolge 'HASH'
prefixes	Array	Jedes Element im Array ist ein definiertes Präfix für den Index. Wenn bei der Erstellung des Indexes keine Präfixe angegeben wurden, hat dieses Array 0 Einträge.
default_score	Zeichenfolge	Reserved Instances. Derzeit auf „1“ eingestellt

Attributinformationen: Die Attributinformationen sind typspezifisch.

Numerische Attribute:

Key (Schlüssel)	Werttyp	Description
Bezeichner	Zeichenfolge	Position des Attributs innerhalb eines Schlüssels. Hash-Mitgliedsname oder JSON-Pfad
alias	Zeichenfolge	Name des Attributs, das in Abfragebeschreibungen verwendet wird.
type	Zeichenfolge	Die Zeichenfolge „NUMERIC“
size	Ganzzahl	Die Anzahl der Schlüssel mit gültigen numerischen Werten in diesem Attribut.

Tag-Attribute:

Tastename	Werttyp	Description
Bezeichner	Zeichenfolge	Position des Attributs innerhalb eines Schlüssels. Hash-Mitgliedsname oder JSON-Pfad
alias	Zeichenfolge	Name des Attributs, das in Abfragebeschreibungen verwendet wird.
type	Zeichenfolge	Die Zeichenfolge „TAG“
SEPARATOR	character	Das Trennzeichen, das bei der Erstellung des Indexes definiert wurde

Tastename	Werttyp	Description
GROSS- UND KLEINSCHR EIBUNG BEACHTEN	–	Diesem Schlüssel ist kein Wert zugeordnet. Es ist nur vorhanden, wenn das Attribut mit dieser Option erstellt wurde.
size	Ganzzahl	Die Anzahl der Schlüssel mit gültigen Tag-Werten in diesem Attribut

Vektor-Attribute:

Tastename	Werttyp	Description
Bezeichner	Zeichenfolge	Position des Attributs innerhalb eines Schlüssels. Hash-Mitgliedsname oder JSON-Pfad
alias	Zeichenfolge	Name des Attributs, das in Abfragebeschreibungen verwendet wird.
type	Zeichenfolge	Die Zeichenfolge „VECTOR“
index	character	Eine weitere Beschreibung des Vektorindex finden Sie unten.

Beschreibung des Vektorindex:

Tastename	Werttyp	Description
Kapazität	Zeichenfolge	Aktuelle Kapazität des Index

Tastename	Werttyp	Description
dimensions	Zeichenfolge	Anzahl der Elemente in jedem Vektor
distance_metric	Zeichenfolge	Einer von „COSINE“, „L2“ oder „IP“
size	Array	Beschreibung des Vektorindex, siehe unten.
data_type	Zeichenfolge	Deklariert Datentyp. Derzeit wird nur FLOAT32 "" unterstützt.
Algorithmus	Array	Weitere Beschreibung des Vektorsuchalgorithmus.

FLAT-Vektor-Suchalgorithmus Beschreibung:

Tastename	Werttyp	Description
name	Zeichenfolge	Name des Algorithmus: FLAT
block_size	number	Größe eines Blocks des FLAT-Index.

Beschreibung des HNSW-Vektorindexes:

Tastename	Werttyp	Description
name	Zeichenfolge	Name des Algorithmus: HNSW
m	number	Der Parameter „M“ für HNSW

Tastename	Werttyp	Description
ef_construction	number	Der Parameter „ef_const ruction“ für HNSW
ef_runtime	number	Der Parameter „ef_runtime“ für HNSW.

FT. _LISTE

Listet alle Indizes auf.

Syntax

```
FT._LIST
```

Gibt ein Array von Zeichenketten zurück, die den Namen des aktuell definierten Indexes entsprechen.

Erste Schritte mit JSON für Valkey und Redis OSS

ElastiCache unterstützt das native JSON-Format (JavaScript Object Notation), das eine einfache, schemalose Methode zur Kodierung komplexer Datensätze in Valkey- und Redis-OSS-Clustern darstellt. Sie können Daten mithilfe des JSON-Formats (JavaScript Object Notation) nativ innerhalb der Cluster speichern und darauf zugreifen und die in diesen Clustern gespeicherten JSON-Daten aktualisieren, ohne dass Sie benutzerdefinierten Code für die Serialisierung und Deserialisierung verwalten müssen.

Neben der Verwendung von Valkey- und Redis OSS-API-Operationen für Anwendungen, die über JSON arbeiten, können Sie jetzt auch bestimmte Teile eines JSON-Dokuments effizient abrufen und aktualisieren, ohne das gesamte Objekt bearbeiten zu müssen. Dies kann die Leistung verbessern und Kosten senken. Sie können den Inhalt Ihres JSON-Dokuments auch mit der [Goessner](#)-Abfrage abfragen.

Nachdem Sie einen Cluster mit einer unterstützten Engine-Version erstellt haben, sind der JSON-Datentyp und die zugehörigen Befehle automatisch verfügbar. API-kompatibel und RDB-kompatibel mit Version 2 des JSON-Moduls, sodass Sie bestehende JSON-basierte Valkey- und Redis OSS-Anwendungen problemlos in diese migrieren können. ElastiCache Weitere Informationen zu den unterstützten Befehlen finden Sie unter. [Unterstützte Valkey- und Redis OSS-Befehle](#)

Die JSON-bezogenen Metriken `JsonBasedCmds` und `JsonBasedCmdsLatency` dienen CloudWatch zur Überwachung der Verwendung dieses Datentyps. Weitere Informationen finden Sie unter [Metriken für Valkey und Redis OSS](#).

Note

Um JSON verwenden zu können, müssen Sie Valkey 7.2 und höher oder Redis OSS 6.2.6 oder höher ausführen.

Themen

- [Überblick über den JSON-Datentyp](#)
- [Unterstützte Valkey- und Redis OSS-Befehle](#)

Überblick über den JSON-Datentyp

ElastiCache unterstützt eine Reihe von Valkey- und Redis OSS-Befehlen für die Arbeit mit dem JSON-Datentyp. Im Folgenden finden Sie eine Übersicht über den JSON-Datentyp und eine detaillierte Liste der unterstützten Befehle.

Terminologie

Begriff	Beschreibung
JSON-Dokument	Bezieht sich auf den Wert eines JSON-Schlüssels.
JSON-Wert	Bezieht sich auf eine Teilmenge eines JSON-Dokuments, einschließlich des Stammzeichnisses, das das gesamte Dokument darstellt. Ein Wert könnte ein Container oder ein Eintrag innerhalb eines Containers sein.
JSON-Element	Äquivalent zu JSON-Wert.

Unterstützter JSON-Standard

Das JSON-Format ist mit [RFC 7159](#) und dem [ECMA-404](#)-JSON-Datenaustauschstandard konform. UTF-8 [Unicode](#) wird im JSON-Text unterstützt.

Stammelement

Das Stammelement kann von jedem JSON-Datentyp stammen. Beachten Sie, dass in früheren RFC 4627 nur Objekte oder Arrays als Stammwerte zugelassen waren. Seit dem Update auf RFC 7159 kann das Stammverzeichnis eines JSON-Dokuments einen beliebigen JSON-Datentyp haben.

Begrenzung der Dokumentgröße

JSON-Dokumente werden intern in einem Format gespeichert, das für schnellen Zugriff und Änderung optimiert ist. Dieses Format führt in der Regel dazu, dass etwas mehr Speicher verbraucht wird als bei der äquivalenten serialisierten Darstellung desselben Dokuments.

Der Speicherverbrauch eines einzelnen JSON-Dokuments ist auf 64 MB begrenzt, was der Größe der In-Memory-Datenstruktur entspricht, nicht der JSON-Zeichenfolge. Sie können mit dem `JSON.DEBUG MEMORY`-Befehl den Speicherplatz überprüfen, der von einem JSON-Dokument verbraucht wird.

JSON ACLs

- Ähnlich wie bei den vorhandenen Pro-Datentyp-Kategorien (`@string`, `@hash` usw.) wird eine neue Kategorie `@json` hinzugefügt, um die Verwaltung des Zugriffs auf JSON-Befehle und -Daten zu vereinfachen. Keine anderen vorhandenen Valkey- oder Redis OSS-Befehle gehören zur Kategorie `@json`. Alle JSON-Befehle erzwingen alle Keyspace- oder Befehlseinschränkungen und -berechtigungen.
- Es gibt fünf bestehende Valkey- und Redis OSS-ACL-Kategorien, die aktualisiert wurden und nun die neuen JSON-Befehle enthalten: `@read`, `@write`, `@fast`, `@slow` und `@admin`. Die folgende Tabelle zeigt die Zuordnung von JSON-Befehlen zu den entsprechenden Kategorien an.

ACL

JSON-Befehl	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		

JSON-Befehl	@read	@write	@fast	@slow	@admin
JSON.ARRINDEX	y		y		
JSON.ARRINSERT		y	y		
JSON.ARRLEN	y		y		
JSON.ARRPOP		y	y		
JSON.ARRTRIM		y	y		
JSON.CLEAR		y	y		
JSON.DEBUG	y			y	y
JSON.DEL		y	y		
JSON.FORGET		y	y		
JSON.GET	y		y		
JSON.MGET	y		y		
JSON.NUMINCRBY		y	y		
JSON.NUMMULTBY		y	y		
JSON.OBJECTS	y		y		

JSON-Befehl	@read	@write	@fast	@slow	@admin
JSON.OBJLEN	y		y		
JSON.RESP	y		y		
JSON.SET		y		y	
JSON.STRAPPEND		y	y		
JSON.STRLEN	y		y		
JSON.STRLEN	y		y		
JSON.TOGGLE		y	y		
JSON.TYPE	y		y		
JSON.NUMINCRBY		y	y		

Begrenzung der Verschachtelungstiefe

Wenn ein JSON-Objekt oder Array ein Element hat, das selbst ein anderes JSON-Objekt oder Array ist, wird gesagt, dass dieses innere Objekt oder Array innerhalb des äußeren Objekts oder Arrays „verschachtelt“ wird. Die maximale Verschachtelungstiefe ist 128. Jeder Versuch, ein Dokument zu erstellen, das eine Verschachtelungstiefe von mehr als 128 enthält, wird mit einem Fehler abgelehnt.

Befehlssyntax

Die meisten Befehle erfordern einen Schlüsselnamen als erstes Argument. Einige Befehle haben auch ein Pfadargument. Das Pfadargument ist standardmäßig das Stammverzeichnis, wenn es optional und nicht im Lieferumfang enthalten ist.

Notation:

- Erforderliche Argumente sind in spitzen Klammern eingeschlossen. Zum Beispiel: <key>
- Optionale Argumente werden in eckige Klammern gesetzt. Zum Beispiel: [path]
- Zusätzliche optionale Argumente sind durch eine Ellipse („...“) gekennzeichnet. Zum Beispiel: [json ...]

Pfadsyntax

Redis JSON unterstützt zwei Arten von Pfadsyntaxen:

- Verbesserte Syntax — Folgt der von [Goessner](#) beschriebenen JSONPath Syntax, wie in der folgenden Tabelle dargestellt. Wir haben die Beschreibungen in der Tabelle zur besseren Übersicht neu angeordnet und geändert.
- Beschränkte Syntax – Hat begrenzte Abfragemöglichkeiten.

Note

Die Ergebnisse einiger Befehle sind davon abhängig, welche Art von Pfadsyntax verwendet wird.

Wenn ein Abfragepfad mit „\$“ beginnt, verwendet er die erweiterte Syntax. Andernfalls wird eine eingeschränkte Syntax verwendet.

Verbesserte Syntax

Symbol/Ausdruck	Beschreibung
\$	Das Stammelement.
. oder []	Untergeordneter Operator
..	Rekursiver Abstieg
*	Platzhalter Alle Elemente in einem Objekt oder Array.
[]	Array-Index-Operator Der Index basiert auf 0.

Symbol/Ausdruck	Beschreibung
[,]	Union-Operator
[start:end:step]	Array-Slice-Operator
?()	Wendet einen Filterausdruck (Skript) auf das aktuelle Array oder Objekt an.
()	Filterausdruck
@	Wird in Filterausdrücken verwendet, die zum aktuellen Knoten verweisen, der verarbeitet wird.
==	Ist gleich; wird in Filterausdrücken verwendet.
!=	Ist nicht gleich; wird in Filterausdrücken verwendet.
>	Größer als; wird in Filterausdrücken verwendet.
>=	Größer als oder gleich; wird in Filterausdrücken verwendet.
<	Kleiner als; wird in Filterausdrücken verwendet.
<=	Kleiner als oder gleich; wird in Filterausdrücken verwendet.
&&	Logisches UND; wird verwendet, um mehrere Filterausdrücke zu kombinieren.
	Logisches ODER; wird verwendet, um mehrere Filterausdrücke zu kombinieren.

Beispiele

Die folgenden Beispiele bauen auf den Beispiel-XML-Daten von [Goessner](#) auf, die wir durch Hinzufügen zusätzlicher Felder geändert haben.

```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",  
      "price": 8.95,  
      "in-stock": true,  
      "sold": true  
    },  
    { "category": "fiction",  
      "author": "Evelyn Waugh",  
      "title": "Sword of Honour",  
      "price": 12.99,  
      "in-stock": false,  
      "sold": true  
    },  
    { "category": "fiction",  
      "author": "Herman Melville",  
      "title": "Moby Dick",  
      "isbn": "0-553-21311-3",  
      "price": 8.99,  
      "in-stock": true,  
      "sold": false  
    },  
    { "category": "fiction",  
      "author": "J. R. R. Tolkien",  
      "title": "The Lord of the Rings",  
      "isbn": "0-395-19395-8",  
      "price": 22.99,  
      "in-stock": false,  
      "sold": false  
    }  
  ],  
  "bicycle": {  
    "color": "red",  
    "price": 19.95,  
    "in-stock": true,  
    "sold": false  
  }  
}
```

Pfad	Beschreibung
<code>\$.store.book[*].author</code>	Die Autoren aller Bücher im Laden.
<code>\$..author</code>	Alle Autoren
<code>\$.store.*</code>	Alle Mitglieder des Ladens.
<code>\$["store"].*</code>	Alle Mitglieder des Ladens.
<code>\$.store..price</code>	Der Preis von allem im Laden.
<code>\$..*</code>	Alle rekursiven Mitglieder der JSON-Struktu.
<code>\$..book[*]</code>	Alle Bücher.
<code>\$..book[0]</code>	Das erste Buch.
<code>\$..book[-1]</code>	Das letzte Buch.
<code>\$..book[0:2]</code>	Die ersten beiden Bücher.
<code>\$..book[0,1]</code>	Die ersten beiden Bücher.
<code>\$..book[0:4]</code>	Bücher von Index 0 bis 3 (Endindex ist nicht inklusive).
<code>\$..book[0:4:2]</code>	Bücher bei Index 0, 2.
<code>\$..book[?(@.isbn)]</code>	Alle Bücher mit einer ISBN-Nummer.
<code>\$..book[?(@.price<10)]</code>	Alle Bücher günstiger als 10 USD.
<code>'\$..book[?(@.price < 10)]'</code>	Alle Bücher günstiger als 10 USD. (Der Pfad muss in Anführungszeichen gesetzt werden, wenn er Leerzeichen enthält.)
<code>'\$..book[?(@["price"] < 10)]'</code>	Alle Bücher günstiger als 10 USD.
<code>'\$..book[?(@.["price"] < 10)]'</code>	Alle Bücher günstiger als 10 USD.

Pfad	Beschreibung
<code>\$..book[?(@.price>=10&&@.price<=100)]</code>	Alle Bücher im Preisbereich von 10 bis 100 USD inklusive.
<code>'\$..book[?(@.price>=10 && @.price<=100)]'</code>	Alle Bücher im Preisbereich von 10 bis 100 USD inklusive. (Der Pfad muss in Anführungszeichen gesetzt werden, wenn er Leerzeichen enthält.)
<code>\$..book[?(@.sold==true @.in-stock==false)]</code>	Alle Bücher verkauft oder ausverkauft.
<code>'\$..book[?(@.sold == true @.in-stock == false)]'</code>	Alle Bücher verkauft oder ausverkauft. (Der Pfad muss in Anführungszeichen gesetzt werden, wenn er Leerzeichen enthält.)
<code>'\$.store.book[?(@.["category"] == "fiction")]'</code>	Alle Bücher der Kategorie Belletristik.
<code>'\$.store.book[?(@.["category"] != "fiction")]'</code>	Alle Bücher in der Kategorie Sachbücher.

Zusätzliche Beispiele für Filterausdrücke:

```
127.0.0.1:6379> JSON.SET k1 . '{"books": [{"price":5,"sold":true,"in-stock":true,"title":"foo"}, {"price":15,"sold":false,"title":"abc"}]}'
OK
127.0.0.1:6379> JSON.GET k1 $.books[?(@.price>1&&@.price<20&&@.in-stock)]
"[{"price":5,"sold":true,"in-stock":true,"title":"foo"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.price>1 && @.price<20 && @.in-stock)]'
"[{"price":5,"sold":true,"in-stock":true,"title":"foo"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?((@.price>1 && @.price<20) && (@.sold==false))]'
"[{"price":15,"sold":false,"title":"abc"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.title == "abc")]'
[{"price":15,"sold":false,"title":"abc"}]

127.0.0.1:6379> JSON.SET k2 . '[1,2,3,4,5]'
127.0.0.1:6379> JSON.GET k2 $.*.[?(@>2)]
"[3,4,5]"
127.0.0.1:6379> JSON.GET k2 '$.*.[?(@ > 2)]'
"[3,4,5]"

127.0.0.1:6379> JSON.SET k3 . '[true,false,true,false,null,1,2,3,4]'
```

```
OK
127.0.0.1:6379> JSON.GET k3 $.*.[?(@==true)]
"[true,true]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ == true)]'
"[true,true]"
127.0.0.1:6379> JSON.GET k3 $.*.[?(@>1)]
"[2,3,4]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ > 1)]'
"[2,3,4]"
```

Beschränkte Syntax

Symbol/Ausdruck	Beschreibung
. oder []	Untergeordneter Operator
[]	Array-Index-Operator Der Index basiert auf 0.

Beispiele

Pfad	Beschreibung
.store.book[0].author	Der Autor des ersten Buches.
.store.book[-1].author	Der Autor des letzten Buches.
.address.city	Name der Stadt.
["store"]["book"][0]["title"]	Der Titel des ersten Buches.
["store"]["book"][-1]["title"]	Der Titel des letzten Buches.

Note

Alle [Goessner](#)-Inhalte, die in dieser Dokumentation erwähnt werden, unterliegen der [Creative-Commons-Lizenz](#).

Häufige Fehlerpräfixe

Jede Fehlermeldung hat ein Präfix. Im Folgenden finden Sie eine Liste mit allgemeinen Fehlerpräfixen.

Präfix	Beschreibung
ERR	Ein allgemeiner Fehler.
LIMIT	Ein Fehler, der auftritt, wenn die Größenbeschränkung überschritten wird. Zum Beispiel wurde die Größenbeschränkung oder Verschachtelungstiefe überschritten.
NONEXISTENT	Ein Schlüssel oder Pfad ist nicht vorhanden.
OUTOFBOUNDARIES	Array-Index außerhalb des gültigen Bereichs.
SYNTAXERR	Syntaxfehler
WRONGTYPE	Falscher Werttyp.

JSON-verwandte Metriken

Die folgenden JSON-Infometriken werden bereitgestellt:

Informationen	Beschreibung
json_total_memory_bytes	JSON-Objekten zugewiesener Gesamtspeicher
json_num_documents	Gesamtzahl der Dokumente in Valkey oder Redis OSS.

Führen Sie den folgenden Befehl aus, um Kernmetriken abzufragen:

```
info json_core_metrics
```

Wie interagiert OSS ElastiCache bei Valkey und Redis mit JSON

Im folgenden Abschnitt wird beschrieben, wie OSS ElastiCache für Valkey und Redis mit dem JSON-Datentyp interagiert.

Rangfolge der Operatoren

Bei der Bewertung bedingter Ausdrücke zum Filtern, haben `&&s` zuerst Vorrang, und dann werden `||s` ausgewertet, wie es in den meisten Sprachen üblich ist. Operationen innerhalb von Klammern werden zuerst ausgeführt.

Verhalten der maximalen Verschachtelungsbeschränkung

Das maximale Limit für Pfadverschachtelung ElastiCache für Redis OSS ist 128. Ein Wert wie `$ . a . b . c . d . . .` kann also nur 128 Level erreichen.

Umgang mit numerischen Werten

JSON hat keine separaten Datentypen für ganze Zahlen und Gleitkommazahlen. Sie werden alle Zahlen genannt.

Numerische Repräsentationen:

Wenn eine JSON-Nummer bei der Eingabe empfangen wird, wird sie in eine der beiden internen Binärdarstellungen umgewandelt: eine 64-Bit-Ganzzahl oder eine doppelt genaue 64-Bit-Gleitkommazahl. Die Ursprüngliche Zeichenfolge und alle ihre Formatierungen werden nicht beibehalten. Wenn also eine Zahl als Teil einer JSON-Antwort ausgegeben wird, wird sie von der internen Binärdarstellung in eine druckbare Zeichenfolge konvertiert, die generische Formatierungsregeln verwendet. Diese Regeln könnten dazu führen, dass eine andere Zeichenfolge generiert wird als empfangen wurde.

Arithmetische Befehle NUMINCRBY und NUMMULTBY:

- Wenn beide Zahlen ganze Zahlen sind und das Ergebnis außerhalb des Bereichs von `int64` liegt, ergibt sich daraus automatisch eine doppelt genaue 64-Bit-Gleitkommazahl.
- Wenn mindestens eine der Zahlen eine Gleitkommazahl ist, ergibt sich daraus eine doppelt genaue 64-Bit-Gleitkommazahl.
- Wenn das Ergebnis den Bereich einer doppelt genauen 64-Bit-Gleitkommazahl überschreitet, gibt der Befehl einen `OVERFLOW`-Fehler aus.

Eine detaillierte Liste der verfügbaren Befehle finden Sie unter [Unterstützte Valkey- und Redis OSS-Befehle](#).

Direktes Array-Filtern

ElastiCache für Valkey oder Redis filtert OSS Array-Objekte direkt.

Bei Daten wie `[0,1,2,3,4,5,6]` und einer Pfadabfrage wie `$[?(@<4)]` oder Daten wie `{"my_key": [0,1,2,3,4,5,6]}` und einer Pfadabfrage wie `ElastiCache würde $.my_key[?(@<4)]` unter beiden Umständen `[1,2,3]` zurückgegeben werden.

Array-Indizierung

ElastiCache für Valkey oder Redis erlaubt OSS sowohl positive als auch negative Indizes für Arrays. Bei einem Array mit der Länge fünf würde 0 das erste Element abfragen, 1 das zweite usw. Negative Zahlen beginnen am Ende des Arrays, also würde -1 das fünfte Element abfragen, -2 das vierte Element usw.

Um ein vorhersehbares Verhalten für Kunden zu gewährleisten, werden Array-Indizes ElastiCache weder nach unten noch nach oben gerundet. Wenn Sie also ein Array mit einer Länge von 5 haben, würde der Aufruf von Index 5 oder höher oder -6 oder niedriger zu keinem Ergebnis führen.

Strikte Syntaxbewertung

MemoryDB erlaubt keine JSON-Pfade mit ungültiger Syntax, selbst wenn eine Teilmenge des Pfads einen gültigen Pfad enthält. Dies soll für unsere Kunden ein korrektes Verhalten sicherstellen.

Unterstützte Valkey- und Redis OSS-Befehle

ElastiCache unterstützt die folgenden Valkey- und Redis OSS-JSON-Befehle:

Themen

- [JSON.ARRAPPEND](#)
- [JSON.ARRINDEX](#)
- [JSON.ARRINSERT](#)
- [JSON.ARRLEN](#)
- [JSON.ARRPOP](#)
- [JSON.ARRTRIM](#)
- [JSON.CLEAR](#)
- [JSON.DEBUG](#)

- [JSON.DEL](#)
- [JSON.FORGET](#)
- [JSON.GET](#)
- [JSON.MGET](#)
- [JSON.MSET](#)
- [JSON.NUMINCRBY](#)
- [JSON.NUMMULTBY](#)
- [JSON.OBJLEN](#)
- [JSON.OBJKEYS](#)
- [JSON.RESP](#)
- [JSON.SET](#)
- [JSON.STRAPPEND](#)
- [JSON.STRLEN](#)
- [JSON.TOGGLE](#)
- [JSON.TYPE](#)

JSON.ARRAPPEND

Hängen Sie einen oder mehrere Werte für die Array-Werte an den Pfad an.

Syntax

```
JSON.ARRAPPEND <key> <path> <json> [json ...]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (erforderlich) – Ein JSON-Pfad.
- **json** (erforderlich) – Der JSON-Wert, der an das Array angehängt werden soll.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die neue Länge des Arrays bei jedem Pfad darstellen.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.

- NONEXISTENT-Fehler, wenn der Pfad nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die neue Länge des Arrays.
- Wenn mehrere Array-Werte ausgewählt sind, gibt der Befehl die neue Länge des ersten aktualisierten Arrays zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Array ist.
- SYNTAXERR-Fehler, wenn eines der eingegebenen JSON-Argumente keine gültige JSON-Zeichenfolge ist.
- NONEXISTENT-Fehler, wenn der Pfad nicht vorhanden ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND k1 $[*] '"c"'
1) (integer) 1
2) (integer) 2
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[[\c\],[a],[a,b],c]"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND k1 [-1] '"c"'
(integer) 3
127.0.0.1:6379> JSON.GET k1
"[[],[a],[a,b],c]"
```

JSON.ARRINDEX

Sucht nach dem ersten Vorkommen eines skalaren JSON-Werts in den Arrays am Pfad.

- Fehler außerhalb des Bereichs werden behandelt, indem der Index auf den Anfang und das Ende des Arrays gerundet wird.
- Wenn `start > end`, return -1 (nicht gefunden).

Syntax

```
JSON.ARRINDEX <key> <path> <json-scalar> [start [end]]
```

- `key` (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- `path` (erforderlich) – Ein JSON-Pfad.
- `json-scalar` (erforderlich) – Der Skalarwert für die Suche nach . JSON-Skalar bezieht sich auf Werte, die keine Objekte oder Arrays sind. Das bedeutet, dass die Werte „Zeichenfolge“, „Zahl“, „Boolescher Wert“ und „Null“ skalare Werte sind.
- `start` (optional) – einschließlich Startindex. Es gilt der Standardwert „0“, falls nicht vorhanden.
- `end` (optional) – Der Endindex (exklusive). Es gilt der Standardwert „0“, wenn keine Angabe gemacht wird, was bedeutet, dass das letzte Element enthalten ist. 0 oder -1 bedeutet, dass das letzte Element enthalten ist.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von ganzen Zahlen. Jeder Wert ist der Index des übereinstimmenden Elements im Array am Pfad. Der Wert ist -1, falls nicht gefunden.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, der Index des übereinstimmenden Elements oder -1, falls nicht gefunden.
- `WRONGTYPE`-Fehler, wenn der Wert im Pfad kein Array ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"], ["a", "b", "c"]]]'
```

```
OK
127.0.0.1:6379> JSON.ARRINDEX k1 $[*] '"b"'
1) (integer) -1
2) (integer) -1
3) (integer) 1
4) (integer) 1
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRINDEX k1 .children '"Tom"'
(integer) 2
```

JSON.ARRINSERT

Fügt einen oder mehrere Werte in die Array-Werte beim Pfad vor dem Index ein.

Syntax

```
JSON.ARRINSERT <key> <path> <index> <json> [json ...]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (erforderlich) – Ein JSON-Pfad.
- **index** (erforderlich) – Ein Array-Index, vor dem Werte eingefügt werden.
- **json** (erforderlich) – Der JSON-Wert, der an das Array angehängt werden soll.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die neue Länge des Arrays bei jedem Pfad darstellen.
- Wenn ein Wert ein leeres Array ist, ist der entsprechende Rückgabewert Null.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.
- **OUTOFBOUNDARIES**-Fehler, wenn das Index-Argument außerhalb des gültigen Bereichs liegt.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die neue Länge des Arrays.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Array ist.
- OUTOFBOUNDARIES-Fehler, wenn das Index-Argument außerhalb des gültigen Bereichs liegt.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]]'
OK
127.0.0.1:6379> JSON.ARRINSERT k1 $[*] 0 '"c"'
1) (integer) 1
2) (integer) 2
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[["c"],["c","a"],["c","a","b"]]"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]]'
OK
127.0.0.1:6379> JSON.ARRINSERT k1 . 0 '"c"'
(integer) 4
127.0.0.1:6379> JSON.GET k1
"[["c"],[],["a"],["a","b"]]"
```

JSON.ARRLEN

Ruft die Länge der Array-Werte am Pfad ab.

Syntax

```
JSON.ARRLEN <key> [path]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.

- `path` (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die Array-Länge bei jedem Pfad darstellen.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, Array-Länge.
- Wenn mehrere Objekte ausgewählt sind, gibt der Befehl die Länge des ersten Arrays zurück.
- `WRONGTYPE`-Fehler, wenn der Wert im Pfad kein Array ist.
- `NONEXISTENT` JSON-Fehler, wenn der Pfad nicht vorhanden ist.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"], ["a", "b", "c"]]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 $[*]
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3

127.0.0.1:6379> JSON.SET k2 . '[[[], "a", ["a", "b"], ["a", "b", "c"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 $[*]
1) (integer) 0
2) (nil)
3) (integer) 2
4) (integer) 3
5) (nil)
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"], ["a", "b", "c"]]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 [*]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k1 [1]
(integer) 1
127.0.0.1:6379> JSON.ARRLEN k1 [2]
(integer) 2

127.0.0.1:6379> JSON.SET k2 . '[[[], "a", ["a", "b"], ["a", "b", "c"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 [1]
(error) WRONGTYPE JSON element is not an array
127.0.0.1:6379> JSON.ARRLEN k2 [0]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k2 [6]
(error) OUTOFBOUNDARIES Array index is out of bounds
127.0.0.1:6379> JSON.ARRLEN k2 a.b
(error) NONEXISTENT JSON path does not exist
```

JSON.ARRPOP

Entfernt und gibt ein Element am Index aus dem Array zurück. Das Öffnen eines leeren Arrays gibt null zurück.

Syntax

```
JSON.ARRPOP <key> [path [index]]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.
- **index** (optional) — Die Position im Array, von der aus das Popping beginnen soll.
 - Ist standardmäßig -1, falls nicht angegeben, was auf das letzte Element verweist.
 - Negativer Wert bedeutet Position vom letzten Element.
 - Außerhalb der Grenzen liegende Indizes werden auf ihre jeweiligen Array-Grenzen gerundet.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Bulk-Strings, die eingeblendete Werte auf jedem Pfad darstellen.
- Wenn ein Wert ein leeres Array ist, ist der entsprechende Rückgabewert Null.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Bulk-String, der den eingeblendeten JSON-Wert darstellt.
- Null, wenn das Array leer ist.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Array ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]]'
OK
127.0.0.1:6379> JSON.ARRPOP k1 $[*]
1) (nil)
2) "\"a\""
3) "\"b\""
127.0.0.1:6379> JSON.GET k1
"[[[],[],[\"a\"]]]"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]]'
OK
127.0.0.1:6379> JSON.ARRPOP k1
"[\"a\"],[\"b\"]"
127.0.0.1:6379> JSON.GET k1
"[[[],[\"a\"]]]"

127.0.0.1:6379> JSON.SET k2 . '[[[], ["a"], ["a", "b"]]]'
OK
127.0.0.1:6379> JSON.ARRPOP k2 . 0
```

```
"[]"
127.0.0.1:6379> JSON.GET k2
"[["a"],["a","b"]]"
```

JSON.ARRTRIM

Trimmt einen Array am Pfad, der dadurch zum Subarray [Start, Ende] wird (beides inklusive).

- Wenn das Array leer ist, tun Sie nichts, und geben Sie 0 zurück.
- Wenn start < 0 ist, behandeln Sie es als 0.
- Wenn end >= size (Größe des Arrays), behandeln Sie es als size-1.
- Wenn start >= size oder start > end, leeren Sie das Array und geben 0 zurück.

Syntax

```
JSON.ARRTRIM <key> <path> <start> <end>
```

- key (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- path (erforderlich) – Ein JSON-Pfad.
- start (erforderlich) – Der Startindex, einschließlich.
- end (erforderlich) – Der Endindex, einschließlich.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die neue Länge des Arrays bei jedem Pfad darstellen.
- Wenn ein Wert ein leeres Array ist, ist der entsprechende Rückgabewert Null.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.
- OUTFOUBOUNDARIES-Fehler, wenn ein Indexargument außerhalb des gültigen Bereichs liegt.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die neue Länge des Arrays.
- Null, wenn das Array leer ist.

- **WRONGTYPE-Fehler**, wenn der Wert im Pfad kein Array ist.
- **OUTOFBOUNDARIES-Fehler**, wenn ein Indexargument außerhalb des gültigen Bereichs liegt.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"], ["a", "b", "c"]]]'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 $[*] 0 1
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 2
127.0.0.1:6379> JSON.GET k1
"[[[],[\"a\"],[\"a\",\"b\"],[\"a\",\"b\"]]]"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 .children 0 1
(integer) 2
127.0.0.1:6379> JSON.GET k1 .children
"[\\"John\\",\\"Jack\\"]"
```

JSON.CLEAR

Entfernt die Arrays oder ein Objekt auf dem Pfad.

Syntax

```
JSON.CLEAR <key> [path]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

- Ganzzahl, die Anzahl der entfernten Container.
- Das Löschen eines leeren Arrays oder Objekts führt zu einem entfernten Container.
- Das Löschen eines Nicht-Container-Werts gibt 0 zurück.

Beispiele

```
127.0.0.1:6379> JSON.SET k1 . '[[[], [0], [0,1], [0,1,2], 1, true, null, "d"]]'
OK
127.0.0.1:6379> JSON.CLEAR k1 $[*]
(integer) 7
127.0.0.1:6379> JSON.CLEAR k1 $[*]
(integer) 4
127.0.0.1:6379> JSON.SET k2 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.CLEAR k2 .children
(integer) 1
127.0.0.1:6379> JSON.GET k2 .children
"[]"
```

JSON.DEBUG

Meldet Informationen. Unterstützte Unterbefehle sind:

- MEMORY <key> [path] – Meldet die Speicherauslastung eines JSON-Wertes in Byte. Falls nicht angegeben, gilt der Root-Standardwert für den Pfad.
- FIELDS <key> [path] – Meldet die Anzahl der Felder im angegebenen Dokumentpfad. Falls nicht angegeben, gilt der Root-Standardwert für den Pfad. Jeder JSON-Wert, der kein Container ist, zählt als ein Feld. Objekte und Arrays zählen rekursiv ein Feld für jeden ihrer JSON-Werte. Jeder Containerwert, mit Ausnahme des Root-Containers, zählt als ein zusätzliches Feld.
- HELP – Druckt Hilfenmeldungen des Befehls.

Syntax

```
JSON.DEBUG <subcommand & arguments>
```

Hängt vom Unterbefehl ab:

MEMORY

- Wenn der Pfad eine erweiterte Syntax ist:
 - Gibt ein Array von ganzen Zahlen zurück, das die Speichergröße (in Byte) des JSON-Werts bei jedem Pfad darstellt.
 - Gibt ein leeres Array zurück, wenn der Valkey- oder Redis-OSS-Schlüssel nicht existiert.
- Wenn der Pfad eine eingeschränkte Syntax hat:
 - Gibt eine Ganzzahl, Speichergröße und den JSON-Wert in Byte zurück.
 - Gibt null zurück, wenn der Valkey- oder Redis-OSS-Schlüssel nicht existiert.

FIELDS

- Wenn der Pfad eine erweiterte Syntax ist:
 - Gibt ein Array von ganzen Zahlen zurück, die die Anzahl der Felder mit JSON-Wert in jedem Pfad darstellen.
 - Gibt ein leeres Array zurück, wenn der Valkey- oder Redis-OSS-Schlüssel nicht existiert.
- Wenn der Pfad eine eingeschränkte Syntax hat:
 - Gibt eine Ganzzahl und die Anzahl der Felder des JSON-Werts zurück.
 - Gibt null zurück, wenn der Valkey- oder Redis-OSS-Schlüssel nicht existiert.

HELP – Gibt ein Array von Hilfefmeldungen zurück.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, [], {"a":1, "b":2}, [1,2,3]]'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1 $[*]
1) (integer) 16
2) (integer) 16
3) (integer) 19
4) (integer) 16
5) (integer) 16
6) (integer) 16
7) (integer) 16
8) (integer) 50
9) (integer) 64
```

```
127.0.0.1:6379> JSON.DEBUG FIELDS k1 $[*]
1) (integer) 1
2) (integer) 1
3) (integer) 1
4) (integer) 1
5) (integer) 1
6) (integer) 0
7) (integer) 0
8) (integer) 2
9) (integer) 3
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646
555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1
(integer) 632
127.0.0.1:6379> JSON.DEBUG MEMORY k1 .phoneNumbers
(integer) 166

127.0.0.1:6379> JSON.DEBUG FIELDS k1
(integer) 19
127.0.0.1:6379> JSON.DEBUG FIELDS k1 .address
(integer) 4

127.0.0.1:6379> JSON.DEBUG HELP
1) JSON.DEBUG MEMORY <key> [path] - report memory size (bytes) of the JSON element.
   Path defaults to root if not provided.
2) JSON.DEBUG FIELDS <key> [path] - report number of fields in the JSON element. Path
   defaults to root if not provided.
3) JSON.DEBUG HELP - print help message.
```

JSON.DEL

Löscht die JSON-Werte am Pfad in einem Dokumentschlüssel. Wenn der Pfad das Stammverzeichnis ist, entspricht dies dem Löschen des Schlüssels aus Valkey oder Redis OSS.

Syntax

```
JSON.DEL <key> [path]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

- Anzahl der gelöschten Elemente.
- 0, wenn der Valkey- oder Redis-OSS-Schlüssel nicht existiert.
- 0, wenn der JSON-Pfad ungültig ist oder nicht existiert.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.DEL k1 $.d.*
(integer) 3
127.0.0.1:6379> JSON.GET k1
"{'a':{}, 'b':{'a':1}, 'c':{'a':1, 'b':2}, 'd':{}, 'e':[1,2,3,4,5]}"
127.0.0.1:6379> JSON.DEL k1 $.e[*]
(integer) 5
127.0.0.1:6379> JSON.GET k1
"{'a':{}, 'b':{'a':1}, 'c':{'a':1, 'b':2}, 'd':{}, 'e':[]}"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}, "e": [1,2,3,4,5]}'
OK
```

```
127.0.0.1:6379> JSON.DEL k1 .d.*
(integer) 3
127.0.0.1:6379> JSON.GET k1
"{\"a\":{\"\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{\"\"e\":[1,2,3,4,5]}}}"
127.0.0.1:6379> JSON.DEL k1 .e[*]
(integer) 5
127.0.0.1:6379> JSON.GET k1
"{\"a\":{\"\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{\"\"e\":[]}}}"
```

JSON.FORGET

Alias von [JSON.DEL](#)

JSON.GET

Gibt das serialisierte JSON in einem oder mehreren Pfaden zurück.

Syntax

```
JSON.GET <key>
[INDENT indentation-string]
[NEWLINE newline-string]
[SPACE space-string]
[NOESCAPE]
[path ...]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **INDENT/NEWLINE/SPACE**(optional) — Steuert das Format der zurückgegebenen JSON-Zeichenfolge, d. h. „pretty print“. Der Standardwert in jedem Fall ist eine leere Zeichenfolge. Die Werte können in jeder Kombination überschrieben werden. Sie können in beliebiger Reihenfolge angegeben werden.
- **NOESCAPE** – Optional, darf aus Gründen der Legacy-Kompatibilität vorhanden sein und hat keine andere Auswirkung.
- **path** (optional) – Null oder mehr JSON-Pfade, standardmäßig das Stammverzeichnis, wenn keine Angabe. Die Pfadargumente müssen am Ende platziert werden.

Ergebnis

Erweiterte Pfad-Syntax:

Wenn ein Pfad angegeben ist:

- Gibt eine serialisierte Zeichenfolge eines Arrays von Werten zurück.
- Wenn kein Wert ausgewählt ist, gibt der Befehl ein leeres Array zurück.

Wenn mehrere Pfade angegeben sind:

- Gibt ein stringiertes JSON-Objekt zurück, in dem jeder Pfad ein Schlüssel ist.
- Wenn es gemischte, erweiterte und eingeschränkte Pfadsyntax gibt, entspricht das Ergebnis der erweiterten Syntax.
- Wenn ein Pfad nicht existiert, ist der entsprechende Wert ein leeres Array.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646
555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.GET k1 $.address.*
["\n21 2nd Street\n","\nNew York\n","\nNY\n","\n10021-3100\n"]
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" $.address.*
["\n\t\n21 2nd Street\n","\n\t\nNew York\n","\n\t\nNY\n","\n\t\n10021-3100\n\n"]
127.0.0.1:6379> JSON.GET k1 $.firstName $.lastName $.age
{"\$.firstName":["\nJohn\n"],"\$.lastName":["\nSmith\n"],"\$.age":["27"]}
127.0.0.1:6379> JSON.SET k2 . '{"a":{, "b":{"a":1}, "c":{"a":1, "b":2}}'
OK
127.0.0.1:6379> json.get k2 $.*
["[{}],{"a":1},{"a":1,"b":2},1,1,2]"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
```

```
{ "street": "21 2nd Street", "city": "New
York", "state": "NY", "zipcode": "10021-3100"}, "phoneNumbers":
[{"type": "home", "number": "212 555-1234"}, {"type": "office", "number": "646
555-4567"}], "children": [], "spouse": null}'
OK
127.0.0.1:6379> JSON.GET k1 .address
"{ \"street\": \"21 2nd Street\", \"city\": \"New York\", \"state\": \"NY\", \"zipcode\":
\"10021-3100\"}"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" .address
"{\n\t\"street\": \"21 2nd Street\", \n\t\"city\": \"New York\", \n\t\"state\": \"NY\", \n
\t\"zipcode\": \"10021-3100\" \n}"
127.0.0.1:6379> JSON.GET k1 .firstName .lastName .age
"{\".firstName\": \"John\", \".lastName\": \"Smith\", \".age\": 27}"
```

JSON.MGET

Wird JSONs am Pfad aus mehreren Dokumentschlüsseln serialisiert. Es gibt null für einen nicht existierenden Schlüssel oder JSON-Pfad zurück.

Syntax

```
JSON.MGET <key> [key ...] <path>
```

- **key** (erforderlich) — Ein oder mehrere Valkey- oder Redis-OSS-Schlüssel des Dokumenttyps.
- **path** (erforderlich) – Ein JSON-Pfad.

Ergebnis

- **Array von Bulk-Strings.** Die Größe des Arrays entspricht der Anzahl der Schlüssel im Befehl. Jedes Element des Arrays wird entweder mit (a) dem serialisierten JSON gefüllt, wie es sich im Pfad befindet, oder (b) mit null, wenn der Schlüssel nicht existiert, der Pfad nicht im Dokument festgehalten oder ungültig (Syntaxfehler) ist.
- Wenn einer der angegebenen Schlüssel existiert und kein JSON-Schlüssel ist, gibt der Befehl den Fehler **WRONGTYPE** zurück.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK
127.0.0.1:6379> JSON.MGET k1 k2 k3 $.address.city
1) ["\New York\"]
2) ["\Boston\"]
3) ["\Seattle\"]
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK

127.0.0.1:6379> JSON.MGET k1 k2 k3 .address.city
1) "\"New York\""
2) "\"Seattle\""
3) "\"Seattle\""
```

JSON.MSET

Wird für Valkey Version 8.1 und höher unterstützt.

Legen Sie JSON-Werte für mehrere Schlüssel fest. Die Operation ist atomar. Entweder sind alle Werte gesetzt oder es sind keine gesetzt.

Syntax

```
JSON.MSET key path json [ key path json ... ]
```

- Wenn der Pfad ein Objektmitglied aufruft:
 - Wenn das übergeordnete Element nicht existiert, gibt der Befehl den Fehler NONEXISTENT zurück.
 - Wenn das übergeordnete Element existiert, aber kein Objekt ist, gibt der Befehl ERROR zurück.
 - Wenn das übergeordnete Element existiert und ein Objekt ist:
 - Wenn das Mitglied nicht existiert, wird ein neues Mitglied an das übergeordnete Objekt angehängt, wenn das übergeordnete Objekt das letzte untergeordnete Objekt im Pfad ist. Andernfalls gibt der Befehl den Fehler NONEXISTENT zurück.
 - Wenn das Mitglied existiert, wird sein Wert durch den JSON-Wert ersetzt.
- Wenn der Pfad einen Array-Index aufruft:
 - Wenn das übergeordnete Element nicht existiert, gibt der Befehl den Fehler NONEXISTENT zurück.
 - Wenn das übergeordnete Element existiert, aber kein Array ist, gibt der Befehl ERROR zurück.
 - Wenn das übergeordnete Element existiert, der Index jedoch außerhalb der Grenzen liegt, gibt der Befehl den Fehler OUTFOUBOUNDARIES zurück.
 - Wenn das übergeordnete Element existiert und der Index gültig ist, wird das Element durch den neuen JSON-Wert ersetzt.
- Wenn der Pfad ein Objekt oder Array aufruft, wird der Wert (Objekt oder Array) durch den neuen JSON-Wert ersetzt.

Ergebnis

- Einfache Zeichenkettenantwort: 'OK', wenn der Vorgang erfolgreich war.
- Einfache Fehlerantwort: Wenn der Vorgang fehlgeschlagen ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.MSET k1 . '[1,2,3,4,5]' k2 . '{"a":{"a":1, "b":2, "c":3}}' k3 .  
'{"a": [1,2,3,4,5]}'  
OK  
127.0.0.1:6379> JSON.GET k1
```

```
"[1,2,3,4,5]"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.MSET k2 $.a.* '0' k3 $.a[*] '0'
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"
127.0.0.1:6379> JSON.GET k3
"{\"a\":[0,0,0,0,0]}"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.MSET k1 . '{"name": "John","address": {"street": "123 Main
St","city": "Springfield"},"phones": ["555-1234","555-5678"]}'
OK
127.0.0.1:6379> JSON.MSET k1 .address.street '"21 2nd Street"' k1 .address.city '"New
York"'
OK
127.0.0.1:6379> JSON.GET k1 .address.street
 "\"21 2nd Street\""
127.0.0.1:6379> JSON.GET k1 .address.city
 "\"New York\""
```

JSON.NUMINCRBY

Gibt die Zahlenwerte am Pfad durch eine bestimmte Zahl in Schritten an.

Syntax

```
JSON.NUMINCRBY <key> <path> <number>
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (erforderlich) – Ein JSON-Pfad.
- **Zahl** (erforderlich) – Eine Zahl

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Bulk-Strings, die eingeblendete Werte auf jedem Pfad darstellen.
- Wenn ein Wert keine Zahl ist, ist der entsprechende Rückgabewert Null.

- **WRONGTYPE**-Fehler, wenn die Zahl nicht analysiert werden kann.
- **OVERFLOW**-Fehler, wenn das Ergebnis außerhalb des Bereichs der doppelt genauen 64-Bit-Gleitkommazahl liegt.
- **NONEXISTENT** wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Bulk-String, der den daraus resultierenden Wert darstellt.
- Wenn mehrere Werte ausgewählt wurden, gibt der Befehl das Ergebnis des zuletzt aktualisierten Arrays zurück.
- **WRONGTYPE**-Fehler, wenn der Wert im Pfad keine Zahl ist.
- **WRONGTYPE**-Fehler, wenn die Zahl nicht analysiert werden kann.
- **OVERFLOW**-Fehler, wenn das Ergebnis außerhalb des Bereichs der doppelt genauen 64-Bit-Gleitkommazahl liegt.
- **NONEXISTENT** wenn der Dokumentschlüssel nicht vorhanden ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 10
"[11,12,13]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[11,12,13]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.a[*] 1
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.b[*] 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.c[*] 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k1
```

```

"{\"a\": [], \"b\": [2], \"c\": [2,3], \"d\": [2,3,4]}"

127.0.0.1:6379> JSON.SET k2 $ '{"a": {}, "b": {"a": 1}, "c": {"a": 1, "b": 2}, "d": {"a": 1, "b": 2, "c": 3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k2 $.a.* 1
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.b.* 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.c.* 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.d.* 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k2
"{\"a\": {}, \"b\": {\"a\": 2}, \"c\": {\"a\": 2, \"b\": 3}, \"d\": {\"a\": 2, \"b\": 3, \"c\": 4}}"

127.0.0.1:6379> JSON.SET k3 $ '{"a": {"a": "a"}, "b": {"a": "a", "b": 1}, "c": {"a": "a", "b": "b"}, "d": {"a": 1, "b": "b", "c": 3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 $.a.* 1
"[null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.b.* 1
"[null,2]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.c.* 1
"[null,null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.d.* 1
"[2,null,4]"
127.0.0.1:6379> JSON.GET k3
"{\"a\": {\"a\": \"a\"}, \"b\": {\"a\": \"a\", \"b\": 2}, \"c\": {\"a\": \"a\", \"b\": \"b\"}, \"d\": {\"a\": 2, \"b\": \"b\", \"c\": 4}}"

```

Eingeschränkte Pfad-Syntax:

```

127.0.0.1:6379> JSON.SET k1 . '{"a": [], "b": [1], "c": [1,2], "d": [1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[1] 10
"12"
127.0.0.1:6379> JSON.GET k1
"{\"a\": [], \"b\": [1], \"c\": [1,2], \"d\": [1,12,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a": [], "b": [1], "c": [1,2], "d": [1,2,3]}'
OK

```

```
127.0.0.1:6379> JSON.NUMINCRBY k1 .a[*] 1
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMINCRBY k1 .b[*] 1
"2"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .c[*] 1
"3"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[*] 1
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[2,3,4]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
  "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k2 .a.* 1
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMINCRBY k2 .b.* 1
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":{}},\"b\":{\"a\":{\"a\":2},\"c\":{\"a\":{\"a\":1,\"b\":2},\"d\":{\"a\":{\"a\":1,\"b\":2,\"c\":{\"a\":3}}}}}"
127.0.0.1:6379> JSON.NUMINCRBY k2 .c.* 1
"3"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":{\"a\":{}},\"b\":{\"a\":{\"a\":2},\"c\":{\"a\":{\"a\":2,\"b\":3},\"d\":{\"a\":{\"a\":1,\"b\":2,\"c\":{\"a\":3}}}}}}}"
127.0.0.1:6379> JSON.NUMINCRBY k2 .d.* 1
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":{\"a\":{\"a\":{}},\"b\":{\"a\":{\"a\":2},\"c\":{\"a\":{\"a\":2,\"b\":3},\"d\":{\"a\":{\"a\":2,\"b\":3,\"c\":{\"a\":4}}}}}}}}}"

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
  "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 .a.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .b.* 1
"2"
127.0.0.1:6379> JSON.NUMINCRBY k3 .c.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .d.* 1
"4"
```


JSON.NUMMULTBY

Multipliziert die Zahlenwerte am Pfad durch eine bestimmte Zahl.

Syntax

```
JSON.NUMMULTBY <key> <path> <number>
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (erforderlich) – Ein JSON-Pfad.
- **Zahl** (erforderlich) – Eine Zahl

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Bulk-Strings, die den sich daraus ergebenden Wert auf jedem Pfad darstellen.
- Wenn ein Wert keine Zahl ist, ist der entsprechende Rückgabewert Null.
- **WRONGTYPE**-Fehler, wenn die Zahl nicht analysiert werden kann.
- **OVERFLOW**-Fehler, wenn das Ergebnis außerhalb des Bereichs eines doppelt genauen 64-Bit-IEEE-Gleitkommawerts liegt.
- **NONEXISTENT** wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Bulk-String, der den daraus resultierenden Wert darstellt.
- Wenn mehrere Werte ausgewählt wurden, gibt der Befehl das Ergebnis des zuletzt aktualisierten Arrays zurück.
- **WRONGTYPE**-Fehler, wenn der Wert im Pfad keine Zahl ist.
- **WRONGTYPE**-Fehler, wenn die Zahl nicht analysiert werden kann.
- **OVERFLOW**-Fehler, wenn das Ergebnis außerhalb des Bereichs eines doppelt genauen 64-Bit-IEEE-Werts liegt.
- **NONEXISTENT** wenn der Dokumentschlüssel nicht vorhanden ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.a[*] 2
"[]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.b[*] 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.c[*] 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k2 $ '{"a":{"a":1}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 $.a.* 2
"[]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.b.* 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.c.* 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.d.* 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k3 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 $.a.* 2
"[null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.b.* 2
"[null,2]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.c.* 2
"[null,null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.d.* 2
```

```
"[2,null,6]"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULBY k1 .d[1] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[1,4,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULBY k1 .a[*] 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULBY k1 .b[*] 2
"2"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULBY k1 .c[*] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULBY k1 .d[*] 2
"6"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULBY k2 .a.* 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULBY k2 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":2},\"b\":{\"a\":2},\"c\":{\"a\":1,\"b\":2},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULBY k2 .c.* 2
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":2},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":4},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULBY k2 .d.* 2
```

```

"6"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"\"a\":2},\"b\":{\"\"a\":2,\"b\":4},\"d\":{\"\"a\":2,\"b\":4,\"c\":6}}}"

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 .a.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"\"a\":\"a\"},\"b\":{\"\"a\":\"a\", \"b\":2},\"c\":{\"\"a\":\"a\", \"b\":\"b\"},\"d\":{ \"a\":1,\"b\":\"b\", \"c\":3}}}"
127.0.0.1:6379> JSON.NUMMULTBY k3 .c.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"\"a\":\"a\"},\"b\":{\"\"a\":\"a\", \"b\":2},\"c\":{\"\"a\":\"a\", \"b\":\"b\"},\"d\":{ \"a\":2,\"b\":\"b\", \"c\":6}}}"

```

JSON.OBJLEN

Ruft die Anzahl der Schlüssel in den Objektwerten am Pfad ab.

Syntax

```
JSON.OBJLEN <key> [path]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die Objektlänge in jedem Pfad darstellen.
- Wenn ein Wert kein Objekt ist, ist der entsprechende Rückgabewert null.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, Anzahl der Schlüssel im Objekt.
- Wenn mehrere Objekte ausgewählt sind, gibt der Befehl die Länge des ersten Objekts zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Objekt ist.
- NONEXISTENT JSON-Fehler, wenn der Pfad nicht vorhanden ist.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3, "b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 $.a
1) (integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 $.a.*
(empty array)
127.0.0.1:6379> JSON.OBJLEN k1 $.b
1) (integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 $.b.*
1) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.c
1) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.c.*
1) (nil)
2) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.d
1) (integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 $.d.*
1) (nil)
2) (nil)
3) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.*
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3
5) (nil)
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 .a
(integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 .a.*
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.OBJLEN k1 .b
(integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 .b.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .c
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .c.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .d
(integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 .d.*
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .*
(integer) 0
```

JSON.OBJKEYS

Ruft Schlüsselnamen in den Objektwerten am Pfad ab.

Syntax

```
JSON.OBJKEYS <key> [path]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Arrays von Bulk-Strings. Jedes Element ist ein Array von Schlüsseln in einem übereinstimmenden Objekt.
- Wenn ein Wert kein Objekt ist, ist der entsprechende Rückgabewert leer.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Array von Bulk-Strings. Jedes Element ist ein Schlüsselname im Objekt.
- Wenn mehrere Objekte ausgewählt sind, gibt der Befehl die Schlüssel des ersten Objekts zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Objekt ist.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3, "b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJKEYS k1 $.*
1) (empty array)
2) 1) "a"
3) 1) "a"
   2) "b"
4) 1) "a"
   2) "b"
   3) "c"
5) (empty array)
127.0.0.1:6379> JSON.OBJKEYS k1 $.d
1) 1) "a"
   2) "b"
   3) "c"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3, "b":4}}, "e":1}'
OK
```

```
127.0.0.1:6379> JSON.OBJKEYS k1 .*  
1) "a"  
127.0.0.1:6379> JSON.OBJKEYS k1 .d  
1) "a"  
2) "b"  
3) "c"
```

JSON.RESP

Gibt den JSON-Wert im angegebenen Pfad im Valkey oder Redis OSS Serialization Protocol (RESP) zurück. Wenn der Wert Container lautet, ist die Antwort ein RESP-Array oder verschachteltes Array.

- JSON null wird dem RESP Null Bulk String zugeordnet.
- Boolesche JSON-Werte werden den jeweiligen RESP Simple Strings zugeordnet.
- Ganzzahlen werden RESP-Ganzzahlen zugeordnet.
- Doppelt genaue 64-Bit-Gleitkommazahlen werden RESP-Bulk-Strings zugeordnet.
- JSON-Strings werden RESP Bulk Strings zugeordnet.
- JSON-Arrays werden als RESP-Arrays dargestellt, wobei das erste Element die einfache Zeichenfolge [ist, gefolgt von den Elementen des Arrays.
- JSON-Objekte werden als RESP-Arrays dargestellt, wobei das erste Element die einfache Zeichenfolge { ist, gefolgt von Schlüssel-Wert-Paaren, von denen jedes ein RESP-Bulk-String ist.

Syntax

```
JSON.RESP <key> [path]
```

- key (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- path (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Arrays. Jedes Array-Element repräsentiert die RESP-Form des Werts in einem Pfad.
- Leeres Array wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Array, das die RESP-Form des Werts in einem Pfad repräsentiert.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646
555-4567"}], "children":[], "spouse":null}'
OK

127.0.0.1:6379> JSON.RESP k1 $.address
1) 1) {
  2) 1) "street"
    2) "21 2nd Street"
  3) 1) "city"
    2) "New York"
  4) 1) "state"
    2) "NY"
  5) 1) "zipcode"
    2) "10021-3100"

127.0.0.1:6379> JSON.RESP k1 $.address.*
1) "21 2nd Street"
2) "New York"
3) "NY"
4) "10021-3100"

127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers
1) 1) [
  2) 1) {
    2) 1) "type"
      2) "home"
    3) 1) "number"
      2) "555 555-1234"
  3) 1) {
```

- 2) 1) "type"
- 2) "office"
- 3) 1) "number"
- 2) "555 555-4567"

```
127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers[*]
```

- 1) 1) {
 - 2) 1) "type"
 - 2) "home"
- 3) 1) "number"
- 2) "212 555-1234"
- 2) 1) {
 - 2) 1) "type"
 - 2) "office"
- 3) 1) "number"
- 2) "555 555-4567"

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 .  
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":  
{"street":"21 2nd Street","city":"New  
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":  
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646  
555-4567"}],"children":[],"spouse":null}'  
OK
```

```
127.0.0.1:6379> JSON.RESP k1 .address
```

- 1) {
- 2) 1) "street"
- 2) "21 2nd Street"
- 3) 1) "city"
- 2) "New York"
- 4) 1) "state"
- 2) "NY"
- 5) 1) "zipcode"
- 2) "10021-3100"

```
127.0.0.1:6379> JSON.RESP k1
```

- 1) {
- 2) 1) "firstName"
- 2) "John"

```
3) 1) "lastName"
    2) "Smith"
4) 1) "age"
    2) (integer) 27
5) 1) "weight"
    2) "135.25"
6) 1) "isAlive"
    2) true
7) 1) "address"
    2) 1) {
        2) 1) "street"
            2) "21 2nd Street"
        3) 1) "city"
            2) "New York"
        4) 1) "state"
            2) "NY"
        5) 1) "zipcode"
            2) "10021-3100"
8) 1) "phoneNumbers"
    2) 1) [
        2) 1) {
            2) 1) "type"
                2) "home"
            3) 1) "number"
                2) "212 555-1234"
        3) 1) {
            2) 1) "type"
                2) "office"
            3) 1) "number"
                2) "555 555-4567"
9) 1) "children"
    2) 1) [
10) 1) "spouse"
    2) (nil)
```

JSON.SET

Legt JSON-Werte im Pfad fest.

Wenn der Pfad ein Objektmitglied aufruft:

- Wenn das übergeordnete Element nicht vorhanden ist, gibt der Befehl den Fehler NONEXISTENT aus.
- Wenn das übergeordnete Element vorhanden aber kein Objekt ist, gibt der Befehl einen FEHLER zurück.
- Wenn das übergeordnete Element existiert und ein Objekt ist:
 - Wenn das Mitglied nicht existiert, wird ein neues Mitglied an das übergeordnete Objekt angehängt, wenn das übergeordnete Objekt das letzte untergeordnete Objekt im Pfad ist. Ansonsten gibt der Befehl den Fehler NONEXISTENT zurück.
 - Wenn das Mitglied existiert, wird sein Wert durch den JSON-Wert ersetzt.

Wenn der Pfad einen Array-Index aufruft:

- Wenn das übergeordnete Element nicht vorhanden ist, gibt der Befehl den Fehler NONEXISTENT aus.
- Wenn das übergeordnete Element vorhanden aber kein Array ist, gibt der Befehl einen FEHLER zurück.
- Wenn das übergeordnete Element vorhanden ist, der Index jedoch außerhalb der Grenzen liegt, gibt der Befehl den Fehler OUTFOUBOUNDARIES zurück.
- Wenn das übergeordnete Element existiert und der Index gültig ist, wird das Element durch den neuen JSON-Wert ersetzt.

Wenn der Pfad ein Objekt oder Array aufruft, wird der Wert (Objekt oder Array) durch den neuen JSON-Wert ersetzt.

Syntax

```
JSON.SET <key> <path> <json> [NX | XX]
```

[NX | XX] Wo Sie 0 oder 1 von [NX | XX] Identifikatoren haben können.

- key (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- path (erforderlich) – Ein JSON-Pfad. Für einen neuen Schlüssel muss der JSON-Pfad der Stamm „.“ sein.
- NX (optional) — Wenn es sich bei dem Pfad um das Stammverzeichnis handelt, legen Sie den Wert nur fest, wenn der Schlüssel nicht existiert. Das heißt, sie fügen ein neues Dokument ein.

Wenn der Pfad nicht der Stamm ist, legen Sie den Wert nur fest, wenn der Pfad nicht existiert. Das heißt, Sie fügen einen Wert in das Dokument ein.

- XX (optional) — Wenn es sich bei dem Pfad um das Stammverzeichnis handelt, legen Sie den Wert nur fest, wenn der Schlüssel vorhanden ist. Das heißt, Sie ersetzen das vorhandene Dokument. Wenn der Pfad nicht der Stamm ist, legen Sie den Wert nur fest, wenn der Pfad existiert. Das heißt, Sie aktualisieren den vorhandenen Wert.

Ergebnis

- Einfache Zeichenfolge 'OK' bei Erfolg.
- Null, wenn die NX- oder XX-Bedingungen nicht erfüllt sind.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.SET k1 $.a.* '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"

127.0.0.1:6379> JSON.SET k2 . '{"a": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k2 $.a[*] '0'
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":[0,0,0,0,0]}"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"c":{"a":1, "b":2}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k1 .c.a '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"c\":{\"a\":0,\"b\":2},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.SET k1 .e[-1] '0'
```

```
OK
127.0.0.1:6379> JSON.GET k1
"{\"c\":{\"a\":0,\"b\":2},\"e\":[1,2,3,4,0]}"
127.0.0.1:6379> JSON.SET k1 .e[5] '0'
(error) OUTOFBOUNDARIES Array index is out of bounds
```

JSON.STRAPPEND

Hängt eine Zeichenfolge an die JSON-Zeichenfolgen im Pfad an.

Syntax

```
JSON.STRAPPEND <key> [path] <json_string>
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.
- **json_string** (erforderlich) – Die JSON-Darstellung einer Zeichenfolge. Beachten Sie, dass eine JSON-Zeichenfolge in Anführungszeichen gesetzt werden muss. Zum Beispiel: `""Beispiel für eine Zeichenfolge""`

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die neue Länge des Arrays bei jedem Pfad darstellen.
- Wenn ein Wert im Pfad keine Zeichenfolge ist, ist der entsprechende Rückgabewert Null.
- SYNTAXERR-Fehler, wenn das angegebene JSON-Argument keine gültige JSON-Zeichenfolge ist.
- NONEXISTENT-Fehler, wenn der Pfad nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die neue Länge der Zeichenfolge.
- Wenn mehrere Zeichenfolgenwerte ausgewählt wurden, gibt der Befehl die neue Länge der zuletzt aktualisierten Zeichenfolge zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad keine Zeichenfolge ist.

- **WRONGTYPE-Fehler**, wenn das angegebene JSON-Argument keine gültige JSON-Zeichenfolge ist.
- **NONEXISTENT-Fehler**, wenn der Pfad nicht vorhanden ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.a '"a"'
1) (integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.* '"a"'
1) (integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 $.b.* '"a"'
1) (integer) 2
2) (nil)
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.* '"a"'
1) (integer) 2
2) (integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.b '"a"'
1) (integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 $.d.* '"a"'
1) (nil)
2) (integer) 2
3) (nil)
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 .a.a '"a"'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .a.* '"a"'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .b.* '"a"'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .c.* '"a"'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .c.b '"a"'
```

```
(integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 .d.* '"a"'
(integer) 2
```

JSON.STRLEN

Ruft die Länge des JSON-Zeichenfolgenwerts im Pfad ab.

Syntax

```
JSON.STRLEN <key> [path]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die Länge des Zeichenfolgenwerts bei jedem Pfad darstellen.
- Wenn ein Wert keine Zeichenfolge ist, ist der entsprechende Rückgabewert Null.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die Länge der Zeichenfolge.
- Wenn mehrere Zeichenfolgenwerte ausgewählt wurden, gibt der Befehl die erste Zeichenfolgenlänge zurück.
- **WRONGTYPE-Fehler**, wenn der Wert im Pfad keine Zeichenfolge ist.
- **NONEXISTENT-Fehler**, wenn der Pfad nicht vorhanden ist.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Beispiele

Erweiterte Pfad-Syntax:


```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 $.a.a
1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.a.*
1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.c.*
1) (integer) 1
2) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.c.b
1) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.d.*
1) (nil)
2) (integer) 1
3) (nil)
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 .a.a
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .a.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.b
(integer) 2
127.0.0.1:6379> JSON.STRLEN k1 .d.*
(integer) 1
```

JSON.TOGGLE

Ein-/Ausblenden der Booleschen Werte zwischen true und false im Pfad.

Syntax

```
JSON.TOGGLE <key> [path]
```

- **key** (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- **path** (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen (0 - falsch, 1 - wahr), die den booleschen Wert in jedem Pfad repräsentieren.
- Wenn ein Wert kein Boolescher Wert ist, ist der entsprechende Rückgabewert Null.
- **NONEXISTENT** wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Zeichenfolge („true“/„false“), die den Booleschen Wert repräsentiert.
- **NONEXISTENT** wenn der Dokumentschlüssel nicht vorhanden ist.
- **WRONGTYPE-Fehler**, wenn der Wert im Pfad kein Boolescher Wert ist.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":true, "b":false, "c":1, "d":null, "e":"foo", "f":
[], "g":{}}'
OK
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 0
2) (integer) 1
3) (nil)
4) (nil)
5) (nil)
6) (nil)
7) (nil)
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 1
2) (integer) 0
3) (nil)
4) (nil)
5) (nil)
```

- 6) (nil)
- 7) (nil)

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . true
OK
127.0.0.1:6379> JSON.TOGGLE k1
"false"
127.0.0.1:6379> JSON.TOGGLE k1
"true"

127.0.0.1:6379> JSON.SET k2 . '{"isAvailable": false}'
OK
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"true"
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"false"
```

JSON.TYPE

Meldet die Art der Werte im jeweiligen Pfad.

Syntax

```
JSON.TYPE <key> [path]
```

- key (erforderlich) — Ein Valkey- oder Redis-OSS-Schlüssel vom JSON-Dokumenttyp.
- path (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Zeichenfolgen, die die Werte in jedem Pfad repräsentieren. Typ {„null“, „boolean“, „string“, „number“, „integer“, „object“ und „array“}.
- Wenn ein Pfad nicht existiert, ist der entsprechende Ausgabewert null.
- Leeres Array wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Zeichenfolge, Typ des Werts
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.
- Null, wenn der JSON-Pfad ungültig ist oder nicht existiert.

Beispiele

Erweiterte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, []]'
OK
127.0.0.1:6379> JSON.TYPE k1 $[*]
1) integer
2) number
3) string
4) boolean
5) null
6) object
7) array
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646
555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.TYPE k1
object
127.0.0.1:6379> JSON.TYPE k1 .children
array
127.0.0.1:6379> JSON.TYPE k1 .firstName
string
127.0.0.1:6379> JSON.TYPE k1 .age
integer
127.0.0.1:6379> JSON.TYPE k1 .weight
number
127.0.0.1:6379> JSON.TYPE k1 .isAlive
```

```
boolean  
127.0.0.1:6379> JSON.TYPE k1 .spouse  
null
```

Verschlagworten Sie Ihre Ressourcen ElastiCache

Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Mithilfe von Tags können Sie Ihre AWS Ressourcen auf unterschiedliche Weise kategorisieren, z. B. nach Zweck, Eigentümer oder Umgebung. Dies ist nützlich, wenn Sie viele Ressourcen desselben Typs haben — In diesem Fall können Sie schnell bestimmte Ressourcen basierend auf den zugewiesenen Tags (Markierungen) bestimmen. In diesem Thema werden Tags (Markierungen) und deren Erstellung beschrieben.

Warning

Als bewährte Vorgehensweise empfehlen wir Ihnen, keine sensiblen Daten in Ihre Tags (Markierungen) aufzunehmen.

Grundlagen zu Tags (Markierungen)

Ein Tag ist eine Bezeichnung, die Sie einer AWS Ressource zuweisen. Jeder Tag (Markierung) besteht aus einem Schlüssel und einem optionalen Wert, beides können Sie bestimmen. Mithilfe von Tags können Sie Ihre AWS Ressourcen auf unterschiedliche Weise kategorisieren, z. B. nach Zweck oder Eigentümer. Sie könnten beispielsweise eine Reihe von Stichwörtern für die ElastiCache Cluster Ihres Kontos definieren, mit deren Hilfe Sie den Besitzer und die Benutzergruppe jeder Instanz nachverfolgen können.

Wir empfehlen die Verwendung von Tag (Markierung)-Schlüsseln, die die Anforderungen der jeweiligen Ressourcentypen erfüllen. Die Verwendung einheitlicher Tag-Schlüssel vereinfacht das Verwalten der -Ressourcen. Sie können die Ressourcen auf Grundlage der hinzugefügten Tags (Markierungen) filtern und danach suchen. Weitere Informationen zum Implementieren einer effektiven Ressourcen-Markierungs-Strategie finden Sie im [Whitepaper AWS Bewährte Methoden zur Markierung](#).

Tags haben keine semantische Bedeutung ElastiCache und werden ausschließlich als Zeichenfolge interpretiert. Außerdem werden Tags (Markierungen) nicht automatisch Ihren

Ressourcen zugewiesen. Sie können Tag (Markierung)-Schlüssel und -Werte bearbeiten und Tags (Markierungen) jederzeit von einer Ressource entfernen. Sie können den Wert eines Tags auf null setzen. Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben. Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht. Wenn Sie einer Replikationsgruppe Tags hinzufügen oder löschen, werden außerdem die Tags aller Knoten in dieser Replikationsgruppe hinzugefügt oder entfernt.

Sie können mit Tags arbeiten, indem Sie die AWS-Managementkonsole, die AWS CLI, und die ElastiCache API verwenden.

Wenn Sie IAM verwenden, können Sie steuern, welche Benutzer in Ihrem AWS Konto berechtigt sind, Tags zu erstellen, zu bearbeiten oder zu löschen. Weitere Informationen finden Sie unter [Berechtigungen auf Ressourcenebene](#).

Ressourcen, die markiert werden können

Sie können die meisten ElastiCache Ressourcen, die bereits in Ihrem Konto vorhanden sind, taggen. In der Tabelle unten werden die Ressourcen aufgeführt, die das Markieren unterstützen. Wenn Sie die AWS-Managementkonsole verwenden, können Sie mithilfe des [Tag-Editors](#) Tags auf Ressourcen anwenden. Auf einigen Ressourcenbildschirmen können Sie beim Erstellen der Ressource Tags für diese Ressource angeben, z. B. ein Tag mit dem Schlüssel „Name“ und einem von Ihnen angegebenen Wert. In den meisten Fällen wendet die Konsole Tags (Markierungen) direkt nach dem Erstellen der Ressource an und nicht während des Erstellens. Die Konsole kann Ressourcen nach dem Name-Tag organisieren, aber dieses Tag hat für den ElastiCache Dienst keine semantische Bedeutung.

Zudem können Sie mit einigen Aktionen zur Ressourcenerstellung Tags beim Erstellen einer Ressource angeben. Wenn Tags (Markierungen) nicht während der Ressourcenerstellung angewendet werden können, wird die Ressourcenerstellung rückgängig gemacht. Auf diese Weise werden Ressourcen entweder mit Tags (Markierungen) oder überhaupt nicht erstellt und keine Ressourcen verbleiben ohne Tags (Markierungen). Indem Sie Ressourcen zum Erstellungszeitpunkt markieren, müssen Sie anschließend keine benutzerdefinierten Skripts ausführen.

Wenn Sie die ElastiCache Amazon-API, die AWS CLI oder ein AWS SDK verwenden, können Sie den Tags Parameter für die entsprechende ElastiCache API-Aktion verwenden, um Tags anzuwenden. Diese sind:

- `CreateServerlessCache`

- `CreateCacheCluster`
- `CreateReplicationGroup`
- `CopyServerlessCacheSnapshot`
- `CopySnapshot`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `CreateServerlessCacheSnapshot`
- `CreateSnapshot`
- `CreateUserGroup`
- `CreateUser`
- `PurchaseReservedCacheNodesOffering`

In der folgenden Tabelle werden die ElastiCache Ressourcen beschrieben, die markiert werden können, und die Ressourcen, die bei der Erstellung mithilfe der ElastiCache API, der AWS CLI oder eines AWS SDK markiert werden können.

Tagging-Unterstützung für Ressourcen ElastiCache

Unterstützt Tags (Markierungen)	Unterstützt Markierung bei der Erstellung
Ja	Ja
Ja	Ja
Ja	Ja
Ja	Ja
Ja	Ja

Unterstützt Tags (Markierungen)	Unterstützt Markierung bei der Erstellung
Ja	Ja
Ja	Ja
Ja	Ja
Ja	Ja
Ja	Ja
Ja	Ja
Ja	Ja

 Note

Globale Datenspeicher können nicht mit Tags versehen werden.

Sie können in Ihren IAM-Richtlinien tagbasierte Berechtigungen auf Ressourcenebene auf ElastiCache API-Aktionen anwenden, die das Tagging bei der Erstellung unterstützen, um eine detaillierte Kontrolle über die Benutzer und Gruppen zu implementieren, die Ressourcen bei der Erstellung taggen können. Ihre Ressourcen sind vor der Erstellung ordnungsgemäß gesichert – Tags, die sofort auf Ihre Ressourcen angewendet werden. Daher sind alle tagbasierten Berechtigungen auf Ressourcenebene, welche die Verwendung von Ressourcen steuern, sofort wirksam. Ihre Ressourcen können nachverfolgt und genauer erfasst werden. Sie können das Markieren neuer Ressourcen gewährleisten und steuern, welche Tag (Markierung)-Schlüssel und Werte für Ihre Ressourcen festgelegt sind.

Weitere Informationen finden Sie unter [Beispiele für das Taggen von Ressourcen](#).

Weitere Informationen zur Markierung von Ressourcen für die Fakturierung finden Sie unter [Überwachung von Kosten mit Kostenzuordnungs-Tags](#).

Markieren von Caches und Snapshots

Die folgenden Regeln gelten für das Markieren im Rahmen von Anforderungsvorgängen:

- **CreateReplicationGroup:**

- Wenn die `--tags` Parameter `--primary-cluster-id` und in der Anforderung enthalten sind, werden die Anforderungs-Tags der Replikationsgruppe hinzugefügt und an alle Cluster in der Replikationsgruppe weitergegeben. Wenn der primäre Cluster über vorhandene Tags verfügt, werden diese mit den Anforderungs-Tags überschrieben, um konsistente Tags für alle Knoten zu gewährleisten.

Wenn keine Anforderungs-Tags vorhanden sind, werden die primären Cluster-Tags der Replikationsgruppe hinzugefügt und an alle Cluster weitergegeben.

- Wenn `--snapshot-name` oder `--serverless-cache-snapshot-name` angegeben wird:

Wenn die Anforderung Tags enthält, wird die Replikationsgruppe nur mit diesen Tags markiert. Wenn die Anforderung keine Tags enthält, werden der Replikationsgruppe die Snapshot-Tags hinzugefügt.

- Wenn `--global-replication-group-id` geliefert wird:

Wenn Tags in der Anfrage enthalten sind, werden die Anforderungs-Tags der Replikationsgruppe hinzugefügt und an alle Cluster weitergegeben.

- **CreateCacheCluster :**

- Wenn `--replication-group-id` geliefert wird:

Wenn Tags in der Anfrage enthalten sind, wird der Cluster nur mit diesen Tags gekennzeichnet. Wenn die Anfrage keine Tags enthält, erbt der Cluster die Tags der Replikationsgruppe anstelle der Tags des primären Clusters.

- Wenn `--snapshot-name` geliefert wird:

Wenn Tags in der Anfrage enthalten sind, wird der Cluster nur mit diesen Tags gekennzeichnet. Wenn die Anfrage keine Tags enthält, werden die Snapshot-Tags dem Cluster hinzugefügt.

- **CreateServerlessCache :**

- Wenn die Anforderung Tags enthält, werden dem Serverless-Cache nur die Anforderungs-Tags hinzugefügt.

- **CreateSnapshot :**

- Wenn `--replication-group-id` geliefert wird:

Wenn die Anforderung Tags enthält, werden nur die Anfrage-Tags zum Snapshot hinzugefügt.
Wenn die Anforderung keine Tags enthält, werden die Replikationsgruppen-Tags zum Snapshot hinzugefügt.

- Wenn `--cache-cluster-id` geliefert wird:

Wenn die Anforderung Tags enthält, werden nur die Anfrage-Tags zum Snapshot hinzugefügt.
Wenn die Anfrage keine Tags enthält, werden die Cluster-Tags dem Snapshot hinzugefügt.

- Für automatische Snapshots:

Tags werden von den Replikationsgruppen-Tags weitergegeben.

- `CreateServerlessCacheSnapshot` :

- Wenn die Anforderung Tags enthält, werden dem Serverless-Cache-Snapshot nur die Anforderungs-Tags hinzugefügt.

- `CopySnapshot` :

- Wenn die Anforderung Tags enthält, werden nur die Request-Tags zum Snapshot hinzugefügt.
Wenn die Anforderung keine Tags enthält, werden die Quell-Snapshot-Tags zum kopierten Snapshot hinzugefügt.

- `CopyServerlessCacheSnapshot` :

- Wenn die Anforderung Tags enthält, werden dem Serverless-Cache-Snapshot nur die Anforderungs-Tags hinzugefügt.

- `AddTagsToResource` und `RemoveTagsFromResource`:

- Die Tags added/removed stammen aus der Replikationsgruppe und die Aktion wird an alle Cluster in der Replikationsgruppe weitergegeben.

Note

`AddTagsToResource` und `RemoveTagsFromResource` kann nicht für Standardparameter und Sicherheitsgruppen verwendet werden.

- `IncreaseReplicaCount` und `ModifyReplicationGroupShardConfiguration`:

- Für alle neuen Cluster, die der Replikationsgruppe hinzugefügt wurden, werden dieselben Tags wie für die Replikationsgruppe angewendet.

Tag-Einschränkungen

Die folgenden grundlegenden Einschränkungen gelten für Tags (Markierungen):

- Maximale Anzahl von Tags (Markierungen) pro Ressource: 50
- Jeder Tag (Markierung) muss für jede Ressource eindeutig sein. Jeder Tag (Markierung) kann nur einen Wert haben.
- Maximale Schlüssellänge – 128 Unicode-Zeichen in UTF-8.
- Maximale Wertlänge – 256 Unicode-Zeichen in UTF-8.
- Obwohl ElastiCache jedes beliebige Zeichen in seinen Tags zulässig ist, können andere Dienste restriktiv sein. Folgende Zeichen sind dienstübergreifend zulässig: Buchstaben, Zahlen und Leerzeichen, die in UTF-8 dargestellt werden können, sowie die folgenden Zeichen: + - = . _ : / @
- Bei Tag-Schlüsseln und -Werten wird zwischen Groß- und Kleinschreibung unterschieden.
- Das `aws :` Präfix ist für die AWS Verwendung reserviert. Wenn der Tag (Markierung) über einen Tag (Markierung)-Schlüssel mit diesem Präfix verfügt, können Sie den Schlüssel oder Wert des Tags (Markierung) nicht bearbeiten oder löschen. Tags (Markierungen) mit dem Präfix `aws :` werden nicht als Ihre Tags (Markierungen) pro Ressourcenlimit angerechnet.

Sie können Ressourcen nicht allein auf Grundlage ihrer Tags (Markierungen) beenden, anhalten oder löschen. Sie müssen den Ressourcenbezeichner angeben. Um Snapshots zu löschen, die Sie mit dem Tag (Markierung)-Schlüssel `DeleteMe` markiert haben, müssen Sie die `DeleteSnapshot`-Aktion mit den Ressourcenbezeichnern der Snapshots verwenden, z. B. `snap-1234567890abcdef0`.

Weitere Informationen zu ElastiCache Ressourcen, die Sie taggen können, finden Sie unter [Ressourcen, die markiert werden können](#).

Beispiele für das Taggen von Ressourcen

- Einen serverlosen Cache mithilfe von Tags erstellen. In diesem Beispiel wird Memcached als Engine verwendet.

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine memcached \  
  --tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- Hinzufügen von Tags zu einem Serverless-Cache

```
aws elasticache add-tags-to-resource \  
--resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \  
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Hinzufügen von Tags zu einer Replikationsgruppe.

```
aws elasticache add-tags-to-resource \  
--resource-name arn:aws:elasticache:us-east-1:111111222233:replicationgroup:my-rg \  
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Erstellen eines Cache-Clusters mithilfe von Tags.

```
aws elasticache create-cache-cluster \  
--cluster-id testing-tags \  
--cluster-description cluster-test \  
--cache-subnet-group-name test \  
--cache-node-type cache.t2.micro \  
--engine valkey \  
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Erstellen eines Cache-Clusters mithilfe von Tags. In diesem Beispiel wird Redis als Engine verwendet.

```
aws elasticache create-cache-cluster \  
--cluster-id testing-tags \  
--cluster-description cluster-test \  
--cache-subnet-group-name test \  
--cache-node-type cache.t2.micro \  
--engine valkey \  
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Erstellen eines Serverless-Snapshots mit Tags. In diesem Beispiel wird Memcached als Engine verwendet.

```
aws elasticache create-serverless-cache-snapshot \  
--serverless-cache-name testing-tags \  
--serverless-cache-snapshot-name bkp-testing-tags-scs \  
--tags Key="work",Value="foo"
```

- Erstellen eines Snapshots mit Tags.

Snapshots sind derzeit nur für Redis verfügbar. Wenn Sie in diesem Fall Tags auf Anfrage hinzufügen, erhält der Snapshot nur die Anfrage-Tags, selbst wenn die Replikationsgruppe Tags enthält.

```
aws elasticache create-snapshot \  
--replication-group-id testing-tags \  
--snapshot-name bkp-testing-tags-rg \  
--tags Key="work",Value="foo"
```

Beispiele für Tag-basierte Zugriffssteuerungsrichtlinien

1. AddTagsToResource-Aktion für einen Cluster nur zulassen, wenn der Cluster das Tag „Project=XYZ“ hat.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "elasticache:AddTagsToResource",  
      "Resource": [  
        "arn:aws:elasticache:*:*:cluster:*"  
      ],  
      "Condition": {  
        "StringEquals": {  
          "aws:ResourceTag/Project": "XYZ"  
        }  
      }  
    }  
  ]  
}
```

2. Zulassen von RemoveTagsFromResource-Aktion von einer Replikationsgruppe, wenn sie die Tags „Project“ und „Service“ enthält und Schlüssel sich von „Project“ und „Service“ unterscheiden.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticache:RemoveTagsFromResource",
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Service": "Elasticache",
          "aws:ResourceTag/Project": "XYZ"
        },
        "ForAnyValue:StringNotEqualsIgnoreCase": {
          "aws:TagKeys": [
            "Project",
            "Service"
          ]
        }
      }
    }
  ]
}
```

3. Zulassen von AddTagsToResource zu einer Ressource nur dann, wenn sich Tags von „Project“ und „Service“ unterscheiden.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticache:AddTagsToResource",
      "Resource": [
        "arn:aws:elasticache:*:*:*:*"
      ],
      "Condition": {
        "ForAnyValue:StringNotEqualsIgnoreCase": {
```

```

        "aws:TagKeys": [
            "Service",
            "Project"
        ]
    }
}

```

4. Verweigerung CreateReplicationGroup-Aktion, wenn die Anfrage Tag Project=Foo enthält.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "elasticache:CreateReplicationGroup",
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "Foo"
        }
      }
    }
  ]
}

```

5. Verweigerung CopySnapshot-Aktion, wenn der Quell-Snapshot das Tag „Project=XYZ“ hat und das Anforderungs-Tag Service=Elasticache ist.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "elasticache:CopySnapshot",

```

```

        "Resource": [
            "arn:aws:elasticache:*:*:snapshot:*"
        ],
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/Project": "XYZ",
                "aws:RequestTag/Service": "Elasticache"
            }
        }
    }
}

```

6. Verweigern der Aktion `CreateCacheCluster`, wenn das Anforderungs-Tag `Project` fehlt oder nicht `Dev`, `QA` oder `Prod` entspricht.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*",
                "arn:aws:elasticache:*:*:securitygroup:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/Project": "true"
                }
            }
        }
    ]
}

```



```

    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:AddTagsToResource"
    ],
    "Resource": "arn:aws:elasticache:*:*:cluster:*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/Project": [
          "Dev",
          "Prod",
          "QA"
        ]
      }
    }
  }
]
}

```

Weitere Informationen zu Bedingungsschlüsseln finden Sie unter [Verwenden von Bedingungsschlüssel](#).

Überwachung von Kosten mit Kostenzuordnungs-Tags

Wenn Sie Ihren Ressourcen in Amazon Kostenzuordnungs-Tags hinzufügen ElastiCache, können Sie die Kosten verfolgen, indem Sie die Ausgaben auf Ihren Rechnungen nach Ressourcen-Tag-Werten gruppieren.

Ein ElastiCache Kostenzuweisungs-Tag ist ein Schlüssel-Wert-Paar, das Sie definieren und einer Ressource zuordnen. ElastiCache Bei Schlüsseln und Werten werden Groß-/Kleinschreibung berücksichtigt. Sie können einen Tag-Schlüssel verwenden, um eine Kategorie zu definieren, und der Tag-Wert kann ein Element in dieser Kategorie sein. So könnten Sie beispielsweise den Tag-Schlüssel `CostCenter` und den Tag-Wert `10010` definieren, um anzugeben, dass die Ressource der Kostenstelle 10010 zugewiesen ist. Sie können mit Tags auch Ressourcen kennzeichnen, die zu Test- oder Produktionszwecken verwendet werden, indem Sie einen Schlüssel wie z. B. `Environment` und Werte wie z. B. `test` oder `production` verwenden. Wir empfehlen, einheitliche

Tag-Schlüssel zu verwenden, um die mit Ihren Ressourcen verknüpften Kosten einfacher verfolgen zu können.

Verwenden Sie Kostenzuordnungs-Tags, um Ihre AWS Rechnung so zu organisieren, dass sie Ihrer eigenen Kostenstruktur entspricht. Melden Sie sich dazu an, um Ihre AWS Kontorechnung mit den Tag-Schlüsselwerten zu erhalten. Um dann die Kosten kombinierter Ressourcen anzuzeigen, organisieren Sie Ihre Fakturierungsinformationen nach Ressourcen mit gleichen Tag-Schlüsselwerten. Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen markieren und dann Ihre Fakturierungsinformationen so organisieren, dass Sie die Gesamtkosten dieser Anwendung über mehrere Services hinweg sehen können.

Sie können auch Tags miteinander kombinieren, um Kosten detaillierter zu verfolgen. Um beispielsweise Ihre Kosten für Services nach Region zu verfolgen, könnten Sie die Tag-Schlüssel Service und Region verwenden. Für eine Ressource lauten die Werte möglicherweise ElastiCache und Asia Pacific (Singapore) und für eine andere Ressource lauten sie ElastiCache und Europe (Frankfurt). Sie können dann Ihre ElastiCache Gesamtkosten nach Regionen aufgeschlüsselt sehen. Weitere Informationen finden Sie unter [Verwendung von Kostenzuordnungs-Tags](#) im AWS Billing-Benutzerhandbuch.

Sie können ElastiCache knotenbasierten ElastiCache Clustern Tags für die Kostenzuweisung hinzufügen. Wenn Sie ein Tag hinzufügen, auflisten, ändern, kopieren oder entfernen, wird die Operation nur auf die angegebenen Cluster angewendet.

Eigenschaften von ElastiCache Kostenzuweisungs-Tags

- Kostenzuweisungs-Tags werden auf ElastiCache Ressourcen angewendet, die in CLI- und API-Vorgängen als ARN angegeben sind. Der Ressourcentyp ist ein „Cluster“.

Beispiel-ARN: `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

- Der Tag-Schlüssel ist der erforderliche Name des Tags. Der Zeichenfolgenwert kann aus 1 bis 128 Unicode-Zeichen bestehen. Ihm darf kein `aws:` als Präfix vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen, Unterstriche (`_`), Punkte (`.`), Doppelpunkte (`:`), Backslashes (`\`), Gleichheitszeichen (`=`), Pluszeichen (`+`), Trennstriche (`-`) oder At-Zeichen (`@`) enthalten.

- Der Tag-Wert ist der optionale Wert des Tags. Der Zeichenfolgenwert kann aus 1 bis 256 Unicode-Zeichen bestehen. Ihm darf kein `aws :` als Präfix vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen, Unterstriche (`_`), Punkte (`.`), Doppelpunkte (`:`), Backslashes (`\`), Gleichheitszeichen (`=`), Pluszeichen (`+`), Trennstriche (`-`) oder At-Zeichen (`@`) enthalten.
- Eine ElastiCache Ressource kann maximal 50 Tags haben.
- Die Werte innerhalb eines Tag-Satzes müssen nicht eindeutig sein. Beispiel: In einem Tag-Satz könnten die Schlüssel `Service` und `Application` beide den Wert `ElastiCache` besitzen.

AWS wendet Ihren Tags keine semantische Bedeutung an. Tags werden ausschließlich als Zeichenketten interpretiert. AWS setzt nicht automatisch irgendwelche Tags für eine ElastiCache Ressource.

Verwaltung Ihrer Kostenzuweisungs-Tags mithilfe der AWS CLI

Sie können die verwenden, AWS CLI um Kostenzuordnungs-Tags hinzuzufügen, zu ändern oder zu entfernen.

Kostenzuweisungs-Tags werden auf ElastiCache Cluster angewendet. Der Cluster, der gekennzeichnet werden soll, wird mit einem ARN (Amazon Resource Name) angegeben.

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Themen

- [Stichwörter auflisten mit dem AWS CLI](#)
- [Hinzufügen von Tags mit dem AWS CLI](#)
- [Ändern von Tags mit dem AWS CLI](#)
- [Entfernen von Tags mit dem AWS CLI](#)

Stichwörter auflisten mit dem AWS CLI

Sie können die verwenden AWS CLI , um Tags für eine vorhandene ElastiCache Ressource aufzulisten, indem Sie die [list-tags-for-resource](#) Operation verwenden.

Der folgende Code verwendet die AWS CLI , um die Tags auf dem Memcached-Cluster `my-cluster` in der Region `us-west-2` aufzulisten.

Für Linux, macOS oder Unix:

```
aws elasticache list-tags-for-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Für Windows:

```
aws elasticache list-tags-for-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Der folgende Code verwendet die AWS CLI , um die Tags auf dem Valkey- oder Redis-OSS-Knoten `my-cluster-001` im `my-cluster` Cluster in der Region `us-west-2` aufzulisten.

Für Linux, macOS oder Unix:

```
aws elasticache list-tags-for-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

Für Windows:

```
aws elasticache list-tags-for-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus und besteht aus einer Liste aller Tags für die Ressource.

```
{  
  "TagList": [  
    {  
      "Value": "10110",  
      "Key": "CostCenter"  
    },  
    {  
      "Value": "EC2",  
      "Key": "Service"  
    }  
  ]  
}
```

```
}
```

Wenn die Ressource keine Tags enthält, ist die Ausgabe leer. TagList

```
{  
  "TagList": []  
}
```

Weitere Informationen finden Sie unter AWS CLI für ElastiCache [list-tags-for-resource](#).

Hinzufügen von Tags mit dem AWS CLI

Sie können das verwenden AWS CLI , um einer vorhandenen ElastiCache Ressource mithilfe der [add-tags-to-resource](#) CLI-Operation Tags hinzuzufügen. Wenn das Tag in der Ressource nicht vorhanden ist, werden Schlüssel und Wert zur Ressource hinzugefügt. Wenn der Schlüssel in der Ressource bereits vorhanden ist, wird der diesem Schlüssel zugeordnete Wert auf den neuen Wert aktualisiert.

Der folgende Code verwendet die AWS CLI , um die Schlüssel Service und Region mit den Werten `elasticache` `us-west-2` jeweils zum Knoten `my-cluster-001` im Cluster `my-cluster` in der Region `us-west-2` hinzuzufügen.

Memcached

Für Linux, macOS oder Unix:

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \  
  --tags Key=Service,Value=elasticache \  
         Key=Region,Value=us-west-2
```

Für Windows:

```
aws elasticache add-tags-to-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^  
  --tags Key=Service,Value=elasticache ^  
         Key=Region,Value=us-west-2
```

Redis

Für Linux, macOS oder Unix:

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \  
  --tags Key=Service,Value=elasticache \  
        Key=Region,Value=us-west-2
```

Für Windows:

```
aws elasticache add-tags-to-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^  
  --tags Key=Service,Value=elasticache ^  
        Key=Region,Value=us-west-2
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus und besteht aus einer Liste aller Tags für die Ressource nach der Operation.

```
{  
  "TagList": [  
    {  
      "Value": "elasticache",  
      "Key": "Service"  
    },  
    {  
      "Value": "us-west-2",  
      "Key": "Region"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter für AWS CLI ElastiCache [add-tags-to-resource](#)

Sie können auch verwenden AWS CLI, um einem Cluster Tags hinzuzufügen, wenn Sie mithilfe des Vorgangs einen neuen Cluster erstellen [create-cache-cluster](#). Sie können keine Tags hinzufügen, wenn Sie einen Cluster mithilfe der ElastiCache Managementkonsole erstellen. Nachdem der Cluster erstellt wurde, können Sie über die Konsole Tags zum Cluster hinzufügen.

Ändern von Tags mit dem AWS CLI

Sie können den verwenden AWS CLI, um die Tags auf einem ElastiCache Cluster zu ändern.

Ändern Sie Tags wie folgt:

- Verwenden Sie [add-tags-to-resource](#) zum Hinzufügen eines neuen Tags und Wertes oder zum Ändern des zu einem vorhandenen Tag gehörenden Wertes.
- Verwenden Sie [remove-tags-from-resource](#) zum Entfernen angegebener Tags aus der Ressource.

zum Entfernen von Tags aus dem Cluster.

Entfernen von Tags mit dem AWS CLI

Sie können den verwenden AWS CLI , um Tags aus einem vorhandenen ElastiCache Memcached-Cluster zu entfernen, indem Sie den [remove-tags-from-resource](#) Vorgang verwenden.

Für Memcached verwendet der folgende Code die, AWS CLI um die Tags mit den Schlüsseln Service und Region aus dem Knoten my-cluster-001 im Cluster my-cluster in der Region us-west-2 zu entfernen.

Für Linux, macOS oder Unix:

```
aws elasticache remove-tags-from-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \  
  --tag-keys PM Service
```

Für Windows:

```
aws elasticache remove-tags-from-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^  
  --tag-keys PM Service
```

Für Redis OSS verwendet der folgende Code die, AWS CLI um die Tags mit den Schlüsseln Service und Region vom Knoten my-cluster-001 im Cluster my-cluster in der Region us-west-2 zu entfernen.

Für Linux, macOS oder Unix:

```
aws elasticache remove-tags-from-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \  
  --tag-keys PM Service
```

Für Windows:

```
aws elasticache remove-tags-from-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^  
  --tag-keys PM Service
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus und besteht aus einer Liste aller Tags für die Ressource nach der Operation.

```
{  
  "TagList": []  
}
```

Weitere Informationen finden Sie unter für AWS CLI ElastiCache [remove-tags-from-resource](#)

Verwaltung Ihrer Kostenzuweisungs-Tags mithilfe der ElastiCache API

Sie können die ElastiCache API verwenden, um Kostenzuweisungs-Tags hinzuzufügen, zu ändern oder zu entfernen.

Kostenzuweisungs-Tags werden ElastiCache für Memcached-Cluster angewendet. Der Cluster, der gekennzeichnet werden soll, wird mit einem ARN (Amazon Resource Name) angegeben.

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Themen

- [Tags mithilfe der API auflisten ElastiCache](#)
- [Hinzufügen von Tags mithilfe der API ElastiCache](#)
- [Ändern von Tags mithilfe der ElastiCache API](#)
- [Tags mithilfe der ElastiCache API entfernen](#)

Tags mithilfe der API auflisten ElastiCache

Sie können die ElastiCache API verwenden, um Tags für eine vorhandene Ressource aufzulisten, indem Sie den [ListTagsForResource](#) Vorgang verwenden.

Für Memcached verwendet der folgende Code die ElastiCache API, um die Tags auf der Ressource `my-cluster` in der Region `us-west-2` aufzulisten.


```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListTagsForResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Version=2015-02-02  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Für Redis OSS verwendet der folgende Code die ElastiCache API, um die Tags auf der Ressource `my-cluster-001` in der Region `us-west-2` aufzulisten.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListTagsForResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Version=2015-02-02  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Hinzufügen von Tags mithilfe der API ElastiCache

Sie können die ElastiCache API verwenden, um einem vorhandenen ElastiCache Cluster Tags hinzuzufügen, indem Sie den [AddTagsToResource](#) Vorgang verwenden. Wenn das Tag in der Ressource nicht vorhanden ist, werden Schlüssel und Wert zur Ressource hinzugefügt. Wenn der Schlüssel in der Ressource bereits vorhanden ist, wird der diesem Schlüssel zugeordnete Wert auf den neuen Wert aktualisiert.

Der folgende Code verwendet die ElastiCache API, um die Schlüssel Service und Region mit den Werten `elasticache us-west-2` bzw. Bei Memcached wird dies auf die Ressource angewendet. `my-cluster` Für Redis OSS wird dies auf die Ressource `my-cluster-001` in der Region `US-West-2` angewendet.

Memcached

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=AddTagsToResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster  
&SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256
&Tags.member.1.Key=Service
&Tags.member.1.Value=elasticache
&Tags.member.2.Key=Region
&Tags.member.2.Value=us-west-2
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Redis

```
https://elasticache.us-west-2.amazonaws.com/
?Action=AddTagsToResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Tags.member.1.Key=Service
&Tags.member.1.Value=elasticache
&Tags.member.2.Key=Region
&Tags.member.2.Value=us-west-2
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie [AddTagsToResource](#) in der Amazon ElastiCache API-Referenz.

Ändern von Tags mithilfe der ElastiCache API

Sie können die ElastiCache API verwenden, um die Tags in einem ElastiCache Cluster zu ändern.

Den Wert eines Tags ändern:

- Verwenden Sie die Operation [AddTagsToResource](#) zum Hinzufügen eines neuen Tags und Wertes oder zum Ändern des Wertes eines vorhandenen Tags.
- Verwenden Sie [RemoveTagsFromResource](#) zum Entfernen von Tags aus der Ressource.

Die Ausgabe beider Operationen ist eine Liste der Tags und deren Werte für die angegebene Ressource.

Verwenden Sie [RemoveTagsFromResource](#) zum Entfernen von Tags aus der Ressource.

Tags mithilfe der ElastiCache API entfernen

Sie können die ElastiCache API verwenden, um Tags aus einem vorhandenen ElastiCache Memcached-Cluster zu entfernen, indem Sie den [RemoveTagsFromResource](#) Vorgang verwenden.

Der folgende Code verwendet die ElastiCache API, um die Tags mit den Schlüsseln Service und Region vom Knoten my-cluster-001 im Cluster my-cluster in der Region us-west-2 zu entfernen.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RemoveTagsFromResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&TagKeys.member.1=Service  
&TagKeys.member.2=Region  
&Version=2015-02-02  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Verwenden des Amazon ElastiCache Well-Architected-Objektivs

In diesem Abschnitt wird Amazon ElastiCache Well-Architected Lens beschrieben, eine Sammlung von Entwurfsprinzipien und Anleitungen für die Gestaltung gut ElastiCache strukturierter Workloads.

- Die ElastiCache Linse ist eine Ergänzung zum [AWS Well-Architected Framework](#).
- Jede Säule hat eine Reihe von Fragen, die helfen sollen, die Diskussion rund um eine ElastiCache Architektur zu beginnen.
 - Jede Frage enthält eine Reihe von führenden Methoden sowie deren Ergebnisse für die Berichterstattung.
 - Erforderlich – notwendig vor dem Wechsel in die Produktion (sofern kein hohes Risiko besteht)
 - Am besten – bestmöglicher Zustand für einen Kunden
 - Gut – unsere Empfehlungen für den Kunden (sofern kein mittleres Risiko besteht)
- Well-Architected-Terminologie
 - [Komponente](#) — Code, Konfiguration und AWS Ressourcen, die zusammen eine Anforderung erfüllen. Komponenten interagieren mit anderen Komponenten und entsprechen in Microservice-Architekturen häufig einem Service.

- [Arbeitslast](#) — Eine Reihe von Komponenten, die zusammen einen geschäftlichen Nutzen bieten. Beispiele für Workloads sind Marketing-Websites, E-Commerce-Websites, das Backend für eine mobile App, Analyseplattformen usw.

 Note

Dieses Handbuch wurde nicht aktualisiert und enthält nun Informationen zum ElastiCache serverlosen Caching und zur neuen Valkey-Engine.

Themen

- [Säule Operational Excellence von Amazon ElastiCache Well-Architected Lens](#)
- [Sicherheitssäule von Amazon ElastiCache Well-Architected Lens](#)
- [Säule der Zuverlässigkeit von Amazon ElastiCache Well-Architected Lens](#)
- [Säule der Leistung und Effizienz von Amazon ElastiCache Well-Architected Lens](#)
- [Säule zur ElastiCache Kostenoptimierung bei Amazon Well-Architected Lens](#)

Säule Operational Excellence von Amazon ElastiCache Well-Architected Lens

Die Säule „Operational Excellence“ konzentriert sich auf den Betrieb und die Überwachung von Systemen, um einen Mehrwert für das Unternehmen zu schaffen, sowie auf die kontinuierliche Verbesserung von Prozessen und Verfahren. Zu den wichtigsten Themen gehören die Automatisierung von Änderungen, die Reaktion auf Ereignisse und die Definition von Standards für die Verwaltung des täglichen Betriebs.

Themen

- [OE 1: Wie verstehen Sie Warnmeldungen und Ereignisse, die von Ihrem ElastiCache Cluster ausgelöst werden, und wie reagieren Sie darauf?](#)
- [OE 2: Wann und wie skalieren Sie Ihre vorhandenen Cluster? ElastiCache](#)
- [OE 3: Wie verwalten Sie Ihre ElastiCache Cluster-Ressourcen und pflegen Ihren Cluster up-to-date?](#)
- [OE 4: Wie verwalten Sie die Verbindungen der Kunden zu Ihren ElastiCache Clustern?](#)

- [OE 5: Wie werden ElastiCache Komponenten für einen Workload bereitgestellt?](#)
- [OE 6: Wie planen und minimieren Sie Ausfälle?](#)
- [OE 7: Wie behebt man Valkey- oder Redis OSS-Engine-Ereignisse?](#)

OE 1: Wie verstehen Sie Warnmeldungen und Ereignisse, die von Ihrem ElastiCache Cluster ausgelöst werden, und wie reagieren Sie darauf?

Einführung auf Fragenebene: Wenn Sie ElastiCache Cluster betreiben, können Sie optional Benachrichtigungen und Warnmeldungen erhalten, wenn bestimmte Ereignisse eintreten. ElastiCache protokolliert standardmäßig [Ereignisse](#), die sich auf Ihre Ressourcen beziehen, z. B. Failover, Knotenaustausch, Skalierungsvorgänge, geplante Wartungsarbeiten und mehr. Jedes Ereignis enthält das Datum und die Uhrzeit, den Quellnamen und den Quelltyp sowie eine Beschreibung.

Vorteil auf Fragenebene: Wenn Sie in der Lage sind, die zugrundeliegenden Ursachen für die Ereignisse, die von Ihrem Cluster generierte Warnmeldungen auslösen, zu verstehen und zu verwalten, können Sie effektiver arbeiten und angemessen auf Ereignisse reagieren.

- [Erforderlich] [Überprüfen Sie die Ereignisse, die ElastiCache auf der ElastiCache Konsole \(nachdem Sie Ihre Region ausgewählt haben\) oder mithilfe des Befehls describe-events \(CLI\) von Amazon Command Line Interface \(AWS CLI\) und der ElastiCache API generiert wurden.](#) Konfigurieren Sie ElastiCache das Senden von Benachrichtigungen für wichtige Cluster-Ereignisse mithilfe von Amazon Simple Notification Service (Amazon SNS). Wenn Sie Amazon SNS mit Ihren Clustern verwenden, können Sie bei Ereignissen programmgesteuert Maßnahmen ergreifen.
- ElastiCache

- Es gibt zwei große Ereigniskategorien: aktuelle und geplante Ereignisse. Die Liste der aktuellen Ereignisse umfasst: Erstellung und Löschung von Ressourcen, Skalierungsvorgänge, Failover, Neustart des Knotens, Erstellung eines Snapshots, Änderung der Cluster-Parameter, Verlängerung des CA-Zertifikats, Fehlerereignisse (Fehler bei der Cluster-Bereitstellung – VPC oder ENI, Skalierungsfehler – ENI- und Snapshot-Fehler). Die Liste der geplanten Ereignisse umfasst: Knoten, dessen Austausch während des Wartungsfensters geplant ist, und Knotenaustausch, der verschoben wurde.
- Auch wenn Sie auf einige dieser Ereignisse möglicherweise nicht sofort reagieren müssen, ist es wichtig, sich zunächst alle Fehlerereignisse anzusehen:
 - ElastiCache:AddCacheNodeFailed
 - ElastiCache:CacheClusterProvisioningFailed

- ElastiCache:CacheClusterScalingFailed
 - ElastiCache:CacheNodesRebooted
 - ElastiCache: SnapshotFailed (Nur Valkey oder Redis OSS)
 - [Ressourcen]:
 - [Verwaltung von ElastiCache Amazon SNS SNS-Benachrichtigungen](#)
 - [Ereignisbenachrichtigungen und Amazon SNS](#)
 - [Am besten] Nutzen Sie AWS Produkt- und Servicefunktionen wie SNS und Lambda Functions, um Reaktionen auf Ereignisse zu automatisieren. Halten Sie sich an bewährte Methoden, indem Sie kleine, häufige, umkehrbare Änderungen als Code vornehmen, um Ihre Operationen im Laufe der Zeit weiterzuentwickeln. Sie sollten CloudWatch Amazon-Metriken verwenden, um Ihre Cluster zu überwachen.
- [Ressourcen]: [Überwachen Sie ElastiCache \(Cluster-Modus deaktiviert\) Lesereplikat-Endpunkte mit AWS Lambda, Amazon Route 53 und Amazon SNS für einen Anwendungsfall](#), der Lambda und SNS verwendet.

OE 2: Wann und wie skalieren Sie Ihre vorhandenen Cluster? ElastiCache

Einführung auf Fragenebene: Die richtige Größe Ihres ElastiCache Clusters ist ein Balanceakt, der bei jeder Änderung der zugrunde liegenden Workload-Typen bewertet werden muss. Ihr Ziel ist es, mit der richtigen Größe für Ihren Workload zu arbeiten.

Vorteil auf Fragenebene: Eine Überlastung Ihrer Ressourcen kann zu einer erhöhten Latenz und einer insgesamt verringerten Leistung führen. Eine Unterauslastung kann andererseits zu einer Überbereitstellung von Ressourcen bei nicht optimaler Kostenoptimierung führen. Durch die richtige Dimensionierung Ihrer Umgebungen können Sie ein Gleichgewicht zwischen Leistungseffizienz und Kostenoptimierung erreichen. Um eine Über- oder Unterauslastung Ihrer Ressourcen zu beheben, ElastiCache können Sie in zwei Dimensionen skalieren. Sie können vertikal skalieren, indem Sie die Knotenkapazität erhöhen oder verringern. Sie können auch horizontal skalieren, indem Sie Knoten hinzufügen und entfernen.

- [Erforderlich] Die Überauslastung der CPU und des Netzwerks auf den Primärknoten sollte dadurch behoben werden, dass die Lesevorgänge ausgelagert und an die Replikatknoten umgeleitet werden. Verwenden Sie Replikatknoten für Lesevorgänge, um die Auslastung des Primärknotens zu reduzieren. Dies kann in Ihrer Valkey- oder Redis OSS-Clientbibliothek konfiguriert werden, indem Sie eine Verbindung zum ElastiCache Reader-Endpunkt herstellen, wenn der Clustermodus

deaktiviert ist, oder indem Sie den Befehl READONLY verwenden, wenn der Clustermodus aktiviert ist.

[Ressourcen]:

- [Verbindungsendpunkte finden in ElastiCache](#)
- [Richtige Clustergröße](#)
- [READONLY-Befehl](#)
- [Erforderlich] Überwachen Sie die Auslastung kritischer Clusterressourcen wie CPU, Arbeitsspeicher und Netzwerk. Die Auslastung dieser spezifischen Clusterressourcen muss nachverfolgt werden, um eine Entscheidung für eine Skalierung und die Art des Skalierungsvorgangs treffen zu können. Wenn der ElastiCache Clustermodus deaktiviert ist, können Primär- und Replikatknoten vertikal skaliert werden. Replikatknoten können auch horizontal von 0 auf 5 Knoten skaliert werden. Wenn der Clustermodus aktiviert ist, gilt dasselbe für jeden Shard Ihres Clusters. Darüber hinaus können Sie die Anzahl der Shards erhöhen oder reduzieren.

[Ressourcen]:

- [Überwachung von Best Practices ElastiCache mithilfe von Amazon CloudWatch](#)
- [Skalierung von ElastiCache Clustern für Valkey und Redis OSS](#)
- [Skalierung von ElastiCache Clustern für Memcached](#)
- [Am besten] Wenn Sie Trends im Zeitverlauf überwachen, können Sie Workload-Änderungen erkennen, die bei punktueller Überwachung unbemerkt bleiben würden. Um längerfristige Trends zu erkennen, verwenden Sie CloudWatch Metriken, um nach längeren Zeiträumen zu suchen. Die Erkenntnisse aus der Beobachtung von CloudWatch Kennzahlen über längere Zeiträume sollten als Grundlage für Ihre Prognose zur Auslastung der Cluster-Ressourcen dienen. CloudWatch Datenpunkte und Metriken sind für bis zu 455 Tage verfügbar.

[Ressourcen]:

- [Überwachung ElastiCache mit CloudWatch Metriken](#)
- [Überwachung von Memcached mit Metriken CloudWatch](#)
- [Überwachung von Best Practices ElastiCache mithilfe von Amazon CloudWatch](#)
- [Optimal] Wenn Ihre ElastiCache Ressourcen damit erstellt werden, empfiehlt CloudFormation es sich, Änderungen mithilfe von CloudFormation Vorlagen vorzunehmen, um die betriebliche Konsistenz zu wahren und unverwaltete Konfigurationsänderungen und Stack-Drifts zu vermeiden.

[Ressourcen]:

- [ElastiCache Referenz zum Ressourcentyp für CloudFormation](#)
- [Am besten] Automatisieren Sie Ihre Skalierungsvorgänge mithilfe von Cluster-Betriebsdaten und definieren Sie Schwellenwerte für CloudWatch die Einrichtung von Alarmen. Verwenden Sie CloudWatch Events und Simple Notification Service (SNS), um Lambda-Funktionen auszulösen, und führen Sie eine ElastiCache API aus, um Ihre Cluster automatisch zu skalieren. Ein Beispiel wäre, Ihrem Cluster einen Shard hinzuzufügen, wenn die Metrik `EngineCPUUtilization` über einen längeren Zeitraum 80 % erreicht. Eine weitere Option wäre die Verwendung von `DatabaseMemoryUsedPercentages` für einen speicherbasierten Schwellenwert.

[Ressourcen]:

- [Amazon CloudWatch Alarms verwenden](#)
- [Was sind CloudWatch Amazon-Events?](#)
- [Verwendung AWS Lambda mit Amazon Simple Notification Service](#)
- [ElastiCache API Referenz](#)

OE 3: Wie verwalten Sie Ihre ElastiCache Cluster-Ressourcen und pflegen Ihren Cluster up-to-date?

Einführung auf Fragenebene: Wenn Sie in großem Maßstab arbeiten, ist es wichtig, dass Sie in der Lage sind, all Ihre Ressourcen zu lokalisieren und zu identifizieren. ElastiCache Bei der Einführung neuer Anwendungsfunktionen müssen Sie für eine Symmetrie der Cluster-Versionen in all Ihren ElastiCache Umgebungstypen sorgen: Entwicklung, Testen und Produktion. Mithilfe von Ressourcenattributen können Sie Umgebungen für unterschiedliche betriebliche Ziele trennen, z. B. bei der Einführung neuer Funktionen und der Aktivierung neuer Sicherheitsmechanismen.

Vorteil auf Fragenebene: Die Trennung Ihrer Entwicklungs-, Test- und Produktionsumgebungen ist die beste betriebliche Methode. Es ist auch eine bewährte Methode, dass auf Ihren Clustern und Knoten in allen Umgebungen die neuesten Softwarepatches installiert werden, wobei wohlverstandene und dokumentierte Prozesse verwendet werden. Durch die Nutzung systemeigener ElastiCache Funktionen kann sich Ihr Entwicklungsteam auf die Erreichung der Geschäftsziele konzentrieren und nicht auf die ElastiCache Wartung.

- [Am besten] Verwenden Sie die neueste verfügbare Engine-Version und installieren Sie die Self-Service-Updates, sobald sie verfügbar sind. ElastiCache aktualisiert die zugrunde liegende Infrastruktur automatisch während des angegebenen Wartungsfensters des Clusters. Die in Ihren Clustern laufenden Knoten werden jedoch über Self-Service-Updates aktualisiert. Es gibt zwei

Arten von Updates: Sicherheitspatches oder kleinere Software-Updates. Stellen Sie sicher, dass Sie den Unterschied zwischen den Patch-Typen verstehen und wissen, wann diese angewendet werden.

[Ressourcen]:

- [Self-Service-Updates bei Amazon ElastiCache](#)
- [Hilfeseite für Amazon ElastiCache Managed Maintenance and Service Updates](#)
- [Am besten] Organisieren Sie Ihre ElastiCache Ressourcen mithilfe von Tags. Verwenden Sie Tags für Replikationsgruppen und nicht für einzelne Knoten. Sie können Tags so konfigurieren, dass sie angezeigt werden, wenn Sie Ressourcen abfragen, und Sie können Tags verwenden, um Suchen durchzuführen und Filter anzuwenden. Sie sollten Ressourcengruppen verwenden, um auf einfache Weise Sammlungen von Ressourcen zu erstellen und zu verwalten, die gemeinsame Tag-Sätze verwenden.

[Ressourcen]:

- [Bewährte Methoden zum Taggen](#)
- [ElastiCache Referenz zum Ressourcentyp für CloudFormation](#)
- [Parametergruppen](#)

OE 4: Wie verwalten Sie die Verbindungen der Kunden zu Ihren ElastiCache Clustern?

Einführung auf Fragenebene: Wenn Sie in großem Maßstab arbeiten, müssen Sie verstehen, wie Ihre Kunden mit dem ElastiCache Cluster kommunizieren, um die betrieblichen Aspekte Ihrer Anwendung (z. B. Reaktionszeiten) verwalten zu können.

Vorteil auf Fragenebene: Durch die Auswahl des am besten geeigneten Verbindungsmechanismus wird sichergestellt, dass Ihre Anwendung nicht aufgrund von Verbindungsfehlern, wie Timeouts, unterbrochen wird.

- [Erforderlich] Trennen Sie Lese- und Schreibvorgänge und stellen Sie eine Verbindung mit dem Replikatknoten her, um Lesevorgänge auszuführen. Beachten Sie jedoch, dass Sie, wenn Sie die Schreibvorgänge von den Lesevorgängen trennen, aufgrund der asynchronen Natur der Valkey- und Redis-OSS-Replikation die Fähigkeit verlieren, einen Schlüssel unmittelbar nach dem Schreiben zu lesen. Der WAIT-Befehl kann genutzt werden, um die Datensicherheit in der Praxis zu verbessern und Replikate dazu zu zwingen, Schreibvorgänge zu bestätigen, bevor sie auf Clients antworten, was die Gesamtleistung beeinträchtigt. Die Verwendung von Replikatknoten für Lesevorgänge kann in Ihrer ElastiCache Client-Bibliothek konfiguriert werden, indem der

ElastiCache Reader-Endpunkt für den Clustermodus deaktiviert ist. Wenn der Clustermodus aktiviert ist, verwenden Sie den Befehl READONLY. Für viele der ElastiCache Clientbibliotheken ist READONLY standardmäßig oder über eine Konfigurationseinstellung implementiert.

[Ressourcen]:

- [Verbindungsendpunkte finden in ElastiCache](#)
- [READONLY](#)
- [Erforderlich] Verwenden Sie Verbindungspooling. Der Aufbau einer TCP-Verbindung kostet sowohl auf Client- als auch auf Serverseite CPU-Zeit. Durch Pooling können Sie die TCP-Verbindung wiederverwenden.

Um den Verbindungsaufwand zu reduzieren, sollten Sie Verbindungspooling verwenden. Mit einem Pool von Verbindungen kann Ihre Anwendung Verbindungen „nach Belieben“ wiederverwenden und freigeben, ohne dass Kosten für den Verbindungsaufbau anfallen. Sie können das Verbindungspooling über Ihre ElastiCache Clientbibliothek (sofern unterstützt) implementieren, wobei ein Framework für Ihre Anwendungsumgebung verfügbar ist, oder es von Grund auf neu erstellen.

- [Am besten] Stellen Sie sicher, dass das Socket-Timeout des Clients auf mindestens eine Sekunde eingestellt ist (im Vergleich zur typischen Standardeinstellung „Keine“ bei verschiedenen Clients).
- Wenn Sie den Timeout-Wert zu niedrig einstellen, kann dies zu möglichen Timeouts führen, wenn die Serverlast hoch ist. Eine zu hohe Einstellung kann dazu führen, dass Ihre Anwendung lange braucht, um Verbindungsprobleme zu erkennen.
- Kontrollieren Sie das Volumen neuer Verbindungen, indem Sie Verbindungspooling in Ihrer Client-Anwendung implementieren. Dadurch werden Latenz und CPU-Auslastung reduziert, die erforderlich sind, um Verbindungen zu öffnen und zu schließen, und es wird ein TLS-Handshake durchgeführt, wenn TLS im Cluster aktiviert ist.

[Ressourcen]: ElastiCache [Für höhere Verfügbarkeit konfigurieren](#)

- [Gut] Der Einsatz von Pipelining (sofern Ihre Anwendungsfälle dies zulassen) kann die Leistung erheblich steigern.
- Mit Pipelining reduzieren Sie die Round-Trip Time (RTT, Roundtrip-Zeit) zwischen Ihren Anwendungsclients und dem Cluster und neue Anfragen können verarbeitet werden, auch wenn der Client die vorherigen Antworten noch nicht gelesen hat.
- Mit Pipelining können Sie mehrere Befehle an den Server senden, ohne auf Antworten/Bestätigungen warten zu müssen. Der Nachteil von Pipelining ist, dass, wenn Sie irgendwann

alle Antworten in Massen abrufen, möglicherweise ein Fehler aufgetreten ist, den Sie erst am Ende erkennen können.

- Implementieren Sie Methoden, um Anfragen erneut zu versuchen, wenn ein Fehler zurückgegeben wird, der die fehlerhafte Anfrage auslöst.

[Ressourcen]: [Pipelining](#)

OE 5: Wie werden ElastiCache Komponenten für einen Workload bereitgestellt?

Einführung auf Fragenebene: ElastiCache Umgebungen können manuell über die AWS Konsole oder programmgesteuert über CLI/APIs, Toolkits usw. bereitgestellt werden. Bewährte Methoden für Operational Excellence sehen vor, Bereitstellungen möglichst mithilfe von Code zu automatisieren. Darüber hinaus können ElastiCache Cluster entweder nach Arbeitslast isoliert oder zur Kostenoptimierung kombiniert werden.

Vorteil auf Frageebene: Durch die Auswahl des für Ihre ElastiCache Umgebungen am besten geeigneten Bereitstellungsmechanismus kann Operation Excellence im Laufe der Zeit verbessert werden. Es wird empfohlen, Operationen möglichst als Code auszuführen, um menschliche Fehler zu minimieren und die Wiederholbarkeit, Flexibilität und Reaktionszeit auf Ereignisse zu erhöhen.

Wenn Sie die Anforderungen an die Workload-Isolierung verstehen, können Sie entscheiden, ob Sie pro Workload dedizierte ElastiCache Umgebungen einrichten oder mehrere Workloads zu einzelnen Clustern oder Kombinationen daraus zusammenfassen möchten. Die Nachteile zu verstehen, kann dazu beitragen, ein Gleichgewicht zwischen Operational Excellence und Kostenoptimierung zu finden

- [Erforderlich] Machen Sie sich mit den verfügbaren Bereitstellungsoptionen vertraut und automatisieren Sie diese Verfahren ElastiCache, wann immer dies möglich ist. Zu den möglichen Automatisierungsmöglichkeiten gehören CloudFormation AWS CLI/SDK und APIs

[Ressourcen]:

- [Referenz zum ElastiCache Amazon-Ressourcentyp](#)
- [elasticache](#)
- [ElastiCache Amazon-API-Referenz](#)
- [Erforderlich] Ermitteln Sie für alle Workloads den erforderlichen Grad der Clusterisolierung.
- [Am besten]: Hohe Isolierung – eine 1:1-Zuordnung zwischen Workload und Cluster. Ermöglicht die feinste Kontrolle über den Zugriff, die Größe, Skalierung und Verwaltung von ElastiCache Ressourcen auf Workload-Basis.

- [Besser]: Mittlere Isolierung – M:1 ist zweckmäßig isoliert, aber möglicherweise von mehreren Workloads gemeinsam genutzt (z. B. ein Cluster, der für das Zwischenspeichern von Workloads vorgesehen ist, und ein anderer, der für Messaging dediziert ist).
- [Gut]: Niedrige Isolierung – M:1 für alle Zwecke, vollständig gemeinsam genutzt. Empfohlen für Workloads, bei denen gemeinsamer Zugriff akzeptabel ist.

OE 6: Wie planen und minimieren Sie Ausfälle?

Einführung auf Fragenebene: Operational Excellence umfasst die Antizipation von Ausfällen durch regelmäßige „Pre-Mortem“-Übungen zur Identifizierung potenzieller Fehlerquellen, sodass sie behoben oder gemildert werden können. ElastiCache bietet eine Failover-API, die zu Testzwecken simulierte Knotenausfälle ermöglicht.

Vorteil auf Fragenebene: Wenn Sie Ausfallszenarien vorab testen, können Sie herausfinden, wie sie sich auf Ihren Workload auswirken. Dies ermöglicht ein sicheres Testen der Reaktionsverfahren und ihrer Wirksamkeit und macht Ihr Team mit deren Ausführung vertraut.

[Erforderlich] Führen Sie regelmäßig Failover-Tests in Entwickler-/Testkonten durch. [TestFailover](#)

OE 7: Wie behebt man Valkey- oder Redis OSS-Engine-Ereignisse?

Einführung auf Fragenebene: Operational Excellence erfordert die Fähigkeit, sowohl Informationen auf Service- als auch auf Engine-Ebene zu untersuchen, um den Zustand und den Status Ihrer Cluster zu analysieren. ElastiCache kann Valkey- oder Redis-OSS-Engine-Protokolle sowohl an Amazon als auch an Amazon Kinesis CloudWatch Data Firehose senden.

Vorteil auf Fragenebene: Durch die Aktivierung von Valkey- oder Redis-OSS-Engine-Protokollen auf ElastiCache Clustern erhalten Sie Einblicke in Ereignisse, die sich auf den Zustand und die Leistung von Clustern auswirken. Die Logs der Valkey- oder Redis OSS-Engine liefern Daten direkt von der Engine, die über den Ereignismechanismus nicht verfügbar sind. ElastiCache Durch sorgfältige Beobachtung sowohl der ElastiCache Ereignisse (siehe oben in OE-1) als auch der Engine-Logs ist es möglich, bei der Fehlerbehebung eine Reihenfolge der Ereignisse sowohl aus ElastiCache Service- als auch aus Engine-Perspektive zu bestimmen.

- [Erforderlich] Stellen Sie sicher, dass die Protokollierungsfunktion der Redis OSS-Engine aktiviert ist. Diese Funktion ist ab ElastiCache Version 6.2 für Redis OSS und neuer verfügbar. Dieser Vorgang kann während der Clustererstellung oder durch Änderung des Clusters nach der Erstellung erfolgen.

- Stellen Sie fest, ob Amazon CloudWatch Logs oder Amazon Kinesis Data Firehose das geeignete Ziel für Redis OSS-Engine-Protokolle ist.
- Wählen Sie ein geeignetes Zielprotokoll in CloudWatch oder Kinesis Data Firehose aus, um die Protokolle beizubehalten. Wenn Sie mehrere Cluster haben, sollten Sie für jeden Cluster ein anderes Zielprotokoll verwenden, da Daten bei der Fehlerbehebung so besser isoliert werden können.

[Ressourcen]:

- Protokollzustellung: [Protokollzustellung](#)
- Ziele für die Protokollierung: [Amazon CloudWatch Logs](#)
- Einführung in Amazon CloudWatch Logs: [Was ist Amazon CloudWatch Logs?](#)
- Einführung in Amazon Kinesis Data Firehose: [Was ist Amazon Kinesis Data Firehose?](#)
- [Am besten] Wenn Sie Amazon CloudWatch Logs verwenden, sollten Sie Amazon CloudWatch Logs Insights nutzen, um das Valkey- oder Redis OSS-Engine-Protokoll nach wichtigen Informationen abzufragen.

Erstellen Sie als Beispiel eine Abfrage für die CloudWatch Log-Gruppe, die die Logs der Valkey- oder Redis-OSS-Engine enthält, die Ereignisse mit dem Zusatz „WARNING“ zurückgeben, wie zum Beispiel: LogLevel

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Ressourcen]: [Analysieren von Protokolldaten mit CloudWatch](#) Logs Insights

Sicherheitssäule von Amazon ElastiCache Well-Architected Lens

Die Säule der Sicherheit konzentriert sich auf den Schutz von Informationen und Systemen. Zu den wichtigsten Themen gehören die Vertraulichkeit und Integrität von Daten, die Ermittlung und Verwaltung von Berechtigungen mithilfe der Berechtigungsverwaltung, der Schutz von Systemen und die Einrichtung von Kontrollen zur Erkennung von Sicherheitsereignissen.

Themen

- [SEC 1: Welche Schritte ergreifen Sie, um den autorisierten Zugriff auf Daten zu ElastiCache kontrollieren?](#)

- [SEC 2: Benötigen Ihre Anwendungen zusätzliche Autorisierungen, die ElastiCache über netzwerkbasierte Kontrollen hinausgehen?](#)
- [SEC 3: Besteht das Risiko, dass Befehle versehentlich ausgeführt werden können und Datenverlust oder -ausfall verursachen?](#)
- [SEC 4: Wie stellen Sie die Datenverschlüsselung im Ruhezustand sicher ElastiCache](#)
- [SEC 5: Wie verschlüsselt man Daten bei der Übertragung mit? ElastiCache](#)
- [SEC 6: Wie schränken Sie den Zugriff auf Ressourcen der Steuerebene ein?](#)
- [SEC 7: Wie erkennen Sie Sicherheitsereignisse und wie reagieren Sie darauf?](#)

SEC 1: Welche Schritte ergreifen Sie, um den autorisierten Zugriff auf Daten zu ElastiCache kontrollieren?

Einführung auf Fragenebene: Alle ElastiCache Cluster sind so konzipiert, dass sie von Amazon Elastic Compute Cloud-Instances in einer VPC, serverlosen Funktionen (AWS Lambda) oder Containern (Amazon Elastic Container Service) aus aufgerufen werden können. Das am häufigsten anzutreffende Szenario ist der Zugriff auf einen ElastiCache Cluster von einer Amazon Elastic Compute Cloud-Instanz innerhalb derselben Amazon Virtual Private Cloud (Amazon Virtual Private Cloud). Bevor Sie von einer EC2 Amazon-Instance aus eine Verbindung zu einem Cluster herstellen können, müssen Sie die EC2 Amazon-Instance autorisieren, auf den Cluster zuzugreifen. Um auf einen ElastiCache Cluster zuzugreifen, der in einer VPC ausgeführt wird, muss dem Cluster Netzwerkzugang gewährt werden.

Vorteil auf Fragenebene: Der Netzwerkeingang für den Cluster wird über VPC-Sicherheitsgruppen gesteuert. Eine Sicherheitsgruppe fungiert als virtuelle Firewall für Ihre EC2 Amazon-Instances, um den eingehenden und ausgehenden Datenverkehr zu kontrollieren. Eingehende Regeln steuern den eingehenden Datenverkehr zu Ihrer Instance und ausgehende Regeln steuern den ausgehenden Datenverkehr von Ihrer Instance. Im Fall von ElastiCache, wenn ein Cluster gestartet wird, muss eine Sicherheitsgruppe zugeordnet werden. Dadurch wird sichergestellt, dass Regeln für ein- und ausgehenden Datenverkehr für alle Knoten gelten, aus denen der Cluster besteht. Darüber hinaus ElastiCache ist es so konfiguriert, dass es ausschließlich in privaten Subnetzen bereitgestellt wird, sodass auf sie nur über das private Netzwerk der VPC zugegriffen werden kann.

- [Erforderlich] Die Ihrem Cluster zugeordnete Sicherheitsgruppe steuert den Netzwerkeingang und den Zugriff auf den Cluster. Standardmäßig sind für eine Sicherheitsgruppe keine Regeln für eingehenden Datenverkehr definiert und daher auch kein Eingangspfad zu. ElastiCache Um dies zu aktivieren, konfigurieren Sie eine eingehende Regel für die Sicherheitsgruppe, die die

Quell-IP-Adresse/den Bereich, den TCP-Verkehr und den Port für Ihren ElastiCache Cluster angibt (Standardport 6379 ElastiCache für beispielsweise Valkey und Redis OSS). Es ist zwar möglich, eine sehr breite Palette von Eingangsquellen zuzulassen, z. B. alle Ressourcen innerhalb einer VPC (0.0.0.0/0), es wird jedoch empfohlen, die Regeln für eingehenden Datenverkehr so detailliert wie möglich zu definieren, z. B. nur den eingehenden Zugriff auf Valkey- oder Redis OSS-Clients zu autorisieren, die auf Amazon Amazon-Instances ausgeführt werden, die einer bestimmten Sicherheitsgruppe zugeordnet sind. EC2

[Ressourcen]:

- [Subnetze und Subnetzgruppen](#)
- [Zugriff auf Ihren Cluster oder die Replikationsgruppe](#)
- [Steuern des Datenverkehrs zu Ressourcen mithilfe von Sicherheitsgruppen](#)
- [Sicherheitsgruppen von Amazon Elastic Compute Cloud für Linux-Instances](#)
- [Erforderlich]AWS Identity and Access Management Richtlinien können Funktionen zugewiesen werden, die ihnen den Zugriff auf Daten ermöglichen. AWS Lambda ElastiCache Um diese Funktion zu aktivieren, erstellen Sie eine IAM-Ausführungsrolle mit der entsprechenden AWSLambdaVPCLambdaAccessExecutionRole Berechtigung und weisen Sie die Rolle dann der AWS Lambda Funktion zu.

[Ressourcen]: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon VPC: [Tutorial: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon VPC](#)

SEC 2: Benötigen Ihre Anwendungen zusätzliche Autorisierungen, die ElastiCache über netzwerkbasierte Kontrollen hinausgehen?

Einführung auf Fragenebene: In Szenarien, in denen es notwendig ist, den Zugriff auf Cluster auf individueller Client-Ebene einzuschränken oder zu kontrollieren, wird empfohlen, sich über den AUTH-Befehl zu authentifizieren. ElastiCache Authentifizierungstoken mit optionaler Benutzer- und Benutzergruppenverwaltung ermöglichen ElastiCache die Eingabe eines Kennworts, bevor Clients Befehle ausführen und auf Schlüssel zugreifen können, wodurch die Sicherheit der Datenebene verbessert wird.

Vorteil auf Fragenebene: Damit Ihre Daten sicher bleiben, ElastiCache bietet es Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten. Dazu gehört auch, dass Clients, bevor sie autorisierte

Befehle ausführen, mithilfe von RBAC (Role-Based Access Control) AUTH oder AUTH-Token (Passwort) eine Verbindung herstellen. ElastiCache

- [Am besten] Definieren Sie für ElastiCache Version 6.x und höher für Redis OSS und ElastiCache Version 7.2 und höher für Valkey Authentifizierungs- und Autorisierungskontrollen, indem Sie Benutzergruppen, Benutzer und Zugriffszeichenfolgen definieren. Weisen Sie Benutzer Benutzergruppen zu und weisen Sie Benutzergruppen dann Clustern zu. Damit RBAC verwendet wird, muss es bei der Clustererstellung ausgewählt und die Verschlüsselung während der Übertragung aktiviert sein. Stellen Sie sicher, dass Sie einen Valkey- oder Redis OSS-Client verwenden, der TLS unterstützt, um RBAC nutzen zu können.

[Ressourcen]:

- [Anwenden von RBAC auf eine Replikationsgruppe für ElastiCache](#)
- [Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge](#)
- [ACL](#)
- [Unterstützte Versionen ElastiCache](#)
- [Optimal] Für ElastiCache Versionen vor 6.x für Redis OSS empfiehlt es sich, neben der Festlegung starker token/password und strikter Kennwortrichtlinien für AUTH auch die Rotation des Passworts/Tokens zu verwenden. ElastiCache kann bis zu zwei (2) Authentifizierungstoken gleichzeitig verwalten. Sie können den Cluster auch so ändern, dass explizit die Verwendung von Authentifizierungstoken erforderlich ist.

[Ressourcen]: [Ändern des AUTH-Tokens auf einem vorhandenen Cluster ElastiCache](#)

SEC 3: Besteht das Risiko, dass Befehle versehentlich ausgeführt werden können und Datenverlust oder -ausfall verursachen?

Einführung auf Fragenebene: Es gibt eine Reihe von Valkey- oder Redis OSS-Befehlen, die sich nachteilig auf den Betrieb auswirken können, wenn sie versehentlich oder von böswilligen Akteuren ausgeführt werden. Diese Befehle können im Hinblick auf die Leistung und Datensicherheit unbeabsichtigte Folgen haben. Beispielsweise kann ein Entwickler in einer Entwicklungsumgebung routinemäßig den Befehl FLUSHALL aufrufen und aufgrund eines Fehlers versehentlich versuchen, diesen Befehl in einem Produktionssystem aufzurufen, was zu unbeabsichtigtem Datenverlust führt.

Vorteil auf Fragenebene: Ab ElastiCache Version 5.0.3 für Redis OSS haben Sie die Möglichkeit, bestimmte Befehle umzubenennen, die Ihre Arbeitslast stören könnten. Durch das Umbenennen der Befehle kann verhindert werden, dass diese versehentlich auf dem Cluster ausgeführt werden.

- [Erforderlich]

[Ressourcen]:

- [ElastiCache Version 5.0.3 für Redis OSS \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [ElastiCache Version 5.0.3 für Redis OSS-Parameteränderungen](#)
- [Redis OSS-Sicherheit](#)

SEC 4: Wie stellen Sie die Datenverschlüsselung im Ruhezustand sicher ElastiCache

Einführung auf Fragenebene: Es ElastiCache handelt sich zwar um einen In-Memory-Datenspeicher, aber es ist möglich, alle Daten zu verschlüsseln, die im Rahmen der Standardoperationen des Clusters dauerhaft (im Speicher) gespeichert werden. Dazu gehören sowohl geplante als auch manuelle Backups, die in Amazon S3 geschrieben wurden, sowie Daten, die aufgrund von Synchronisierungs- und Swap-Vorgängen auf dem Festplattenspeicher gespeichert wurden. Instance-Typen der M6g- und R6g-Familien verfügen außerdem über eine ständig aktive In-Memory-Verschlüsselung.

Vorteil auf Fragenebene: ElastiCache Bietet optionale Verschlüsselung im Ruhezustand, um die Datensicherheit zu erhöhen.

- [Erforderlich] Die Verschlüsselung im Ruhezustand kann auf einem ElastiCache Cluster (Replikationsgruppe) nur aktiviert werden, wenn dieser erstellt wurde. Ein vorhandener Cluster kann nicht geändert werden, um mit der Verschlüsselung von Daten im Ruhezustand zu beginnen. Stellt standardmäßig die Schlüssel bereit und verwaltet ElastiCache sie, die bei der Verschlüsselung im Ruhezustand verwendet werden.

[Ressourcen]:

- [Einschränkungen bei der Verschlüsselung im Ruhezustand](#)
- [Aktivieren der Verschlüsselung im Ruhezustand](#)
- [Am besten] Nutzen Sie EC2 Amazon-Instance-Typen, die Daten verschlüsseln, während sie sich im Arbeitsspeicher befinden (z. B. M6g oder R6g). Wenn möglich, sollten Sie erwägen, eigene Schlüssel für die Verschlüsselung im Ruhezustand zu verwalten. In Umgebungen mit strengeren Datensicherheitsumgebungen kann AWS Key Management Service (KMS) zur Selbstverwaltung der Customer Master Keys (CMK) verwendet werden. Durch die ElastiCache Integration mit sind Sie in der Lage AWS Key Management Service, die Schlüssel zu erstellen, zu besitzen und zu verwalten, die für die Verschlüsselung ruhender Daten in Ihrem ElastiCache Cluster verwendet werden.

[Ressourcen]:

- [Verwenden Sie vom Kunden verwaltete Schlüssel von AWS Key Management Service](#)
- [AWS Schlüsselverwaltungsdienst](#)
- [AWS -KMS-Konzepte](#)

SEC 5: Wie verschlüsselt man Daten bei der Übertragung mit? ElastiCache

Einführung auf Fragenebene: Es ist eine gängige Anforderung, zu verhindern, dass Daten während der Übertragung beschädigt werden. Dabei handelt es sich um Daten innerhalb von Komponenten eines verteilten Systems sowie zwischen Anwendungsclients und Clusterknoten. ElastiCache unterstützt diese Anforderung, indem es die Verschlüsselung von Daten ermöglicht, die zwischen Clients und Clustern sowie zwischen den Clusterknoten selbst übertragen werden. Instance-Typen der M6g- und R6g-Familien verfügen außerdem über eine ständig aktive In-Memory-Verschlüsselung.

Vorteil auf Frageebene: Die Verschlüsselung ElastiCache während der Übertragung durch Amazon ist eine optionale Funktion, mit der Sie die Sicherheit Ihrer Daten an den anfälligsten Stellen erhöhen können, wenn sie von einem Ort zum anderen übertragen werden.

- [Erforderlich] Die Verschlüsselung bei der Übertragung kann bei der Erstellung nur auf einem Cluster (Replikationsgruppe) aktiviert werden. Bitte beachten Sie, dass die Implementierung der Verschlüsselung bei der Übertragung aufgrund der zusätzlichen Verarbeitung, die für die Ver-/Entschlüsselung von Daten erforderlich ist, Auswirkungen auf die Leistung hat. Um die Auswirkungen zu verstehen, wird empfohlen, Ihren Workload vor und nach der Aktivierung encryption-in-transit zu vergleichen.

[Ressourcen]:

- [Übersicht über die Verschlüsselung bei der Übertragung](#)

SEC 6: Wie schränken Sie den Zugriff auf Ressourcen der Steuerebene ein?

Einführung auf Fragenebene: IAM-Richtlinien und ARN ermöglichen detaillierte Zugriffskontrollen ElastiCache für Valkey und Redis OSS und ermöglichen so eine strengere Kontrolle bei der Verwaltung der Erstellung, Änderung und Löschung von Clustern.

Vorteil auf Frageebene: Die Verwaltung von ElastiCache Amazon-Ressourcen wie Replikationsgruppen, Knoten usw. kann auf AWS Konten beschränkt werden, die über spezifische

Berechtigungen auf der Grundlage von IAM-Richtlinien verfügen, wodurch die Sicherheit und Zuverlässigkeit der Ressourcen verbessert wird.

- [Erforderlich] Verwalten Sie den Zugriff auf ElastiCache Amazon-Ressourcen, indem Sie AWS Benutzern bestimmte AWS Identity and Access Management Richtlinien zuweisen, sodass Sie genauer kontrollieren können, welche Konten welche Aktionen auf Clustern ausführen können.

[Ressourcen]:

- [Überblick über die Verwaltung der Zugriffsberechtigungen für Ihre Ressourcen ElastiCache](#)
- [Verwendung identitätsbasierter Richtlinien \(IAM-Richtlinien\) für Amazon ElastiCache](#)

SEC 7: Wie erkennen Sie Sicherheitsereignisse und wie reagieren Sie darauf?

Einführung auf Fragenebene: ElastiCache Exportiert bei der Implementierung mit aktiviertem RBAC CloudWatch Metriken, um Benutzer über Sicherheitsereignisse zu informieren. Diese Metriken helfen bei der Identifizierung von fehlgeschlagenen Authentifizierungsversuchen, Zugriffsschlüsseln oder der Ausführung von Befehlen, für die RBAC-Benutzer, die eine Verbindung herstellen, nicht autorisiert sind.

Darüber hinaus tragen AWS Produkte und Servicere Ressourcen dazu bei, Ihre gesamte Arbeitslast zu sichern, indem sie Bereitstellungen automatisieren und alle Aktionen und Änderungen für eine spätere Überprüfung/Prüfung protokollieren.

Vorteil auf Fragenebene: Durch die Überwachung von Ereignissen ermöglichen Sie Ihrem Unternehmen, gemäß Ihren Anforderungen, Richtlinien und Verfahren zu reagieren. Durch die Automatisierung der Überwachung und Reaktion auf diese Sicherheitsereignisse wird Ihre allgemeine Sicherheitslage gestärkt.

- [Erforderlich] Machen Sie sich mit den veröffentlichten CloudWatch Kennzahlen vertraut, die sich auf RBAC-Authentifizierungs- und Autorisierungsfehler beziehen.
 - AuthenticationFailures = Fehlgeschlagene Versuche, sich bei Valkey oder Redis OSS zu authentifizieren
 - KeyAuthorizationFailures = Fehlgeschlagene Versuche von Benutzern, ohne Erlaubnis auf Schlüssel zuzugreifen
 - CommandAuthorizationFailures = Fehlgeschlagene Versuche von Benutzern, Befehle ohne Erlaubnis auszuführen

[Ressourcen]:

- [Metriken für Valkey oder Redis OSS](#)
- [Am besten] Es wird empfohlen, Warnmeldungen und Benachrichtigungen für diese Metriken einzurichten und je nach Bedarf zu reagieren.

[Ressourcen]:

- [CloudWatch Amazon-Alarme verwenden](#)
- [Am besten] Verwenden Sie den Befehl Valkey oder Redis OSS ACL LOG, um weitere Informationen zu sammeln

[Ressourcen]:

- [ACL LOG](#)
- [Am besten] Machen Sie sich mit den Funktionen der AWS Produkte und Services im Zusammenhang mit der Überwachung, Protokollierung und Analyse von ElastiCache Bereitstellungen und Ereignissen vertraut

[Ressourcen]:

- [Protokollierung Amazon ElastiCache Amazon-API-Aufrufen mit AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-Backup-Check](#)
- [Überwachung der Nutzung mit Metrics CloudWatch](#)

Säule der Zuverlässigkeit von Amazon ElastiCache Well-Architected Lens

Der Schwerpunkt der Zuverlässigkeit liegt auf Workloads, die ihre vorgesehenen Funktionen erfüllen, und darauf, wie sie bei einem Ausfall schnell wiederhergestellt werden können, um die Anforderungen zu erfüllen. Zu den wichtigsten Themen gehören der Entwurf verteilter Systeme, die Wiederherstellungsplanung und die Anpassung an sich ändernde Anforderungen.

Themen

- [REL 1: Wie unterstützen Sie Bereitstellungen von Hochverfügbarkeitsarchitekturen \(HA\)?](#)
- [REL 2: Wie erreichen Sie Ihre Recovery Point Objectives \(RPOs\) mit? ElastiCache](#)
- [REL 3: Wie unterstützen Sie die Anforderungen an Disaster Recovery \(DR, Notfallwiederherstellung\)?](#)
- [REL 4: Wie planen Sie Failovers effektiv?](#)
- [REL 5: Sind Ihre ElastiCache Komponenten so konzipiert, dass sie skalierbar sind?](#)

REL 1: Wie unterstützen Sie Bereitstellungen von Hochverfügbarkeitsarchitekturen (HA)?

Einführung auf Fragenebene: Wenn Sie die Hochverfügbarkeitsarchitektur von Amazon verstehen, können ElastiCache Sie bei Verfügbarkeitsereignissen in einem stabilen Zustand arbeiten.

Vorteil auf Fragenebene: Wenn Sie Ihre ElastiCache Cluster so gestalten, dass sie ausfallsicher sind, wird eine höhere Verfügbarkeit Ihrer Bereitstellungen gewährleistet. ElastiCache

- [Erforderlich] Ermitteln Sie das Maß an Zuverlässigkeit, das Sie für Ihren Cluster benötigen. ElastiCache Verschiedene Workloads haben unterschiedliche Resilienzstandards, von rein kurzlebigen bis hin zu geschäftskritischen Workloads. Definieren Sie die Anforderungen für jede Art von Umgebung, die Sie betreiben, z. B. Entwicklung, Test und Produktion.

Caching-Engine: ElastiCache für Memcached im Vergleich zu Valkey und ElastiCache Redis OSS

1. ElastiCache für Memcached bietet keinen Replikationsmechanismus und wird hauptsächlich für kurzlebige Workloads verwendet.
 2. ElastiCache für Valkey und Redis bietet OSS HA-Funktionen, die im Folgenden beschrieben werden
- [Am besten] Verwenden Sie ElastiCache für Workloads, die HA erfordern, den Clustermodus mit mindestens zwei Replikaten pro Shard, auch für Workloads mit geringem Durchsatz, die nur einen Shard benötigen.
 1. Wenn der Clustermodus aktiviert ist, wird Multi-AZ automatisch aktiviert.

Multi-AZ minimiert Ausfallzeiten, indem automatische Failovers vom Primärknoten bis zu den Replikaten im Falle einer geplanten oder ungeplanten Wartung durchgeführt und AZ-Ausfälle minimiert werden.

2. Bei Shard-Workloads sorgen mindestens drei Shards für eine schnellere Wiederherstellung bei Failover-Ereignissen, da das Valkey- oder Redis OSS-Cluster-Protokoll erfordert, dass die Mehrheit der Primärknoten verfügbar ist, um ein Quorum zu erreichen.
3. Richten Sie zwei oder mehr Replikate für die Verfügbarkeit ein.

Zwei Replikate bieten eine verbesserte Leseskalierbarkeit sowie Leseverfügbarkeit in Szenarien, in denen ein Replikat gewartet wird.

4. Verwenden Sie Graviton2-basierte Knotentypen (Standardknoten in den meisten Regionen).

ElastiCache hat die Leistung auf diesen Knoten optimiert. Dadurch erhalten Sie eine bessere Replikations- und Synchronisierungsleistung, was zu einer insgesamt verbesserten Verfügbarkeit führt.

5. Überwachen Sie und passen Sie die Größe an, um zu erwartende Verkehrsspitzen zu bewältigen: Bei hoher Auslastung reagiert die Engine möglicherweise nicht mehr, was sich auf die Verfügbarkeit auswirkt. `BytesUsedForCache` und `DatabaseMemoryUsagePercentage` sind gute Indikatoren für Ihre Speicherauslastung, wohingegen `ReplicationLag` sie anhand Ihrer Schreibrate ein Indikator für den Zustand Ihrer Replikation sind. Sie können diese Metriken verwenden, um die Clusterskalierung auszulösen.
6. Stellen Sie die clientseitige Resilienz sicher, indem Sie [vor einem Produktions-Failover-Ereignis Tests mit der Failover-API](#) durchführen.

[Ressourcen]:

- [Konfigurieren Sie ElastiCache für Redis OSS für eine höhere Verfügbarkeit](#)
- [Hohe Verfügbarkeit mit Replikationsgruppen](#)

REL 2: Wie erreichen Sie Ihre Recovery Point Objectives (RPOs) mit? ElastiCache

Einführung auf Fragenebene: Machen Sie sich mit Workload-RPO vertraut, um fundierte Entscheidungen über ElastiCache Sicherungs- und Wiederherstellungsstrategien treffen zu können.

Vorteil auf Fragenebene: Eine integrierte RPO-Strategie kann die Geschäftskontinuität im Falle eines Notfallwiederherstellungs-Szenarios verbessern. Die Gestaltung Ihrer Sicherungs- und Wiederherstellungsrichtlinien kann Ihnen helfen, Ihre Recovery Point Objectives (RPO) für Ihre Daten zu erreichen. ElastiCache bietet Snapshot-Funktionen, die in Amazon S3 gespeichert werden, zusammen mit einer konfigurierbaren Aufbewahrungsrichtlinie. Diese Snapshots werden während eines definierten Backup-Fensters aufgenommen und vom Service automatisch verarbeitet. Wenn Ihr Workload zusätzliche Backup-Granularität erfordert, haben Sie die Möglichkeit, bis zu 20 manuelle Backups pro Tag zu erstellen. Manuell erstellte Backups unterliegen keiner Serviceaufbewahrungsrichtlinie und können auf unbestimmte Zeit aufbewahrt werden.

- [Erforderlich] Machen Sie sich mit dem RPO Ihrer ElastiCache Bereitstellungen vertraut und dokumentieren Sie es.
 - Beachten Sie, dass Memcached keine Backup-Prozesse anbietet.
 - Informieren Sie sich über die Funktionen der ElastiCache Backup- und Wiederherstellungsfunktionen.

- [Am besten] Etablieren Sie einen gut kommunizierten Prozess für die Sicherung Ihres Clusters.
 - Initiieren Sie bei Bedarf manuelle Backups.
 - Überprüfen Sie die Aufbewahrungsrichtlinien für automatische Backups.
 - Beachten Sie, dass manuelle Backups auf unbestimmte Zeit aufbewahrt werden.
 - Planen Sie Ihre automatischen Backups in Zeiten geringer Auslastung.
 - Führen Sie Backup-Operationen anhand von Lesereplikaten durch, um sicherzustellen, dass die Auswirkungen auf die Cluster-Leistung so gering wie möglich gehalten werden.
- [Gut] Nutzen Sie die Funktion für geplante Backups von ElastiCache , um Ihre Daten regelmäßig innerhalb eines bestimmten Zeitfensters zu sichern.
 - Testen Sie regelmäßig Wiederherstellungen aus Ihren Backups.
- [Ressourcen]:
 - [Redis OSS](#)
 - [Backup und Wiederherstellung für ElastiCache](#)
 - [Erstellen manueller Backups](#)
 - [Planen automatischer Backups](#)
 - [ElastiCache Cluster Backup und wiederherstellen](#)

REL 3: Wie unterstützen Sie die Anforderungen an Disaster Recovery (DR, Notfallwiederherstellung)?

Einführung auf Fragenebene: Disaster Recovery ist ein wichtiger Aspekt jeder Workload-Planung. ElastiCache bietet verschiedene Optionen für die Implementierung von Disaster Recovery auf der Grundlage der Anforderungen an die Belastbarkeit von Workloads. Mit Amazon ElastiCache Global Datastore können Sie in Ihren Cluster in einer Region schreiben und die Daten zum Lesen aus zwei anderen regionsübergreifenden Replikatclustern zur Verfügung stellen, wodurch Lesevorgänge mit niedriger Latenz und regionsübergreifende Notfallwiederherstellung ermöglicht werden.

Vorteil auf Fragenebene: Durch das Nachvollziehen und die Planung verschiedener Notfallszenarien kann die Geschäftskontinuität gewährleistet werden. DR-Strategien müssen gegenüber Kosten, Leistungseinbußen und Datenverlustpotenzial abgewogen werden.

- [Erforderlich] Entwickeln und dokumentieren Sie DR-Strategien für all Ihre ElastiCache Komponenten auf der Grundlage der Workload-Anforderungen. ElastiCache ist insofern einzigartig, als einige Anwendungsfälle völlig kurzlebig sind und keine DR-Strategie erfordern, während andere

am anderen Ende des Spektrums liegen und eine extrem robuste DR-Strategie erfordern. Alle Optionen müssen gegenüber Kostenoptimierung abgewogen werden – eine größere Resilienz erfordert eine umfassendere Infrastruktur.

Machen Sie sich mit den DR-Optionen vertraut, die auf regionaler und multiregionaler Ebene verfügbar sind.

- Multi-AZ-Bereitstellungen werden empfohlen, um sich vor AZ-Ausfällen zu schützen. Stellen Sie sicher, dass bei der Bereitstellung der Cluster-Modus in Multi-AZ-Architekturen aktiviert ist und mindestens 3 verfügbar sind. AZs
- Global Datastore wird als Schutz vor regionalen Ausfällen empfohlen.
- [Am besten] Aktivieren Sie Global Datastore für Workloads, die Resilienz auf regionaler Ebene erfordern.
 - Planen Sie ein Failover auf die sekundäre Region für den Fall einer Verschlechterung der primären Region ein.
 - Testen Sie den Failover-Prozess für mehrere Regionen, bevor Sie ein Failover in der Produktion durchführen.
 - Überwachen Sie die ReplicationLag-Metrik, um die möglichen Auswirkungen von Datenverlusten bei Failover-Ereignissen zu verstehen.
- [Ressourcen]:
 - [Minimieren von Ausfällen](#)
 - [Regionsübergreifende Replikation mithilfe globaler Datenspeicher AWS](#)
 - [Wiederherstellen aus einem Backup mit optionaler Cluster-Größenanpassung](#)
 - [Minimierung der Ausfallzeiten bei Valkey und ElastiCache Redis OSS mit Multi-AZ](#)

REL 4: Wie planen Sie Failovers effektiv?

Einführung auf Fragenebene: Die Aktivierung von Multi-AZ mit automatischen Failovers ist eine bewährte Methode. ElastiCache In bestimmten Fällen ersetzt OSS ElastiCache bei Valkey und Redis die Primärknoten im Rahmen des Servicebetriebs. Beispiele hierfür sind geplante Wartungsereignisse und der unwahrscheinliche Falle eines Ausfalls des Knotens oder eines Problems mit der Availability Zone. Erfolgreiche Failover hängen sowohl von Ihrer Konfiguration als auch von Ihrer ElastiCache Client-Bibliothekskonfiguration ab.

Vorteil auf Fragenebene: Wenn Sie sich an bewährte Methoden für ElastiCache Failover in Verbindung mit Ihrer spezifischen ElastiCache Client-Bibliothek halten, können Sie potenzielle Ausfallzeiten bei Failover-Ereignissen minimieren.

- [Erforderlich] Verwenden Sie bei deaktiviertem Clustermodus Timeouts, damit Ihre Clients anhand der aktualisierten IP-Adresse des primären Endpunkts erkennen, ob die Verbindung zum alten Primärknoten getrennt und erneut eine Verbindung zum neuen Primärknoten hergestellt werden muss. Im aktivierten Clustermodus ist die Clientbibliothek dafür zuständig, Änderungen in der zugrundeliegenden Cluster-Topologie zu erkennen. Dies wird in den meisten Fällen durch Konfigurationseinstellungen in der ElastiCache Client-Bibliothek erreicht, mit denen Sie auch die Häufigkeit und die Methode der Aktualisierung konfigurieren können. Jede Clientbibliothek bietet eigene Einstellungen. Weitere Informationen finden Sie in der entsprechenden Dokumentation.

[Ressourcen]:

- [Minimierung der Ausfallzeiten ElastiCache bei Valkey und Redis OSS mit Multi-AZ](#)
- Informieren Sie sich über die Best Practices Ihrer Kundenbibliothek. ElastiCache
- [Erforderlich] Erfolgreiche Failover hängen von einer fehlerfreien Replikationsumgebung zwischen dem Primärknoten und den Replikatknoten ab. Informieren Sie sich über die asynchrone Natur der Valkey- und Redis-OSS-Replikation sowie über die verfügbaren CloudWatch Kennzahlen zur Berichterstattung über die Replikationsverzögerung zwischen Primär- und Replikatknoten. In Anwendungsfällen, die eine höhere Datensicherheit erfordern, können Sie den Befehl WAIT nutzen, um Replikate zu zwingen, Schreibvorgänge zu bestätigen, bevor sie auf verbundene Clients antworten.

[Ressourcen]:

- [Metriken für Valkey oder Redis OSS](#)
- [Überwachung von Best Practices ElastiCache mithilfe von Amazon CloudWatch](#)
- [Am besten] Überprüfen Sie regelmäßig die Reaktionsfähigkeit Ihrer Anwendung während eines Failovers mithilfe der ElastiCache Test Failover API.

[Ressourcen]:

- [Testen des automatischen Failovers auf eine Read Replica am ElastiCache](#)
- [Testen des automatischen Failovers](#)

REL 5: Sind Ihre ElastiCache Komponenten so konzipiert, dass sie skalierbar sind?

Einführung auf Fragenebene: Wenn Sie die Skalierungsmöglichkeiten und die verfügbaren Bereitstellungstopologien verstehen, können sich Ihre ElastiCache Komponenten im Laufe der Zeit an sich ändernde Workload-Anforderungen anpassen. ElastiCache bietet 4-Wege-Skalierung: in/out (horizontal) und (vertikal). up/down

Vorteil auf Frageebene: Die Einhaltung von Best Practices für ElastiCache Bereitstellungen bietet ein Höchstmaß an Flexibilität bei der Skalierung und erfüllt gleichzeitig das Well-Architected-Prinzip der horizontalen Skalierung, um die Auswirkungen von Ausfällen zu minimieren.

- [Erforderlich] Machen Sie sich mit dem Unterschied zwischen Topologien im aktivierten und im deaktivierten Clustermodus vertraut. In fast allen Fällen wird empfohlen, die Bereitstellung mit aktiviertem Clustermodus vorzunehmen, da dies im Zeitverlauf eine größere Skalierbarkeit ermöglicht. Komponenten im deaktiviertem Clustermodus können nur begrenzt horizontal skaliert werden, indem Lesereplikate hinzugefügt werden.
- [Erforderlich] Machen Sie sich damit vertraut, wann und wie Sie skalieren müssen.
 - Für mehr READIOPS: Replikate hinzufügen
 - Für mehr WRITEOPS: Shards hinzufügen (aufskalieren)
 - Für mehr Netzwerk-I/O – netzwerkoptimierte Instances verwenden, hochskalieren
- [Am besten] Stellen Sie Ihre ElastiCache Komponenten mit aktiviertem Cluster-Modus bereit, wobei Sie sich eher auf mehr, kleinere Knoten als auf weniger, größere Knoten konzentrieren. Dadurch wird der Explosionsradius eines Knotenausfalls effektiv begrenzt.
- [Am besten] Nehmen Sie Replikate in Ihre Cluster auf, um die Reaktionsfähigkeit bei Skalierungsereignissen zu verbessern
- [Gut] Verwenden Sie bei deaktiviertem Cluster-Modus Read Replicas, um die gesamte Lesekapazität zu erhöhen. ElastiCache unterstützt bis zu 5 Read Replicas im deaktivierten Cluster-Modus sowie vertikale Skalierung.
- [Ressourcen]:
 - [Skalierung von Clustern ElastiCache](#)
 - [Online-Hochskalieren](#)

Säule der Leistung und Effizienz von Amazon ElastiCache Well-Architected Lens

Die Säule der Leistungseffizienz konzentriert sich auf die effiziente Nutzung von IT- und Computing-Ressourcen. Zu den wichtigsten Themen gehören die Auswahl der richtigen Ressourcentypen und -größen je nach Workload-Anforderungen, die Überwachung der Leistung und das Treffen fundierter Entscheidungen, um die Effizienz aufrechtzuerhalten, wenn sich die Geschäftsanforderungen weiterentwickeln.

Themen

- [PE 1: Wie überwachen Sie die Leistung Ihres ElastiCache Amazon-Clusters?](#)
- [PE 2: Wie verteilen Sie die Arbeit auf Ihre ElastiCache Clusterknoten?](#)
- [PE 3: Wie verfolgen und protokollieren Sie bei Caching-Workloads die Effektivität und Leistung Ihres Caches?](#)
- [PE 4: Wie optimiert Ihr Workload den Einsatz von Netzwerkressourcen und Verbindungen?](#)
- [PE 5: Wie gehen Sie mit dem Löschen von Schlüsseln um and/or ?](#)
- [PE 6: Wie modellieren und interagieren Sie mit Daten in ElastiCache?](#)
- [PE 7: Wie protokollieren Sie langsam laufende Befehle in Ihrem ElastiCache Amazon-Cluster?](#)
- [PE8: Wie hilft Auto Scaling dabei, die Leistung des ElastiCache Clusters zu steigern?](#)

PE 1: Wie überwachen Sie die Leistung Ihres ElastiCache Amazon-Clusters?

Einführung auf Fragenebene: Wenn Sie die vorhandenen Überwachungsmetriken verstehen, können Sie die aktuelle Auslastung ermitteln. Eine ordnungsgemäße Überwachung kann dazu beitragen, potenzielle Engpässe zu identifizieren, die sich auf die Leistung Ihres Clusters auswirken.

Vorteil auf Fragenebene: Wenn Sie die Metriken Ihres Clusters kennen, können Sie Optimierungsmethoden entwickeln, die zu einer geringeren Latenz und einem höheren Durchsatz führen können.

- [Erforderlich] Basisleistungstests mit einer Teilmenge Ihres Workloads.
 - Sie sollten die Leistung des tatsächlichen Workloads mithilfe von Mechanismen wie Lasttests überwachen.
 - Überwachen Sie die CloudWatch Metriken, während Sie diese Tests durchführen, um sich ein Bild von den verfügbaren Metriken zu machen und eine Leistungsbasis festzulegen.

- [Am besten] Benennen Sie rechenintensive Befehle ElastiCache für Valkey- und Redis-OSS-Workloads um, z. B. um die Fähigkeit von Benutzern einzuschränken KEYS, blockierende Befehle auf Produktionsclustern auszuführen.
- ElastiCache Workloads, auf denen Engine 6.x für Redis OSS ausgeführt wird, können die rollenbasierte Zugriffskontrolle nutzen, um bestimmte Befehle einzuschränken. Der Zugriff auf die Befehle kann gesteuert werden, indem Benutzer und Benutzergruppen mit der AWS Konsole oder CLI erstellt und die Benutzergruppen einem Cluster zugeordnet werden. Wenn RBAC in Redis OSS 6 aktiviert ist, können wir „- @dangerous“ verwenden. Dadurch werden teure Befehle wie KEYS, MONITOR, SORT usw. für diesen Benutzer nicht zugelassen.
- Benennen Sie Befehle für Engine-Version 5.x mithilfe des `rename-command` Parameters in der Cluster-Parametergruppe um.
- [Besser] Analysieren Sie langsame Abfragen und suchen Sie nach Optimierungsmöglichkeiten.
- ElastiCache Für Valkey- und Redis-OSS-Workloads erfahren Sie mehr über Ihre Abfragen, indem Sie das Slow Log analysieren. Sie können beispielsweise den folgenden Befehl verwenden, `valkey-cli slowlog get 10` um die letzten 10 Befehle anzuzeigen, die die Latenzschwellenwerte überschritten haben (standardmäßig 10 Millisekunden).
- Bestimmte Abfragen können effizienter ausgeführt werden, wenn komplexe OSS-Datenstrukturen ElastiCache für Valkey und Redis verwendet werden. Für Bereichsabfragen im numerischen Stil kann eine Anwendung beispielsweise einfache numerische Indizes mit sortierten Sätzen implementieren. Die Verwaltung dieser Indizes kann die Anzahl der am Datensatz durchgeführten Scans reduzieren und Daten mit höherer Leistungseffizienz zurückgeben.
- ElastiCache Für OSS-Workloads von Valkey und Redis `redis-benchmark` bietet es eine einfache Oberfläche zum Testen der Leistung verschiedener Befehle mithilfe benutzerdefinierter Eingaben wie Anzahl der Clients und Datengröße.
- Da Memcached nur einfache Befehle auf Schlüsselebene unterstützt, sollten Sie erwägen, zusätzliche Schlüssel als Indizes zu erstellen, um zu vermeiden, dass der Schlüsselbereich wiederholt wird, um Client-Abfragen zu bearbeiten.
- [Ressourcen]:
 - [Überwachung der Nutzung mit Metrics CloudWatch](#)
 - [CloudWatch Amazon-Alarme verwenden](#)
 - [OSS-spezifische Parameter für Valkey und Redis](#)
 - [SLOWLOG](#)
 - [Benchmark](#)

PE 2: Wie verteilen Sie die Arbeit auf Ihre ElastiCache Clusterknoten?

Einführung auf Fragenebene: Die Art und Weise, wie Ihre Anwendung eine Verbindung zu ElastiCache Amazon-Knoten herstellt, kann sich auf die Leistung und Skalierbarkeit des Clusters auswirken.

Vorteil auf Fragenebene: Durch die richtige Nutzung der verfügbaren Knoten im Cluster wird sichergestellt, dass die Arbeit auf die verfügbaren Ressourcen verteilt wird. Die folgenden Methoden tragen ebenfalls dazu bei, ungenutzte Ressourcen zu vermeiden.

- [Erforderlich] Sorgen Sie dafür, dass sich die Clients mit dem richtigen ElastiCache Endpunkt verbinden.
 - ElastiCache für Valkey und Redis implementiert OSS je nach verwendetem Clustermodus unterschiedliche Endpunkte. Wenn der Clustermodus aktiviert ist, ElastiCache wird ein Konfigurationsendpunkt bereitgestellt. ElastiCache stellt bei deaktiviertem Clustermodus einen primären Endpunkt bereit, der normalerweise für Schreibvorgänge verwendet wird, und einen Leser-Endpunkt für die Verteilung von Lesevorgängen zwischen Replikaten. Die korrekte Implementierung dieser Endpunkte führt zu einer besseren Leistung und einfacheren Skalierungsvorgängen. Vermeiden Sie es, eine Verbindung zu einzelnen Knotenendpunkten herzustellen, es sei denn, dies ist ausdrücklich erforderlich.
 - ElastiCache stellt für Memcached-Cluster mit mehreren Knoten einen Konfigurationsendpunkt bereit, der Auto Discovery aktiviert. Es wird empfohlen, einen Hashing-Algorithmus zu verwenden, um die Arbeit gleichmäßig auf die Cache-Knoten zu verteilen. Viele Memcached-Clientbibliotheken implementieren konsistentes Hashing. Überprüfen Sie die Dokumentation der von Ihnen verwendeten Bibliothek darauf, ob sie konsistentes Hashing unterstützt und wie es implementiert wird. Weitere Informationen zur Implementierung dieser Funktionen finden Sie [hier](#).
- [Besser] Nutzen Sie die Vorteile von Clustern, die im OSS-Clustermodus aktiviert sind, ElastiCache für Valkey und Redis, um die Skalierbarkeit zu verbessern.
 - ElastiCache für Valkey und Redis OSS (Clustermodus aktiviert) unterstützen Cluster [Online-Skalierungsoperationen](#) (out/in and up/down), um Daten dynamisch auf mehrere Shards zu verteilen. Durch die Verwendung des Konfigurationsendpunkts wird sichergestellt, dass sich Ihre clusterfähigen Clients an Änderungen in der Cluster-Topologie anpassen können.
 - Sie können den Cluster auch neu verteilen, indem Sie Hashslots zwischen verfügbaren Shards in Ihrem ElastiCache for Valkey- und Redis OSS-Cluster (Cluster-Modus aktiviert) verschieben. Dies trägt dazu bei, die Arbeit effizienter auf die verfügbaren Shards zu verteilen.

- [Besser] Implementieren Sie eine Strategie zur Identifizierung und Korrektur von Tastaturbefehlen in Ihrem Workload.
 - Bedenken Sie die Auswirkungen multidimensionaler Valkey- oder Redis-OSS-Datenstrukturen wie Listen, Streams, Sets usw. Diese Datenstrukturen werden in einzelnen Schlüsseln gespeichert, die sich auf einem einzigen Knoten befinden. Ein sehr großer multidimensionaler Schlüssel nimmt potenziell mehr Netzwerkkapazität und Speicher in Anspruch als andere Datentypen und kann eine unverhältnismäßige Nutzung dieses Knotens bewirken. Wenn möglich, sollten Sie Ihren Workload so gestalten, dass der Datenzugriff auf viele einzelne Schlüssel verteilt wird.
 - Tastaturbefehle im Workload können die Leistung des verwendeten Knotens beeinträchtigen. ElastiCache Bei OSS-Workloads von Valkey und Redis können Sie Hotkeys erkennen, `valkey-cli --hotkeys` wenn eine LFU-Richtlinie zur maximalen Speicherbelegung vorhanden ist.
 - Erwägen Sie, Tastaturbefehle auf mehreren Knoten zu replizieren, um den Zugriff auf sie gleichmäßiger zu verteilen. Bei diesem Ansatz muss der Client auf mehrere Primärknoten schreiben (der Valkey- oder Redis-OSS-Knoten selbst bietet diese Funktionalität nicht) und zusätzlich zum ursprünglichen Schlüsselnamen eine Liste mit Schlüsselnamen zum Lesen führen.
 - ElastiCache [Engine 7.2 für Valkey und höher sowie ElastiCache Version 6 für Redis OSS und höher unterstützen alle servergestütztes clientseitiges Caching](#). Auf diese Weise können Anwendungen auf Änderungen an einem Schlüssel warten, bevor sie über das Netzwerk zurückrufen. ElastiCache
- [Ressourcen]:
 - [Konfigurieren Sie ElastiCache für Valkey und Redis OSS für eine höhere Verfügbarkeit](#)
 - [Verbindungsendpunkte finden in ElastiCache](#)
 - [Bewährte Methoden für den Lastausgleich](#)
 - [Online-Resharding für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
 - [Clientseitiges Caching in Valkey und Redis OSS](#)

PE 3: Wie verfolgen und protokollieren Sie bei Caching-Workloads die Effektivität und Leistung Ihres Caches?

Einführung auf Fragenebene: Caching ist ein häufig auftretender Workload, ElastiCache und es ist wichtig, dass Sie wissen, wie Sie die Effektivität und Leistung Ihres Caches verwalten können.

Vorteil auf Fragenebene: Ihre Anwendung kann Anzeichen einer stagnierenden Leistung aufweisen. Ihre Fähigkeit, Cache-spezifische Metriken als Grundlage für Ihre Entscheidung zur Steigerung der App-Leistung zu verwenden, ist für Ihren Cache-Workload von entscheidender Bedeutung.

- [Erforderlich] Messen und verfolgen Sie die Cache-Trefferquote im Zeitverlauf. Die Effizienz Ihres Caches wird durch seine „Cache-Trefferquote“ bestimmt. Die Cache-Trefferquote wird durch die Summe der Schlüsseltreffer geteilt durch die Gesamtzahl der Treffer und Fehlversuche definiert. Je näher die Quote an 1 liegt, desto effektiver ist Ihr Cache. Eine niedrige Cache-Trefferquote wird durch die Menge der Cache-Fehlversuche verursacht. Cache-Fehlversuche treten auf, wenn der angeforderte Schlüssel nicht im Cache gefunden wird. Ein Schlüssel befindet sich nicht im Cache, weil er entweder bereinigt oder gelöscht wurde, abgelaufen ist oder nie existiert hat. Machen Sie sich klar, warum sich Schlüssel nicht im Cache befinden, und entwickeln Sie geeignete Strategien, um sie im Cache vorliegen zu haben.

[Ressourcen]:

- [Metriken für Valkey und Redis OSS](#)
- [Erforderlich] Messen und erfassen Sie die Leistung Ihres Anwendungs-Caches in Verbindung mit den Werten für Latenz und CPU-Auslastung, um zu ermitteln, ob Sie Anpassungen an Ihren time-to-live oder anderen Anwendungskomponenten vornehmen müssen. ElastiCache bietet eine Reihe von CloudWatch Metriken für aggregierte Latenzen für jede Datenstruktur. Diese Latenzmetriken werden anhand der Commandstats-Statistik des Befehls INFO berechnet und beinhalten nicht die Netzwerk- und I/O-Zeit. Dies ist nur die Zeit, die für die Verarbeitung der ElastiCache Operationen benötigt wird.

[Ressourcen]:

- [Metriken für Valkey und Redis OSS](#)
- [Überwachung von Best Practices ElastiCache mithilfe von Amazon CloudWatch](#)
- [Am besten] Wählen Sie die richtige Caching-Strategie für Ihre Anforderungen aus. Eine niedrige Cache-Trefferquote wird durch die Menge der Cache-Fehlversuche verursacht. Wenn Ihr Workload so konzipiert ist, dass es nur wenige Cache-Fehlversuche gibt (z. B. Kommunikation in Echtzeit), sollten Sie Ihre Caching-Strategien überprüfen und die für Ihren Workload am besten geeigneten Lösungen anwenden, z. B. Abfrage-Instrumente zur Messung von Speicher und Leistung. Die tatsächlichen Strategien, die Sie zum Auffüllen und Verwalten Ihres Cache implementieren müssen, hängt von den Daten, die Ihre Clients zwischenspeichern müssen, und den Zugriffsmustern auf diese Daten ab. Es ist beispielsweise unwahrscheinlich, dass Sie dieselbe

Strategie sowohl für personalisierte Empfehlungen in einer Streaming-Anwendung als auch für aktuelle Nachrichten verwenden.

[Ressourcen]:

- [Caching-Strategien für Memcached](#)
- [Bewährte Methoden für das Caching](#)
- [Skalierbare Leistung mit Amazon ElastiCache Whitepaper](#)

PE 4: Wie optimiert Ihr Workload den Einsatz von Netzwerkressourcen und Verbindungen?

Einführung auf Fragenebene: ElastiCache Für Valkey, Memcached und Redis wird OSS von vielen Anwendungsclients unterstützt, und die Implementierungen können variieren. Sie müssen das vorhandene Netzwerk- und Verbindungsmanagement verstehen, um mögliche Auswirkungen auf die Leistung zu analysieren.

Vorteil auf Fragenebene: Die effiziente Nutzung von Netzwerkressourcen kann die Leistungseffizienz Ihres Clusters verbessern. Die folgenden Empfehlungen können die Netzwerkanforderungen reduzieren und die Latenz und den Durchsatz des Clusters verbessern.

- [Erforderlich] Verwalten Sie proaktiv Verbindungen zu Ihrem Cluster. ElastiCache
 - Verbindungspooling in der Anwendung reduziert den Overhead im Cluster, der durch das Öffnen und Schließen von Verbindungen entsteht. Überwachen Sie das Verbindungsverhalten in Amazon CloudWatch mithilfe von `CurrConnections` und `NewConnections`.
 - Vermeiden Sie Verbindungslecks, indem Sie die Client-Verbindungen gegebenenfalls ordnungsgemäß schließen. Zu den Strategien zur Verbindungsverwaltung gehören das ordnungsgemäße Schließen von Verbindungen, die nicht verwendet werden, und das Festlegen von Verbindungs-Timeouts.
 - Für Memcached-Workloads gibt es eine konfigurierbare Speichermenge, die für die Verarbeitung von Verbindungen reserviert ist und als `memcached_connections_overhead` bezeichnet wird.
- [Besser] Komprimieren Sie große Objekte, um den Speicherplatz zu reduzieren und den Netzwerkdurchsatz zu verbessern.
 - Datenkomprimierung kann den erforderlichen Netzwerkdurchsatz (Gbit/s) reduzieren, erhöht jedoch den Arbeitsaufwand für die Anwendung zum Komprimieren und Dekomprimieren von Daten.

- Durch die Komprimierung wird auch der Speicherverbrauch von Schlüsseln reduziert.
- Berücksichtigen Sie auf der Grundlage Ihrer Anwendungsanforderungen die Kompromisse zwischen Komprimierungsverhältnis und Komprimierungsgeschwindigkeit.
- [Ressourcen]:
 - [ElastiCache - Globaler Datenspeicher](#)
 - [Memcached-spezifische Parameter](#)
 - [ElastiCache Version 5.0.3 für Redis OSS verbessert I/O die Handhabung, um die Leistung zu steigern](#)
 - [Metriken für Valkey und Redis OSS](#)
 - [ElastiCache Für höhere Verfügbarkeit konfigurieren](#)

PE 5: Wie gehen Sie mit dem Löschen von Schlüsseln um and/or ?

Einführung auf Fragenebene: Workloads haben unterschiedliche Anforderungen und erwartetes Verhalten, wenn sich ein Clusterknoten den Grenzwerten für den Speicherverbrauch nähert. ElastiCache hat unterschiedliche Richtlinien für den Umgang mit diesen Situationen.

Vorteil auf Fragenebene: Die richtige Verwaltung des verfügbaren Speichers und das Verständnis der Lösungsrichtlinien tragen dazu bei, dass das Clusterverhalten bei Überschreitung der Speicherlimits von Instances besser erkannt wird.

- [Erforderlich] Instrumentieren Sie den Datenzugriff, um zu beurteilen, welche Richtlinie angewendet werden soll. Identifizieren Sie eine geeignete Richtlinie für maximalen Arbeitsspeicher, um zu kontrollieren, ob und wie Bereinigungen im Cluster durchgeführt werden.
 - Eine Bereinigung erfolgt, wenn der maximale Arbeitsspeicher des Clusters verbraucht ist und eine Richtlinie vorhanden ist, welche die Bereinigung zulässt. Das Verhalten des Clusters in dieser Situation hängt von der angegebenen Bereinigungsrichtlinie ab. Diese Richtlinie kann mithilfe der Parametergruppe `maxmemory-policy` auf dem Cluster verwaltet werden.
 - Die Standardrichtlinie `volatile-lru` gibt Speicherplatz frei, indem Schlüssel mit einer festgelegten Ablaufzeit (TTL-Wert) bereinigt werden. Die Richtlinien „Least frequently used“ (LFU, am wenigsten häufig verwendet) und „Least recently used“ (LRU, am wenigsten zuletzt verwendet) entfernen Schlüssel je nach Nutzung.
 - Für Memcached-Workloads gibt es eine standardmäßige LRU-Richtlinie, welche die Bereinigungen auf jedem Knoten steuert. Die Anzahl der Räumungen in Ihrem ElastiCache Amazon-Cluster kann mithilfe der Räumungsmetrik auf Amazon überwacht werden. CloudWatch

- [Besser] Standardisieren Sie das Löschverhalten, um die Auswirkungen auf die Leistung Ihres Clusters zu kontrollieren und unerwartete Leistungsengpässe zu vermeiden.
 - ElastiCache Denn bei OSS-Workloads von Valkey und Redis UNLINK ist das explizite Entfernen von Schlüsseln aus dem Cluster wie folgt DEL: Die angegebenen Schlüssel werden entfernt. Der Befehl führt jedoch die eigentliche Speicherrückgewinnung in einem anderen Thread durch, sodass er nicht blockiert, während dies mit DEL der Fall ist. Die eigentliche Entfernung erfolgt später asynchron.
 - Bei ElastiCache Version 6.x für Redis OSS-Workloads kann das Verhalten des DEL Befehls in der Parametergruppe mithilfe von Parameter geändert werden. `lazyfree-lazy-user-del`
- [Ressourcen]:
 - [Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen](#)
 - [UNLINK](#)
 - [Cloud-Finanzmanagement mit AWS](#)

PE 6: Wie modellieren und interagieren Sie mit Daten in ElastiCache?

Einführung auf Fragenebene: ElastiCache ist stark von der Anwendung abhängig von den Datenstrukturen und dem verwendeten Datenmodell, muss aber auch der zugrunde liegende Datenspeicher (falls vorhanden) berücksichtigt werden. Machen Sie sich mit den verfügbaren Datenstrukturen vertraut und stellen Sie sicher, dass Sie die Datenstrukturen verwenden, die Ihren Anforderungen am besten entsprechen.

Vorteil auf Frageebene: Die Datenmodellierung ElastiCache besteht aus mehreren Ebenen, einschließlich Anwendungsfällen, Datentypen und Beziehungen zwischen Datenelementen. Darüber hinaus verfügen alle Datentypen und Befehle über eigene, gut dokumentierte Leistungssignaturen.

- [Am besten] Eine bewährte Methode besteht darin, das unbeabsichtigte Überschreiben von Daten zu reduzieren. Verwenden Sie eine Namenskonvention, die überlappende Schlüsselnamen minimiert. Bei der herkömmlichen Benennung Ihrer Datenstrukturen wird eine hierarchische Methode wie `APPNAME:CONTEXT:ID` bzw. `ORDER-APP:CUSTOMER:123` verwendet.

[Ressourcen]:

- [Benennung von Schlüsseln](#)
- [Best] ElastiCache for Valkey und Redis OSS-Befehle haben eine Zeitkomplexität, die durch die Big-O-Notation definiert wird. Diese Zeitkomplexität eines Befehls ist eine algorithmic/mathematical Darstellung seiner Auswirkungen. Bei der Einführung eines neuen Datentyps in Ihre Anwendung

müssen Sie die Zeitkomplexität der zugehörigen Befehle sorgfältig überprüfen. Befehle mit einer Zeitkomplexität von $O(1)$ sind zeitlich konstant und hängen nicht von der Größe der Eingabe ab. Befehle mit einer Zeitkomplexität von $O(N)$ sind jedoch zeitlich linear und hängen von der Größe der Eingabe ab. Aufgrund des Single-Thread-Designs von ElastiCache Valkey und Redis OSS führt ein großes Volumen an Vorgängen mit hoher Zeitkomplexität zu einer geringeren Leistung und möglichen Zeitüberschreitungen.

[Ressourcen]:

- [Befehle](#)
- [Am besten] Verwenden Sie diese Option APIs , um sich über die grafische Benutzeroberfläche einen Überblick über das Datenmodell in Ihrem Cluster zu verschaffen.

[Ressourcen]:

- [Redis OSS Commander](#)
- [Redis OSS-Browser](#)
- [Redsmin](#)

PE 7: Wie protokollieren Sie langsam laufende Befehle in Ihrem ElastiCache Amazon-Cluster?

Einführung auf Fragenebene: Die Leistungsoptimierung bietet Vorteile durch die Erfassung, Aggregation und Benachrichtigung von Befehlen mit langer Laufzeit. Wenn Sie wissen, wie lange die Ausführung von Befehlen dauert, können Sie feststellen, welche Befehle zu einer schlechten Leistung führen und welche Befehle die Engine daran hindern, optimal zu funktionieren. ElastiCache kann diese Informationen auch an Amazon CloudWatch oder Amazon Kinesis Data Firehose weiterleiten.

Vorteil auf Fragenebene: Die Protokollierung an einem festen Standort und die Bereitstellung von Benachrichtigungsereignissen bei langsamen Befehlen können zu einer detaillierten Leistungsanalyse beitragen und zur Auslösung automatisierter Ereignisse verwendet werden.

- [Erforderlich] ElastiCache Ausführung einer Valkey-Engine 7.2 oder neuer oder Ausführung einer Redis OSS-Engine Version 6.0 oder neuer, ordnungsgemäß konfigurierte Parametergruppe und aktiviertes SLOWLOG-Logging auf dem Cluster.
- Die erforderlichen Parameter sind nur verfügbar, wenn die Engine-Versionskompatibilität auf Valkey 7.2 und höher oder Redis OSS Version 6.0 oder höher eingestellt ist.

- Die SLOWLOG-Protokollierung erfolgt, wenn die Serverausführungszeit eines Befehls länger als ein bestimmter Wert dauert. Das Verhalten des Clusters hängt von den zugehörigen Parametergruppenparametern ab, d. h. `slowlog-log-slower-than` und `slowlog-max-len`.
- Änderungen werden sofort wirksam.
- [Am besten] Nutzen Sie unsere Kinesis Data Firehose Firehose-Funktionen. CloudWatch
 - Nutzen Sie die Filter- und Alarmfunktionen von CloudWatch CloudWatch Logs Insights und Amazon Simple Notification Services, um die Leistung zu überwachen und Ereignisse zu benachrichtigen.
 - Verwenden Sie die Streaming-Funktionen von Kinesis Data Firehose, um SLOWLOG-Protokolle im permanenten Speicher zu archivieren oder um eine automatische Cluster-Parameteroptimierung auszulösen.
 - Stellen Sie fest, ob das JSON- oder Plain-TEXT-Format Ihren Anforderungen am besten entspricht.
 - Stellen Sie IAM-Berechtigungen für die Veröffentlichung in CloudWatch oder Kinesis Data Firehose bereit.
- [Besser] Konfigurieren Sie `slowlog-log-slower-than` auf einen anderen Wert als den Standardwert.
 - Dieser Parameter bestimmt, wie lange ein Befehl innerhalb der Valkey- oder Redis-OSS-Engine ausgeführt werden kann, bevor er als langsam laufender Befehl protokolliert wird. Der Standardwert ist 10 000 Mikrosekunden (10 Millisekunden). Der Standardwert ist für einige Workloads möglicherweise zu hoch.
 - Ermitteln Sie auf der Grundlage der Anwendungsanforderungen und der Testergebnisse einen Wert, der für Ihren Workload besser geeignet ist. Ein zu niedriger Wert kann jedoch zu übermäßigen Daten führen.
- [Besser] Übernehmen Sie für `slowlog-max-len` den Standardwert.
 - Dieser Parameter bestimmt die Obergrenze dafür, wie viele langsam laufende Befehle zu einem bestimmten Zeitpunkt im Valkey- oder Redis-OSS-Speicher erfasst werden. Mit dem Wert 0 wird die Erfassung effektiv deaktiviert. Je höher der Wert, desto mehr Einträge werden im Speicher gespeichert. Dadurch wird die Wahrscheinlichkeit verringert, dass wichtige Informationen bereinigt werden, bevor sie überprüft werden können. Der Standardwert lautet 128.
 - Der Standardwert ist für die meisten Workloads geeignet. Wenn Daten in einem erweiterten Zeitfenster von der Valkey-CLI aus über den SLOWLOG-Befehl analysiert werden müssen, sollten Sie erwägen, diesen Wert zu erhöhen. Dadurch können mehr Befehle im OSS-Speicher von Valkey oder Redis verbleiben.

Wenn Sie die SLOWLOG-Daten entweder an CloudWatch Logs oder Kinesis Data Firehose ausgeben, werden die Daten dauerhaft gespeichert und können außerhalb des ElastiCache Systems analysiert werden, wodurch die Notwendigkeit reduziert wird, eine große Anzahl langsam laufender Befehle im Valkey- oder Redis-OSS-Speicher zu speichern.

- [Ressourcen]:
 - [Wie aktiviere ich Slow Log in einem Cluster?](#)
 - [Protokollzustellung](#)
 - [Redis OSS-spezifische Parameter](#)
 - <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
 - [Amazon Kinesis Data Firehose](#)

PE8: Wie hilft Auto Scaling dabei, die Leistung des ElastiCache Clusters zu steigern?

Einführung auf Fragenebene: Durch die Implementierung der Funktion von Valkey oder Redis OSS Auto Scaling können sich Ihre ElastiCache Komponenten im Laufe der Zeit anpassen, um die gewünschten Shards oder Replicas automatisch zu erhöhen oder zu verringern. Dies kann erreicht werden, indem entweder die Zielverfolgung oder die geplante Skalierungsrichtlinie implementiert wird.

Vorteil auf Fragenebene: Wenn Sie die Arbeitslastspitzen verstehen und entsprechend planen, können Sie eine verbesserte Caching-Leistung und einen unterbrechungsfreien Betrieb sicherstellen. ElastiCache Auto Scaling überwacht kontinuierlich Ihre CPU-/Speicherauslastung, um sicherzustellen, dass Ihr Cluster mit den gewünschten Leistungsniveaus arbeitet.

- [Erforderlich] Beim Starten eines Clusters ElastiCache für Valkey oder Redis OSS:
 1. Stellen Sie sicher, dass der Clustermodus aktiviert ist.
 2. Stellen Sie sicher, dass die Instance zu einer Familie mit einem bestimmten Typ und einer bestimmten Größe gehört, die Auto Scaling unterstützt.
 3. Vergewissern Sie sich, dass der Cluster nicht in globalen Datenspeichern, Outposts oder lokalen Zonen ausgeführt wird

[Ressourcen]:

- [Skalierung von Clustern in Valkey und Redis OSS \(Clustermodus aktiviert\)](#)
- [Verwenden von Auto Scaling mit Shards](#)
- [Verwenden von Auto Scaling mit Replikaten](#)

- [Am besten] Stellen Sie fest, ob Ihr Workload lese- oder schreibintensiv ist, um die Skalierungsrichtlinie zu definieren. Verwenden Sie nur eine Nachverfolgungsmetrik, um eine optimale Leistung zu erzielen. Es wird empfohlen, für jede Dimension mehrere Richtlinien zu vermeiden, da Auto-Scaling-Richtlinien aufskalieren, wenn das Ziel erreicht wird, aber erst dann abskalieren, wenn alle Ziel-Nachverfolgungsrichtlinien für die Abskalierung bereit sind.

[Ressourcen]:

- [Auto-Scaling-Richtlinien](#)
- [Definieren einer Skalierungsrichtlinie](#)
- [Am besten] Wenn Sie die Leistung im Zeitverlauf überwachen, können Sie Workload-Änderungen erkennen, die bei punktueller Überwachung unbemerkt bleiben würden. Sie können die entsprechenden CloudWatch Metriken für die Clusterauslastung über einen Zeitraum von vier Wochen analysieren, um den Zielschwellenwert zu bestimmen. Wenn Sie sich immer noch nicht sicher sind, welchen Wert Sie wählen möchten, empfehlen wir, mit dem minimal unterstützten vordefinierten Metrikwert zu beginnen.

[Ressourcen]:

- [Überwachung der Nutzung mit Metriken CloudWatch](#)
- [Besser] Wir empfehlen, Ihre Anwendung mit den erwarteten minimalen und maximalen Workloads zu testen, um die genaue Anzahl der Workloads zu ermitteln, die der Cluster shards/replicas benötigt, um Skalierungsrichtlinien zu entwickeln und Verfügbarkeitsprobleme zu minimieren.

[Ressourcen]:

- [Registrieren eines skalierbaren Ziels](#)
- [Registrierung eines skalierbaren Ziels mit dem AWS CLI](#)

Säule zur ElastiCache Kostenoptimierung bei Amazon Well-Architected Lens

Die Säule der Kostenoptimierung konzentriert sich auf die Vermeidung unnötiger Kosten. Zu den wichtigsten Themen gehören das Verständnis und die Kontrolle, wofür Geld ausgegeben wird, die Auswahl des am besten geeigneten Knotentyps (verwenden Sie Instances, die das Daten-Tiering auf der Grundlage der Workload-Anforderungen unterstützen), die richtige Anzahl von Ressourcentypen (wie viele Lesereplikate), die Analyse der Ausgaben im Zeitverlauf und die Skalierung, um Geschäftsanforderungen zu erfüllen, ohne zu viel Geld auszugeben.

Themen

- [COST 1: Wie identifizieren und verfolgen Sie die mit Ihren ElastiCache Ressourcen verbundenen Kosten? Wie entwickeln Sie Mechanismen, die es Benutzern ermöglichen, Ressourcen zu erstellen, zu verwalten und zu entsorgen?](#)
- [COST 2: Wie nutzen Sie Tools zur kontinuierlichen Überwachung, um die mit Ihren ElastiCache Ressourcen verbundenen Kosten zu optimieren?](#)
- [COST 3: Sollten Sie einen Instance-Typ verwenden, der Daten-Tiering unterstützt? Was sind die Vorteile von Daten-Tiering? Wann sollten Daten-Tiering-Instances nicht verwendet werden?](#)

COST 1: Wie identifizieren und verfolgen Sie die mit Ihren ElastiCache Ressourcen verbundenen Kosten? Wie entwickeln Sie Mechanismen, die es Benutzern ermöglichen, Ressourcen zu erstellen, zu verwalten und zu entsorgen?

Einführung auf Fragenebene: Um Kostenmetriken zu verstehen, müssen mehrere Teams beteiligt werden und zusammenarbeiten: Softwareentwicklung, Datenmanagement, Produktbesitzer, Finanzen und Führung. Um die wichtigsten Kostentreiber zu identifizieren, müssen alle Beteiligten die Hebel der Kontrolle der Servicenutzung und die Kompromisse beim Kostenmanagement verstehen. Dies ist häufig der entscheidende Unterschied zwischen erfolgreichen und weniger erfolgreichen Bemühungen zur Kostenoptimierung. Wenn Sie sicherstellen, dass Sie über Prozesse und Tools verfügen, mit denen Sie die Ressourcen verfolgen können, die von der Entwicklung über die Produktion bis hin zur Außerbetriebnahme geschaffen wurden, können Sie die damit verbundenen Kosten besser kontrollieren ElastiCache.

Vorteil auf Fragenebene: Die kontinuierliche Erfassung aller mit Ihrem Workload verbundenen Kosten erfordert ein tiefes Verständnis der Architektur, zu der ElastiCache auch eine ihrer Komponenten gehört. Darüber hinaus sollten Sie über einen Kostenmanagementplan verfügen, um die Nutzung zu erfassen und mit Ihrem Budget zu vergleichen.

- [Erforderlich] Richten Sie ein Cloud Center of Excellence (CCoE) mit einer der Gründungschartas ein, das die Definition, Nachverfolgung und Umsetzung von Kennzahlen rund um die Nutzung in Ihrem Unternehmen selbst übernimmt. ElastiCache Wenn ein CCoE existiert und funktioniert, stellen Sie sicher, dass es weiß, wie die damit verbundenen Kosten zu lesen und nachzuverfolgen sind. ElastiCache Wenn Ressourcen erstellt werden, überprüfen Sie anhand von IAM-Rollen und -Richtlinien, dass nur bestimmte Teams und Gruppen Ressourcen instanziiieren können. Damit wird sichergestellt, dass die Kosten mit den Geschäftsergebnissen in Verbindung stehen und dass im Hinblick auf die Kosten eine klare Linienverantwortung festgelegt wird.

1. CCoE sollte Kostenkennzahlen identifizieren, definieren und veröffentlichen, die regelmäßig — monatlich — aktualisiert werden und sich auf die wichtigsten ElastiCache Nutzungsdaten beziehen, wie z. B.:
 - a. Verwendete Knotentypen und ihre Attribute: Standard oder speicheroptimiert, On-Demand-Instances oder Reserved Instances, Regionen und Availability Zones
 - b. Arten von Umgebungen: kostenlose Umgebungen, Entwicklungsumgebungen, Testumgebungen und Produktionsumgebungen
 - c. Strategien zur Speicherung und Aufbewahrung von Backups
 - d. Datentransfer innerhalb und zwischen Regionen
 - e. Instances, die in Amazon Outposts ausgeführt werden
2. CCoE besteht aus einem funktionsübergreifenden Team, in dem nicht ausschließlich Softwareentwicklung, Datenmanagement, Produktteam, Finanzen und Führungsteams in Ihrem Unternehmen vertreten sind.

[Ressourcen]:

- [Schaffen eines Cloud-Kompetenzzentrums](#)
- [ElastiCacheAmazon-Preisgestaltung](#)
- [Erforderlich] Verwenden Sie Kostenzuordnungs-Tags, um die Kosten auf einer niedrigen Granularitätsstufe zu verfolgen. Verwenden Sie AWS Cost Management, um Ihre AWS Kosten und Nutzung im Laufe der Zeit zu visualisieren, zu verstehen und zu verwalten.
 1. Verwenden Sie Tags, um Ihre Ressourcen zu organisieren, und Tags zur Kostenzuweisung, um Ihre AWS Kosten detailliert zu verfolgen. Nachdem Sie die Kostenzuordnungs-Tags aktiviert haben, AWS verwendet es die Kostenzuordnungs-Tags, um Ihre Ressourcenkosten in Ihrem Kostenzuordnungsbericht zu organisieren, sodass Sie Ihre AWS Kosten leichter kategorisieren und verfolgen können. AWS stellt zwei Arten von Kostenverrechnungs-Tags bereit: AWS generierte Tags und benutzerdefinierte Tags. AWS definiert, erstellt und wendet die AWS generierten Tags für Sie an, und Sie definieren, erstellen und wenden benutzerdefinierte Tags an. Sie müssen beide Arten von Tags separat aktivieren, bevor sie in AWS-Kostenmanagement oder einem Kostenzuordnungsbericht angezeigt werden können.
 2. Verwenden Sie Tags für die Kostenzuweisung, um Ihre AWS Rechnung so zu organisieren, dass sie Ihrer eigenen Kostenstruktur entspricht. Wenn Sie Ihren Ressourcen in Amazon Kostenzuordnungs-Tags hinzufügen ElastiCache, können Sie die Kosten verfolgen, indem Sie die Ausgaben auf Ihren Rechnungen nach Ressourcen-Tag-Werten gruppieren. Erwägen Sie, ob Sie Tags miteinander kombinieren, um Kosten detaillierter zu verfolgen.

[Ressourcen]:

- [Verwenden von AWS Kostenzuweisungs-Tags](#)
 - [Überwachen von Kosten mit Kostenzuordnungs-Tags](#)
 - [AWS Cost Explorer](#)
- [Am besten] Verknüpfen Sie ElastiCache Kosten mit Kennzahlen, die sich auf das gesamte Unternehmen auswirken.
 1. Berücksichtigen Sie sowohl Geschäftsmetriken als auch betriebliche Metriken wie die Latenz – welche Konzepte in Ihrem Geschäftsmodell sind rollenübergreifend verständlich? Die Metriken müssen für so viele Rollen wie möglich in der Organisation verständlich sein.
 2. Beispiele: Benutzer, die gleichzeitig bedient werden, maximale und durchschnittliche Latenz pro Vorgang und Benutzer, Werte der Benutzerinteraktion, Anzahl der rates/week, session length/ user Benutzerrückgänge, Abbruchrate, Cache-Trefferquote und verfolgte Schlüssel

[Ressourcen]:

- [Überwachung der Nutzung mit Metriken CloudWatch](#)
- [Gut] Sorgen Sie für einen Überblick über die up-to-date Architektur und den Betrieb der Kennzahlen und Kosten für den gesamten verwendeten Workload ElastiCache.
 1. Verstehen Sie Ihr gesamtes Lösungsökosystem, das in der ElastiCache Regel Teil eines vollständigen Ökosystems von AWS Diensten in seinem Technologieangebot ist, von Kunden über API Gateway, Redshift bis hin QuickSight zu Reporting-Tools (zum Beispiel).
 2. Tragen Sie die Komponenten Ihrer Lösung wie Clients, Verbindungen, Sicherheit, In-Memory-Betrieb, Speicher, Ressourcenautomatisierung, Datenzugriff und -verwaltung in Ihrem Architekturdiagramm auf. Jede Ebene ist mit der gesamten Lösung verbunden und hat ihre eigenen Bedürfnisse und Funktionen, die and/or Ihnen helfen, die Gesamtkosten im Griff zu behalten.
 3. Ihr Diagramm sollte die Nutzung von Rechenleistung, Netzwerk, Speicher, Lebenszyklusrichtlinien, die Erfassung von Kennzahlen sowie die betrieblichen und funktionalen ElastiCache Elemente Ihrer Anwendung beinhalten
 4. Die Anforderungen Ihres Workloads werden sich wahrscheinlich im Laufe der Zeit ändern. Es ist wichtig, dass Sie Ihr Wissen über die zugrundeliegenden Komponenten sowie Ihre primären funktionalen Ziele aufrechterhalten und dokumentieren, um Ihr Workload-Kostenmanagement proaktiv zu gestalten.

5. Die Unterstützung der Geschäftsleitung in Bezug auf Transparenz, Rechenschaftspflicht, Priorisierung und Ressourcen ist entscheidend für eine effektive Kostenmanagementstrategie für Ihr Unternehmen. ElastiCache

COST 2: Wie nutzen Sie Tools zur kontinuierlichen Überwachung, um die mit Ihren ElastiCache Ressourcen verbundenen Kosten zu optimieren?

Einführung auf Fragenebene: Sie müssen ein ausgewogenes Verhältnis zwischen Ihren ElastiCache Kosten- und Anwendungsleistungskennzahlen anstreben. Amazon CloudWatch bietet Einblicke in wichtige Betriebskennzahlen, anhand derer Sie beurteilen können, ob Ihre ElastiCache Ressourcen im Verhältnis zu Ihren Anforderungen über- oder unterausgelastet sind. Im Hinblick auf die Kostenoptimierung müssen Sie verstehen, wann Ihre Ressourcen überlastet sind, und in der Lage sein, geeignete Mechanismen zu entwickeln, um die Größe Ihrer ElastiCache Ressourcen zu ändern und gleichzeitig Ihre Betriebs-, Verfügbarkeits-, Belastbarkeits- und Leistungsanforderungen aufrechtzuerhalten.

Vorteil auf Fragenebene: Im Idealfall haben Sie genügend Ressourcen bereitgestellt, um die betrieblichen Anforderungen Ihres Workloads zu erfüllen, und verfügen nicht über unterausgelastete Ressourcen, was zu einer suboptimalen Kostensituation führen kann. Sie müssen in der Lage sein, überdimensionierte ElastiCache Ressourcen sowohl zu identifizieren als auch zu vermeiden, sie über einen längeren Zeitraum zu betreiben.

- [Erforderlich] Wird verwendet CloudWatch , um Ihre ElastiCache Cluster zu überwachen und zu analysieren, wie sich diese Metriken auf Ihre AWS Cost Explorer Explorer-Dashboards beziehen.
 1. ElastiCache bietet sowohl Metriken auf Host-Ebene (z. B. CPU-Auslastung) als auch Metriken, die für die Cache-Engine-Software spezifisch sind (z. B. Cache-Abrufe und Cache-Fehlschläge). Diese Metriken werden für jeden Cache-Knoten in 60-Sekunden-Intervallen erfasst und veröffentlicht.
 2. ElastiCache Leistungsmetriken (CPUUtilization, EngineUtilization SwapUsage CurrConnections, und Räumungen) können darauf hinweisen, dass Sie skalieren müssen up/down (verwenden Sie larger/smaller Cache-Knotentypen) oder Shards). in/out (add more/less Machen Sie sich mit den Auswirkungen von Skalierungsentscheidungen auf die Kosten vertraut, indem Sie eine Playbook-Matrix erstellen, in der die zusätzlichen Kosten sowie die Mindest- und Höchstdauer geschätzt werden, die erforderlich sind, um die Schwellenwerte für die Anwendungsleistung zu erreichen.

[Ressourcen]:

- [Überwachung der Nutzung mithilfe von Metriken CloudWatch](#)
- [Welche Metriken sollte ich überwachen?](#)
- [ElastiCacheAmazon-Preisgestaltung](#)
- [Erforderlich] Verstehen und dokumentieren Sie Ihre Backup-Strategie und die Auswirkungen auf die Kosten.
 1. Mit ElastiCache, die Backups werden in Amazon S3 gespeichert, was eine dauerhafte Speicherung bietet. Sie müssen die Kostenauswirkungen verstehen, die sich aus Ihrer Fähigkeit ergeben, sich nach Ausfällen zu erholen.
 2. Aktivieren Sie automatische Backups, mit denen Backup-Dateien gelöscht werden, die das Aufbewahrungslimit überschritten haben.

[Ressourcen]:

- [Planen automatischer Backups](#)
- [Amazon Simple Storage Service – Preise](#)
- [Am besten] Verwenden Sie reservierte Knoten für Ihre Instances als bewusste Strategie zur Kostenkontrolle für Workloads, die wohlverstanden und dokumentiert sind. Für reservierte Knoten wird eine Vorausgebühr erhoben, die von der Art des Knotens und der Dauer der Reservierung – ein oder drei Jahre – abhängt. Diese Gebühr ist wesentlich geringer als die stündliche Nutzungsgebühr, die bei On-Demand-Knoten anfällt.
 1. Möglicherweise müssen Sie Ihre ElastiCache Cluster mit On-Demand-Knoten betreiben, bis Sie genügend Daten gesammelt haben, um die Reserved Instance-Anforderungen abzuschätzen. Planen und dokumentieren Sie die Ressourcen, die Sie benötigen, um Ihre Anforderungen zu erfüllen, und vergleichen Sie die erwarteten Kosten für die verschiedenen Instance-Typen (On-Demand-Instances und Reserved Instances).
 2. Evaluieren Sie regelmäßig die neuen verfügbaren Cache-Knotentypen und beurteilen Sie, ob es aus Sicht der Kosten- und Betriebsmetriken sinnvoll ist, Ihre Instance-Flotte auf neue Cache-Knotentypen zu migrieren.

COST 3: Sollten Sie einen Instance-Typ verwenden, der Daten-Tiering unterstützt?
Was sind die Vorteile von Daten-Tiering? Wann sollten Daten-Tiering-Instances nicht verwendet werden?

Einführung auf Fragenebene: Die Auswahl des geeigneten Instance-Typs wirkt sich nicht nur auf die Leistung und das Service-Level aus, sondern hat auch finanzielle Auswirkungen. Mit Instance-Typen

sind unterschiedliche Kosten verbunden. Die Auswahl eines oder mehrerer großer Instance-Typen, die alle Speicheranforderungen im Arbeitsspeicher erfüllen, könnte sich anbieten. Dies könnte jedoch erhebliche Kostenauswirkungen haben, sobald das Projekt ausgereift ist. Um sicherzustellen, dass der richtige Instance-Typ ausgewählt wird, muss die Leerlaufzeit der ElastiCache Objekte regelmäßig überprüft werden.

Vorteil auf Fragenebene: Sie sollten genau wissen, wie sich verschiedene Instance-Typen derzeit und in Zukunft auf Ihre Kosten auswirken. Geringe oder regelmäßige Workload-Änderungen sollten nicht zu unverhältnismäßigen Kostenänderungen führen. Wenn der Workload es zulässt, bieten Instance-Typen, die Daten-Tiering unterstützen, einen besseren Preis pro verfügbarem Speicher. Aufgrund des pro Instance verfügbaren SSD-Speichers unterstützen Daten-Tiering-Instances eine viel höhere Gesamtkapazität für Daten pro Instance.

- [Erforderlich] Kennen der Einschränkungen von Daten-Tiering-Instances
 1. Nur ElastiCache für Valkey- oder Redis OSS-Cluster verfügbar.
 2. Nur begrenzte Instance-Typen unterstützen Daten-Tiering.
 3. Nur ElastiCache Version 6.2 für Redis OSS und höher wird unterstützt
 4. Große Objekte werden nicht auf SSD ausgelagert. Objekte über 128 MiB werden im Speicher aufbewahrt.
- [Ressourcen]:
 - [Daten-Tiering](#)
 - [ElastiCacheAmazon-Preisgestaltung](#)
- [Erforderlich] Finden Sie heraus, auf welchen Anteil Ihrer Datenbank in Prozent Ihr Workload regelmäßig zugreift.
 1. Daten-Tiering-Instances eignen sich für Workloads, die häufig auf einen kleinen Teil Ihres gesamten Datensatzes zugreifen, aber dennoch einen schnellen Zugriff auf die verbleibenden Daten benötigen. Mit anderen Worten, das Verhältnis von heißen zu warmen Daten beträgt etwa 20:80.
 2. Entwickeln Sie eine Nachverfolgung der Leerlaufzeit von Objekten auf Clusterebene.
 3. Große Implementierungen mit über 500 GB Daten sind gute Kandidaten.
- [Erforderlich] Machen Sie sich bewusst, dass Daten-Tiering-Instances für bestimmte Workloads nicht optional sind.

1. Der Zugriff auf weniger häufig verwendete Objekte ist mit geringen Leistungseinbußen verbunden, da diese auf eine lokale SSD ausgelagert werden. Wenn Ihre Anwendung empfindlich auf die Reaktionszeit reagiert, testen Sie die Auswirkungen auf Ihren Workload.
2. Nicht geeignet für Caches, die hauptsächlich große Objekte mit einer Größe von über 128 MiB speichern.

[Ressourcen]:

- [Einschränkungen](#)
- [Am besten] Reserved-Instance-Typen unterstützen Daten-Tiering. Dadurch werden die niedrigsten Kosten in Bezug auf die Menge an Datenspeicher pro Instance gewährleistet.
 1. Möglicherweise müssen Sie Ihre ElastiCache Cluster mit Instances ohne Datenklassifizierung betreiben, bis Sie Ihre Anforderungen besser verstanden haben.
 2. Analysieren Sie das Datennutzungsmuster Ihrer ElastiCache Cluster.
 3. Erstellen Sie einen automatisierten Auftrag, der in regelmäßigen Abständen die Leerlaufzeit von Objekten erfasst.
 4. Wenn Sie feststellen, dass ein großer Prozentsatz (etwa 80 %) der Objekte für einen Zeitraum inaktiv ist, der für Ihren Workload als angemessen erachtet wird, dokumentieren Sie die Ergebnisse und schlagen Sie vor, den Cluster auf Instances zu migrieren, die Daten-Tiering unterstützen.
 5. Evaluieren Sie regelmäßig die neuen verfügbaren Cache-Knotentypen und beurteilen Sie, ob es aus Sicht der Kosten- und Betriebsmetriken sinnvoll ist, Ihre Instance-Flotte auf neue Cache-Knotentypen zu migrieren.

[Ressourcen]:

- [OBJECT IDLETIME](#)
- [ElastiCacheAmazon-Preisgestaltung](#)

Allgemeine Schritte zur Fehlerbehebung und bewährte Methoden mit ElastiCache

Die folgenden Themen enthalten Hinweise zur Behebung von Fehlern und Problemen, die bei der Verwendung auftreten können. ElastiCache Wenn Sie auf ein Problem stoßen, das hier nicht aufgeführt ist, können Sie die Schaltfläche „Feedback“ auf dieser Seite verwenden, um es zu melden.

Weitere Tipps zur Fehlerbehebung und Antworten auf häufig gestellte Supportfragen finden Sie im [AWS Knowledge Center](#)

Themen

- [Verbindungsprobleme](#)
- [Valkey- oder Redis OSS-Client-Fehler](#)
- [Fehlerbehebung bei hoher Latenz in Serverless ElastiCache](#)
- [Behebung von Drosselungsproblemen in Serverless ElastiCache](#)
- [Anhaltende Verbindungsprobleme](#)
- [Verwandte Themen](#)

Verbindungsprobleme

Wenn Sie keine Verbindung zu Ihrem ElastiCache Cache herstellen können, ziehen Sie eine der folgenden Möglichkeiten in Betracht:

1. Verwenden von TLS: Wenn beim Versuch, eine Verbindung zu Ihrem ElastiCache Endpunkt herzustellen, eine Verbindung nicht mehr hergestellt wird, verwenden Sie TLS möglicherweise nicht in Ihrem Client. Wenn Sie ElastiCache Serverless verwenden, ist die Verschlüsselung bei der Übertragung immer aktiviert. Stellen Sie sicher, dass Ihr Client TLS verwendet, um eine Verbindung zum Cache herzustellen. [Erfahren Sie mehr über das Herstellen einer Verbindung zu einem TLS-fähigen Cache.](#)
2. VPC: Auf ElastiCache Caches kann nur innerhalb einer VPC zugegriffen werden. Stellen Sie sicher, dass die EC2 Instance, von der aus Sie auf den Cache zugreifen, und der ElastiCache Cache in derselben VPC erstellt wurden. Alternativ müssen Sie das [VPC-Peering zwischen der VPC](#), in der sich Ihre EC2 Instance befindet, und der VPC, in der Sie Ihren Cache erstellen, aktivieren.
3. Sicherheitsgruppen: ElastiCache verwendet Sicherheitsgruppen, um den Zugriff auf Ihren Cache zu kontrollieren. Berücksichtigen Sie dabei Folgendes:
 - a. Stellen Sie sicher, dass die von Ihrem ElastiCache Cache verwendete Sicherheitsgruppe eingehenden Zugriff von Ihrer EC2 Instance aus darauf zulässt. [Hier](#) erfahren Sie, wie Sie Regeln für eingehenden Datenverkehr in Ihrer Sicherheitsgruppe korrekt einrichten.
 - b. Stellen Sie sicher, dass die von Ihrem ElastiCache Cache verwendete Sicherheitsgruppe den Zugriff auf die Ports Ihres Caches ermöglicht (6379 und 6380 für serverlose und 6379 standardmäßig für knotenbasierte Cluster). ElastiCache verwendet diese Ports, um Valkey-

oder Redis OSS-Befehle zu akzeptieren. [Erfahren Sie hier mehr darüber, wie Sie den Portzugriff einrichten.](#)

Wenn die Verbindung weiterhin schwierig ist, finden Sie [Anhaltende Verbindungsprobleme](#) weitere Schritte.

Valkey- oder Redis OSS-Client-Fehler

ElastiCache Auf Serverless kann nur über Clients zugegriffen werden, die das Valkey- oder Redis OSS-Clustermodus-Protokoll unterstützen. Auf knotenbasierte Cluster kann von Clients in beiden Modi zugegriffen werden, abhängig von der Clusterkonfiguration.

Wenn bei Ihrem Client Fehler auftreten, sollten Sie Folgendes beachten:

1. Clustermodus: Wenn bei Ihnen CROSSLOT-Fehler oder Fehler mit dem [SELECT-Befehl](#) auftreten, versuchen Sie möglicherweise, mit einem Valkey- oder Redis-OSS-Client, der das Cluster-Protokoll nicht unterstützt, auf einen Cache mit aktiviertem Clustermodus zuzugreifen. ElastiCache Serverless unterstützt nur Clients, die das Valkey- oder Redis OSS-Clusterprotokoll unterstützen. Wenn Sie Valkey oder Redis OSS im Modus „Cluster Mode Disabled“ (CMD) verwenden möchten, müssen Sie einen knotenbasierten Cluster erstellen.
2. CROSSLOT-Fehler: Wenn der `ERR CROSSLOT Keys in request don't hash to the same slot` Fehler auftritt, versuchen Sie möglicherweise, auf Schlüssel zuzugreifen, die nicht zu demselben Steckplatz in einem Clustermodus-Cache gehören. Zur Erinnerung: ElastiCache Serverless arbeitet immer im Clustermodus. Operationen mit mehreren Schlüsseln, Transaktionen oder Lua-Skripten mit mehreren Schlüsseln sind nur zulässig, wenn sich alle beteiligten Schlüssel im selben Hash-Slot befinden.

[Weitere bewährte Methoden zur Konfiguration von Valkey- oder Redis OSS-Clients finden Sie in diesem Blogbeitrag.](#)

Fehlerbehebung bei hoher Latenz in Serverless ElastiCache

Wenn bei Ihrem Workload eine hohe Latenz auftritt, können Sie anhand der `SuccessfulWriteRequestLatency` Messwerte CloudWatch `SuccessfulReadRequestLatency` und überprüfen, ob die Latenz mit ElastiCache Serverless zusammenhängt. Diese Metriken messen die Latenz, die innerhalb von ElastiCache Serverless liegt. Die clientseitige Latenz und die Netzwerkausfallzeiten zwischen Ihrem Client und dem ElastiCache serverlosen Endpunkt sind nicht enthalten.

Fehlerbehebung bei der clientseitigen Latenz

Wenn Sie eine erhöhte Latenz auf der Clientseite, aber keinen entsprechenden Anstieg CloudWatch `SuccessfulReadRequestLatency` der `SuccessfulWriteRequestLatency` Messwerte für die serverseitige Latenz feststellen, sollten Sie Folgendes beachten:

- Stellen Sie sicher, dass die Sicherheitsgruppe den Zugriff auf die Ports 6379 und 6380 zulässt: ElastiCache Serverless verwendet den 6379-Port für den primären Endpunkt und den 6380-Port für den Leser-Endpunkt. Einige Clients stellen für jede neue Verbindung eine Verbindung zu beiden Ports her, auch wenn Ihre Anwendung die Funktion „Aus Replikat lesen“ nicht verwendet. Wenn Ihre Sicherheitsgruppe keinen eingehenden Zugriff auf beide Ports zulässt, kann der Verbindungsaufbau länger dauern. Erfahren Sie [hier](#) mehr darüber, wie Sie den Portzugriff einrichten.

Fehlerbehebung bei serverseitiger Latenz

Einige Schwankungen und gelegentliche Spitzenwerte sollten keinen Anlass zur Sorge geben. Wenn die Average Statistik jedoch einen starken Anstieg zeigt und anhält, sollten Sie im Health Dashboard und in Ihrem Personal Health Dashboard nach weiteren Informationen suchen. Falls erforderlich, erwägen Sie, einen Support-Fall mit zu eröffnen. Support

Ziehen Sie die folgenden bewährten Methoden und Strategien zur Reduzierung der Latenz in Betracht:

- **Read from Replica aktivieren:** Wenn Ihre Anwendung dies zulässt, empfehlen wir, die Funktion „Read from Replica“ in Ihrem Valkey- oder Redis OSS-Client zu aktivieren, um Lesevorgänge zu skalieren und eine geringere Latenz zu erreichen. Wenn diese Option aktiviert ist, versucht ElastiCache Serverless, Ihre Leseanfragen an Replikat-Cache-Knoten weiterzuleiten, die sich in derselben Availability Zone (AZ) wie Ihr Client befinden, wodurch AZ-übergreifende Netzwerklatenzen vermieden werden. Beachten Sie, dass die Aktivierung der Funktion „Aus Replikat lesen“ in Ihrem Client bedeutet, dass Ihre Anwendung eine eventuelle Datenkonsistenz akzeptiert. Ihre Anwendung empfängt möglicherweise für einige Zeit ältere Daten, wenn Sie versuchen, sie zu lesen, nachdem Sie in einen Schlüssel geschrieben haben.
- Stellen Sie sicher, dass Ihre Anwendung im selben AZs Cache bereitgestellt wird: Möglicherweise stellen Sie eine höhere clientseitige Latenz fest, wenn Ihre Anwendung nicht im selben AZs Cache bereitgestellt wird. Wenn Sie einen serverlosen Cache erstellen, können Sie die Subnetze angeben, von denen aus Ihre Anwendung auf den Cache zugreift, und ElastiCache Serverless erstellt VPC-Endpunkte in diesen Subnetzen. Stellen Sie sicher, dass Ihre Anwendung in derselben

Umgebung bereitgestellt wird. AZs Andernfalls kann es bei Ihrer Anwendung beim Zugriff auf den Cache zu einem AZ-übergreifenden Hop kommen, was zu einer höheren clientseitigen Latenz führt.

- **Verbindungen wiederverwenden:** ElastiCache Serverlose Anfragen werden über eine TLS-fähige TCP-Verbindung unter Verwendung des RESP-Protokolls gestellt. Das Initiieren der Verbindung (einschließlich der Authentifizierung der Verbindung, falls konfiguriert) nimmt Zeit in Anspruch, sodass die Latenz der ersten Anfrage höher als üblich ist. Anfragen über eine bereits initialisierte Verbindung sorgen für eine gleichbleibend niedrige ElastiCache Latenz. Aus diesem Grund sollten Sie die Verwendung von Verbindungspooling oder die Wiederverwendung vorhandener Valkey- oder Redis OSS-Verbindungen in Betracht ziehen.
- **Skalierungsgeschwindigkeit:** ElastiCache Serverless skaliert automatisch, wenn Ihre Anforderungsrate steigt. Ein plötzlicher starker Anstieg der Anforderungsrate, der schneller ist als die Geschwindigkeit, mit der ElastiCache Serverless skaliert, kann für einige Zeit zu einer erhöhten Latenz führen. ElastiCache Serverless kann die unterstützte Anforderungsrate in der Regel schnell erhöhen. Es dauert bis zu 10 bis 12 Minuten, bis sich die Anforderungsrate verdoppelt.
- **Untersuchen Sie Befehle mit langer Laufzeit:** Einige Valkey- oder Redis-OSS-Befehle, einschließlich Lua-Skripten oder Befehle für große Datenstrukturen, können lange laufen. ElastiCache Veröffentlicht Metriken auf Befehlsebene, um diese Befehle zu identifizieren. Mit [ElastiCache Serverless](#) können Sie die BasedECPUs Metriken verwenden.
- **Gedrosselte Anfragen:** Wenn Anfragen in ElastiCache Serverless gedrosselt werden, kann es zu einem Anstieg der clientseitigen Latenz in Ihrer Anwendung kommen. [Wenn Anfragen in Serverless gedrosselt werden, sollten Sie einen ElastiCache Anstieg der Serverless-Metrik feststellen. ThrottledRequests ElastiCache](#) Im folgenden Abschnitt finden Sie Informationen zur Behebung gedrosselter Anfragen.
- **Gleichmäßige Verteilung von Schlüsseln und Anfragen:** ElastiCache Bei Valkey und Redis OSS kann eine ungleichmäßige Verteilung von Schlüsseln oder Anfragen pro Steckplatz zu einem Hot-Slot führen, was zu einer erhöhten Latenz führen kann. ElastiCache Serverless unterstützt bis zu 30.000 ECPUs/second (90.000 ECPUs/second bei Verwendung von Read from Replica) an einem einzigen Steckplatz in einer Arbeitslast, die einfache Befehle ausführt. SET/GET Wir empfehlen, Ihre Schlüssel- und Anforderungsverteilung auf die einzelnen Slots zu überprüfen und für eine gleichmäßige Verteilung zu sorgen, falls Ihre Anforderungsrate diese Grenze überschreitet.

Behebung von Drosselungsproblemen in Serverless ElastiCache

In serviceorientierten Architekturen und verteilten Systemen wird die Begrenzung der Geschwindigkeit, mit der API-Aufrufe durch verschiedene Servicekomponenten verarbeitet werden, als Drosselung bezeichnet. Dadurch werden Leistungsspitzen geglättet, Abweichungen im Komponentendurchsatz vermieden und bei unerwarteten Betriebsereignissen besser vorhersehbare Wiederherstellungen ermöglicht. ElastiCache Serverless wurde für diese Art von Architekturen konzipiert, und die meisten Valkey- oder Redis OSS-Clients verfügen über integrierte Wiederholungsversuche für gedrosselte Anfragen. Ein gewisses Maß an Drosselung ist nicht zwangsläufig ein Problem für Ihre Anwendung. Eine anhaltende Drosselung eines latenzsensitiven Teils des Datenworkflows kann sich jedoch negativ auf die Benutzererfahrung auswirken und die allgemeine Effizienz des Systems beeinträchtigen.

[Wenn Anfragen in Serverless gedrosselt werden, sollten Sie einen Anstieg der ElastiCache Serverless-Metrik feststellen. ThrottledRequests ElastiCache](#) Wenn Sie eine hohe Anzahl gedrosselter Anfragen feststellen, sollten Sie Folgendes beachten:

- **Skalierungsgeschwindigkeit:** ElastiCache Serverless wird automatisch skaliert, wenn Sie mehr Daten aufnehmen oder Ihre Anforderungsrate erhöhen. Wenn Ihre Anwendung schneller skaliert als die Geschwindigkeit, mit der Serverless skaliert, werden Ihre Anfragen möglicherweise gedrosselt, während ElastiCache Serverless an Ihre Arbeitslast angepasst wird. ElastiCache ElastiCache Serverless kann die Speichergröße in der Regel schnell erhöhen. Es dauert bis zu 10 bis 12 Minuten, bis sich die Speichergröße in Ihrem Cache verdoppelt hat.
- **Gleichmäßige Verteilung von Schlüsseln und Anfragen:** ElastiCache Bei Valkey und Redis OSS kann eine ungleichmäßige Verteilung von Schlüsseln oder Anfragen pro Steckplatz zu einem Hot-Slot führen. Ein Hot-Slot kann zu einer Drosselung von Anfragen führen, wenn die Anforderungsrate für einen einzelnen Slot 30.000 übersteigt ECPUs/second und es sich um eine Arbeitslast handelt, die einfache Befehle ausführt. SET/GET Ähnlich kann ElastiCache bei Memcached ein Hotkey zu einer Drosselung von Anfragen führen, wenn die Anforderungsrate 30.000 /Sekunde überschreitet. ECPUs
- **Aus Replikat lesen:** Wenn Ihre Anwendung dies zulässt, sollten Sie in Erwägung ziehen, die Funktion „Aus Replikat lesen“ zu verwenden. Die meisten Valkey- oder Redis-OSS-Clients können so konfiguriert werden, dass sie Lesevorgänge skalieren „, sodass Lesevorgänge direkt an Replikatknoten weitergeleitet werden. Mit dieser Funktion können Sie den Lesetraffic skalieren. Darüber hinaus leitet ElastiCache Serverless automatisch Lesevorgänge von Replikatanfragen an Knoten weiter, die sich in derselben Availability Zone wie Ihre Anwendung befinden, was zu einer geringeren Latenz führt. Wenn Read from Replica aktiviert ist, können Sie bei Workloads mit

einfachen Befehlen bis zu 90.000 ECPUs/second an einem einzigen Steckplatz erreichen. SET/GET

Anhaltende Verbindungsprobleme

Bei der Behebung anhaltender Verbindungsprobleme mit müssen die folgenden Punkte überprüft werden ElastiCache:

Themen

- [Sicherheitsgruppen](#)
- [Netzwerk ACLs](#)
- [Routing-Tabellen](#)
- [DNS-Auflösung](#)
- [Identifizieren von Problemen mit serverseitiger Diagnose](#)
- [Netzwerkverbindungsaktivität](#)
- [Netzwerkbezogene Grenzen](#)
- [CPU-Verwendung](#)
- [Verbindungen, die von der Serverseite beendet werden](#)
- [Clientseitige Fehlerbehebung für Amazon-Instances EC2](#)
- [Aufschlüsselung der Zeit, die zum Abschließen einer einzelnen Anfrage benötigt wird](#)

Sicherheitsgruppen

Sicherheitsgruppen sind virtuelle Firewalls, die Ihren ElastiCache Client (EC2 Instance, AWS Lambda Funktion, Amazon ECS-Container usw.) und Ihren ElastiCache Cache schützen. Sicherheitsgruppen sind statusbehaftet, was bedeutet, dass nach dem Zulassen des eingehenden oder ausgehenden Datenverkehrs die Antworten für diesen Datenverkehr automatisch im Kontext dieser bestimmten Sicherheitsgruppe autorisiert werden.

Die Statusfunktion erfordert, dass die Sicherheitsgruppe alle autorisierten Verbindungen verfolgt, und es gibt ein Limit für verfolgte Verbindungen. Wenn das Limit erreicht ist, schlagen neue Verbindungen fehl. Im Abschnitt zur Fehlerbehebung finden Sie Hilfe dazu, wie Sie feststellen können, ob die Grenzwerte auf dem Client oder auf der ElastiCache Seite erreicht wurden.

Sie können dem Client und dem ElastiCache Cluster eine einzelne Sicherheitsgruppe gleichzeitig oder einzelne Sicherheitsgruppen für jede Gruppe zuweisen.

In beiden Fällen müssen Sie den ausgehenden TCP-Verkehr auf dem ElastiCache Port von der Quelle und den eingehenden Verkehr auf demselben Port zulassen. ElastiCache Der Standardport ist 11211 für Memcached und 6379 für Valkey oder Redis OSS. Standardmäßig gestatten Sicherheitsgruppen allen ausgehenden Datenverkehr. In diesem Fall ist nur die eingehende Regel in der Zielsicherheitsgruppe erforderlich.

Weitere Informationen finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cluster in einer Amazon VPC](#).

Netzwerk ACLs

Network Access Control Lists (ACLs) sind statuslose Regeln. Der Datenverkehr muss in beide Richtungen (Eingehend und Ausgehend) zugelassen sein, um erfolgreich zu sein. Netzwerke ACLs werden Subnetzen zugewiesen, nicht bestimmten Ressourcen. Es ist möglich, dass der Client-Ressource dieselbe ACL zugewiesen ElastiCache wird, insbesondere wenn sie sich im selben Subnetz befinden.

Standardmäßig lässt das Netzwerk den gesamten ACLs Datenverkehr zu. Es ist jedoch möglich, sie anzupassen, um Datenverkehr zu verweigern oder zu erlauben. Darüber hinaus erfolgt die Auswertung der ACL-Regeln sequenziell, was bedeutet, dass die Regel mit der niedrigsten Zahl, die dem Datenverkehr entspricht, dies zulässt oder verweigert. Die Mindestkonfiguration, um den Valkey- oder Redis-OSS-Verkehr zuzulassen, ist:

Clientseitige Netzwerk-ACL:

- Regeln für eingehenden Datenverkehr:
- Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
- Type (Typ): Custom TCP Rule (TCP-Regel anpassen);
- Protokoll: TCP
- Portbereich: 1024-65535
- Quelle: 0.0.0.0/0 (oder erstellen Sie individuelle Regeln für die Cluster-Subnetze) ElastiCache
- Erlauben/Verweigern
- Regeln für ausgehenden Datenverkehr:

- Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
- Type (Typ): Custom TCP Rule (TCP-Regel anpassen);
- Protokoll: TCP
- Portbereich: 6379
- Quelle: 0.0.0.0/0 (oder die Cluster-Subnetze. ElastiCache Beachten Sie, dass die Verwendung bestimmter Optionen im Falle eines Failovers oder der Skalierung des Clusters zu Problemen führen IPs kann.
- Erlauben/Verweigern

ElastiCache Netzwerk-ACL:

- Regeln für eingehenden Datenverkehr:
 - Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
 - Type (Typ): Custom TCP Rule (TCP-Regel anpassen);
 - Protokoll: TCP
 - Portbereich: 6379
 - Quelle: 0.0.0.0/0 (oder erstellen Sie individuelle Regeln für die Cluster-Subnetze) ElastiCache
 - Erlauben/Verweigern
- Regeln für ausgehenden Datenverkehr:
 - Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
 - Type (Typ): Custom TCP Rule (TCP-Regel anpassen);
 - Protokoll: TCP
 - Portbereich: 1024-65535
 - Quelle: 0.0.0.0/0 (oder die Cluster-Subnetze. ElastiCache Beachten Sie, dass die Verwendung bestimmter Optionen im Falle eines Failovers oder der Skalierung des Clusters zu Problemen führen IPs kann.
 - Erlauben/Verweigern

Weitere Informationen finden Sie unter [Netzwerk ACLs](#).

Routing-Tabellen

Ähnlich wie beim Netzwerk ACLs kann jedes Subnetz unterschiedliche Routing-Tabellen haben. Wenn sich Clients und der ElastiCache Cluster in unterschiedlichen Subnetzen befinden, stellen Sie sicher, dass ihre Routing-Tabellen es ihnen ermöglichen, einander zu erreichen.

Bei komplexeren Umgebungen mit mehreren VPCs dynamischen Routing- oder Netzwerk-Firewalls kann es schwierig werden, Fehler zu beheben. Siehe [Netzwerkverbindbarkeit](#), um zu bestätigen, dass Ihre Netzwerkeinstellungen angemessen sind.

DNS-Auflösung

ElastiCache stellt die Dienstendpunkte auf der Grundlage von DNS-Namen bereit. Die verfügbaren Endpunkte sind `Configuration`, `Primary`, `Reader`, und `Node`-Endpunkte. Weitere Informationen finden Sie unter [Verbindungsendpunkte ermitteln](#).

Im Falle eines Failovers oder einer Clusteränderung kann sich die dem Endpunktnamen zugeordnete Adresse ändern und wird automatisch aktualisiert.

Benutzerdefinierte DNS-Einstellungen (d. h. wenn der VPC-DNS-Dienst nicht verwendet wird) kennen die ElastiCache bereitgestellten DNS-Namen möglicherweise nicht. Stellen Sie sicher, dass Ihr System die ElastiCache Endpunkte mithilfe von Systemtools wie `dig` (wie unten gezeigt) oder erfolgreich auflösen kann. `nslookup`

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com
example-001.xxxxxx.0001.use1.cache.amazonaws.com.
1.2.3.4
```

Sie können die Namensauflösung auch über den VPC DNS-Dienst erzwingen:

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com @169.254.169.253
example-001.tihewd.0001.use1.cache.amazonaws.com.
1.2.3.4
```

Identifizieren von Problemen mit serverseitiger Diagnose

CloudWatch Metriken und Laufzeitinformationen von der ElastiCache Engine sind gängige Informationsquellen zur Identifizierung potenzieller Ursachen von Verbindungsproblemen. Eine gute Analyse beginnt üblicherweise mit den folgenden Elementen:

- **CPU-Auslastung:** Valkey und Redis OSS sind Multithread-Anwendungen. Die Ausführung jedes Befehls erfolgt jedoch in einem einzigen (Haupt-) Thread. Aus diesem Grund ElastiCache bietet es die Metriken und. `CPUUtilization` `EngineCPUUtilization` `EngineCPUUtilization` liefert die CPU-Auslastung, die für den Valkey- oder Redis-OSS-Prozess vorgesehen ist, und `CPUUtilization` die Nutzung für alle V. CPUs Knoten mit mehr als einer vCPU haben normalerweise unterschiedliche Werte für `CPUUtilization` und `EngineCPUUtilization`, wobei der zweite Wert in der Regel höher ist. Hoch `EngineCPUUtilization` kann durch eine erhöhte Anzahl von Anforderungen oder komplexe Vorgänge verursacht werden, die eine erhebliche CPU-Zeit in Anspruch nehmen. Sie können beide folgendermaßen identifizieren:
- **Erhöhte Anzahl von Anforderungen:** Prüfen Sie auf Erhöhungen für andere Metriken, die den `EngineCPUUtilization`-Muster. Nützliche Metriken sind:
 - **CacheHitsundCacheMisses:** Die Anzahl der erfolgreichen Anforderungen oder Anforderungen, die kein gültiges Element im Cache gefunden haben. Wenn das Verhältnis von Fehlern im Vergleich zu Treffern hoch ist, verschwendet die Anwendung Zeit und Ressourcen mit unfruchtbaren Anfragen.
 - **SetTypeCmds und GetTypeCmds:** Diese Metriken, die mit `EngineCPUUtilization` korreliert sind, können helfen zu verstehen, ob die Last für Schreibanforderungen signifikant höher ist, gemessen durch `SetTypeCmds`, oder liest, gemessen durch `GetTypeCmds`. Wenn die Last überwiegend Lesevorgänge ist, kann die Verwendung mehrerer Read-Replicas die Anforderungen über mehrere Knoten hinweg ausgleichen und die primäre für Schreibvorgänge ersparen. In Clustern mit deaktiviertem Clustermodus können Read-Replicas verwendet werden, indem in der Anwendung mithilfe des Reader-Endpunkts eine zusätzliche Konfiguration erstellt wird. ElastiCache Weitere Informationen finden Sie unter [Verbindungsendpunkte ermitteln](#). Die Lesevorgänge müssen an diese zusätzliche Verbindung gesendet werden. Schreibvorgänge werden über den regulären primären Endpunkt durchgeführt. Im Clustermodus ist es ratsam, eine Bibliothek zu verwenden, die Read Replicas nativ unterstützt. Mit den richtigen Flags kann die Bibliothek automatisch die Cluster-Topologie und die Replikatknoten erkennen, die Lesevorgänge über den Befehl [READONLY Valkey oder Redis OSS aktivieren und die Leseanfragen](#) an die Replikat senden.
- **Erhöhte Anzahl von Verbindungen:**
 - **CurrConnectionsundNewConnections:** `CurrConnection` ist die Anzahl der etablierten Verbindungen zum Zeitpunkt der Datenpunkt-Sammlung, während `NewConnections` zeigt an, wie viele Verbindungen in der Periode erstellt wurden.

Das Erstellen und Behandeln von Verbindungen bedeutet einen erheblichen CPU-Overhead. Darüber hinaus wirkt sich der TCP-3-Wege-Handshake, der zum Erstellen neuer Verbindungen erforderlich ist, negativ auf die Gesamtreaktionszeiten aus.

Ein ElastiCache Knoten mit Tausenden `NewConnections` pro Minute bedeutet, dass eine Verbindung mit nur wenigen Befehlen hergestellt und verwendet wird, was nicht optimal ist. Es ist eine bewährte Vorgehensweise, Verbindungen herzustellen und sie für neue Vorgänge wiederzuverwenden. Dies ist möglich, wenn die Clientanwendung Verbindungspooling oder persistente Verbindungen unterstützt und ordnungsgemäß implementiert. Beim Verbindungspooling wird die Anzahl der `currConnections` hat keine großen Variationen, und die `NewConnection` sollte so niedrig wie möglich sein. Valkey und Redis OSS bieten eine optimale Leistung mit einer geringen Anzahl von `CurrConnections`. Wenn Sie `CurrConnection` in der Größenordnung von zehn oder Hunderten halten, wird die Nutzung von Ressourcen minimiert, um einzelne Verbindungen wie Clientpuffer und CPU-Zyklen zu unterstützen, um die Verbindung zu bedienen.

- **Netzwerkdurchsatz:**
 - Ermitteln Sie die Bandbreite: Die Netzwerkbandbreite der ElastiCache Knoten ist proportional zur Knotengröße. Da Anwendungen unterschiedliche Merkmale aufweisen, können die Ergebnisse je nach Workload variieren. Beispiele dafür sind Anwendungen mit einer hohen Rate kleiner Anforderungen eher die CPU-Auslastung als der Netzwerkdurchsatz, während größere Schlüssel eine höhere Netzwerkauslastung verursachen. Aus diesem Grund ist es ratsam, die Knoten mit der tatsächlichen Workload zu testen, um die Grenzen besser zu verstehen.

Die Simulation der Last aus der Anwendung würde genauere Ergebnisse liefern. Benchmark-Tools können jedoch eine gute Vorstellung von den Grenzen geben.

- In Fällen, in denen die Anforderungen überwiegend Lesevorgänge sind, verringert die Verwendung von Replikaten für Lesevorgänge die Belastung des primären Knotens. Wenn der Anwendungsfall überwiegend Schreibvorgänge ist, wird die Verwendung vieler Replikate die Netzwerknutzung verstärken. Für jedes Byte, das auf den primären Knoten geschrieben wird, werden N Bytes an die Replikate gesendet, wobei N die Anzahl der Replikate ist. Die bewährte Methode für schreibintensive Workloads ist die Verwendung von ElastiCache Redis OSS mit aktiviertem Cluster-Modus, sodass die Schreibvorgänge auf mehrere Shards verteilt oder auf einen Knotentyp mit mehr Netzwerkfähigkeiten skaliert werden können.

- Die CloudWatchmetrics NetworkBytesIn und NetworkBytesOut geben jeweils die Datenmenge an, die in den Knoten eingeht oder ihn verlässt. ReplicationBytes ist der Verkehr, der der Datenreplikation gewidmet ist.

Weitere Informationen finden Sie unter [Netzwerkbezogene Grenzen](#).

- Komplexe Befehle: Redis OSS-Befehle werden in einem einzigen Thread bedient, was bedeutet, dass Anfragen sequentiell bedient werden. Ein einzelner langsamer Befehl kann sich auf andere Anforderungen und Verbindungen auswirken, was zu Timeouts führt. Die Verwendung von Befehlen, die auf mehrere Werte, Schlüssel oder Datentypen wirken, muss sorgfältig durchgeführt werden. Verbindungen können abhängig von der Anzahl der Parameter oder der Größe der Ein- oder Ausgabewerte blockiert oder beendet werden.

Ein berüchtigtes Beispiel ist die KEYS-Befehl. Es fegt den gesamten Schlüsselraum auf der Suche nach einem bestimmten Muster und blockiert die Ausführung anderer Befehle während der Ausführung. Redis OSS verwendet die Notation „Big O“, um die Komplexität seiner Befehle zu beschreiben.

Keys Befehl hat $O(N)$ Zeitkomplexität, wobei N die Anzahl der Schlüssel in der Datenbank ist. Je größer die Anzahl der Schlüssel ist, desto langsamer wird der Befehl. KEYS kann auf verschiedene Arten Probleme verursachen: Wenn kein Suchmuster verwendet wird, gibt der Befehl alle verfügbaren Schlüsselnamen zurück. In Datenbanken mit tausend oder Millionen von Elementen wird eine riesige Ausgabe erstellt und die Netzwerkpuffer überflutet.

Wenn ein Suchmuster verwendet wird, werden nur die Schlüssel, die dem Muster entsprechen, an den Client zurückgegeben. Die Engine wird jedoch immer noch den gesamten Schlüsselraum durchsucht, und die Zeit, um den Befehl abzuschließen, ist gleich.

Eine Alternative für KEYS ist die SCAN-Befehl. Es iteriert über den Schlüsselraum und begrenzt die Iterationen in einer bestimmten Anzahl von Elementen, wodurch längere Blöcke auf der Engine vermieden werden.

Der Scan hat die COUNT, der zum Festlegen der Größe der Iterationsblöcke verwendet wird. Der Standardwert ist 10 (10 Elemente pro Iteration).

Abhängig von der Anzahl der Elemente in der Datenbank sind kleine COUNT-Werte-Blöcke erfordern mehr Iterationen, um einen vollständigen Scan abzuschließen, und größere Werte halten die Engine bei jeder Iteration länger beschäftigt. Während kleine Zählwerte SCAN

langsamer in großen Datenbanken machen, können größere Werte die gleichen Probleme verursachen, die für KEYS beschrieben sind.

Als Beispiel wird das Ausführen des SCAN-Befehl mit Zählwert als 10 erfordert 100.000 Wiederholungen in einer Datenbank mit 1 Million Schlüsseln. Wenn die durchschnittliche Netzwerk-Roundtrip-Zeit 0,5 Millisekunden beträgt, werden etwa 50.000 Millisekunden (50 Sekunden) für die Übertragung von Anforderungen ausgegeben.

Auf der anderen Seite, wenn der Zählwert 100.000 wäre, wäre eine einzelne Iteration erforderlich und nur 0,5 ms würden ausgegeben, um sie zu übertragen. Die Engine wäre jedoch für andere Operationen vollständig blockiert, bis der Befehl den gesamten Schlüsselraum beendet hat.

Außerdem KEYS, sind einige andere Befehle potenziell schädlich, wenn sie nicht korrekt verwendet werden. Eine Liste aller Befehle und ihrer jeweiligen Zeitkomplexität finden Sie unter [Valkey- und Redis OSS-Befehle](#).

Beispiele für mögliche Probleme:

- **Lua-Skripte:** Valkey und Redis OSS bieten einen eingebetteten Lua-Interpreter, der die Ausführung von Skripten auf der Serverseite ermöglicht. Lua-Skripte auf Valkey und Redis OSS werden auf Engine-Ebene ausgeführt und sind per Definition atomar, was bedeutet, dass kein anderer Befehl oder Skript ausgeführt werden darf, während ein Skript ausgeführt wird. Lua-Skripte bieten die Möglichkeit, mehrere Befehle, Entscheidungsalgorithmen, Datenanalyse und andere Befehle direkt auf der Engine auszuführen. Während die Atomizität von Skripten und die Möglichkeit, die Anwendung zu entladen, verlockend sind, müssen Skripte mit Sorgfalt und für kleine Operationen verwendet werden. Bei ElastiCache aktivierter Option ist die Ausführungszeit von Lua-Skripten auf 5 Sekunden begrenzt. Skripte, die nicht in den Schlüsselraum geschrieben wurden, werden nach Ablauf der 5 Sekunden automatisch beendet. Um Datenbeschädigungen und Inkonsistenzen zu vermeiden, wird der Knoten ein Failover ausgeführt, wenn die Skriptausführung nicht innerhalb von 5 Sekunden abgeschlossen wurde und während der Ausführung Schreibvorgänge stattfand. [Transaktionen](#) sind die Alternative, um die Konsistenz mehrerer verwandter wichtiger Änderungen in Redis OSS zu gewährleisten. Eine Transaktion ermöglicht die Ausführung eines Blocks von Befehlen und überwacht vorhandene Schlüssel auf Änderungen. Wenn sich einer der überwachten Schlüssel vor Abschluss der Transaktion ändert, werden alle Änderungen verworfen.
- **Massenlöschung von Elementen:** Die DEL akzeptiert mehrere Parameter, bei denen es sich um die Schlüsselnamen handelt, die gelöscht werden sollen. Löschvorgänge sind synchron und

erfordern erhebliche CPU-Zeit, wenn die Liste der Parameter groß ist oder eine große Liste, eine Menge, eine sortierte Menge oder einen Hash enthält (Datenstrukturen, die mehrere Unterelemente enthalten). Mit anderen Worten, selbst das Löschen eines einzelnen Schlüssels kann erhebliche Zeit in Anspruch nehmen, wenn es viele Elemente enthält. Die Alternative zu DEL ist UNLINK, einem asynchronen Befehl, der seit Redis OSS 4 verfügbar ist. UNLINK muss, DEL wann immer möglich, vorgezogen werden. Ab ElastiCache Redis OSS 6.x verhält sich der DEL Befehl aufgrund des `lazyfree-lazy-user-del` Parameters so, UNLINK als ob er aktiviert wäre. Weitere Informationen finden Sie unter [Redis OSS 6.0-Parameteränderungen](#).

- Befehle, die auf mehrere Tasten wirken: DEL wurde zuvor als Befehl erwähnt, der mehrere Argumente akzeptiert und seine Ausführungszeit wird direkt proportional dazu sein. Redis OSS bietet jedoch viele weitere Befehle, die ähnlich funktionieren. Als Beispiele MSET und MGET ermöglichen das gleichzeitige Einfügen oder Abrufen mehrerer String-Schlüssel. Ihre Nutzung kann vorteilhaft sein, um die Netzwerklatenz zu reduzieren, die mehreren einzelnen SET oder GET-Befehle. Eine umfangreiche Liste von Parametern wirkt sich jedoch auf die CPU-Auslastung aus.

Obwohl die CPU-Auslastung allein nicht die Ursache für Konnektivitätsprobleme ist, kann es zu viel Zeit für die Verarbeitung einzelner oder einiger Befehle über mehrere Schlüssel zu einem Ausfall anderer Anforderungen führen und die CPU-Gesamtauslastung erhöhen.

Die Anzahl der Schlüssel und ihre Größe beeinflussen die Komplexität des Befehls und damit die Fertigstellungszeit.

Weitere Beispiele für Befehle, die auf mehrere Tasten wirken können: HMGET, HMSET, MSETNX, PFCOUNT, PFMERGE, SDIFF, SDIFFSTORE, SINTER, SINTERSTORE, SUNION, SUNIONSTORE, TOUCH, ZDIFF, ZDIFFSTORE, ZINTER oder ZINTERSTORE.

- Befehle, die auf mehrere Datentypen einwirken: Redis OSS bietet auch Befehle, die auf einen oder mehrere Schlüssel wirken, unabhängig von ihrem Datentyp. ElastiCache für Redis bietet OSS die Metrik `KeyBasedCmds` zur Überwachung solcher Befehle. Diese Metrik summiert die Ausführung der folgenden Befehle im ausgewählten Zeitraum:
 - Komplexität von $O(N)$:
 - KEYS
 - $O(1)$
 - EXISTS
 - OBJECT
 - PTTL

- RANDOMKEY
- TTL
- TYPE
- EXPIRE
- EXPIREAT
- MOVE
- PERSIST
- PEXPIRE
- PEXPIREAT
- UNLINK ($O(N)$), um Speicher zurückzugewinnen. Die Speicherwiederherstellungsaufgabe geschieht jedoch in einem getrennten Thread und blockiert die Engine nicht
- Unterschiedliche Komplexitätszeiten je nach Datentyp:
 - DEL
 - DUMP
 - RENAME wird als Befehl mit $O(1)$ -Komplexität betrachtet, führt aber DEL intern. Die Ausführungszeit hängt von der Größe des umbenannten Schlüssels ab.
 - RENAMENX
 - RESTORE
 - SORT
- Große Hashes: Hash ist ein Datentyp, der einen einzelnen Schlüssel mit mehreren Schlüssel-Wert-Unterelementen erlaubt. Jeder Hash kann 4.294.967.295 Elemente speichern und Operationen auf großen Hashes können teuer werden. Ähnlich wie KEYS, haben Hashes die HKEYS Befehl mit $O(N)$ Zeitkomplexität, wobei N die Anzahl der Elemente im Hash ist. HSCAN hat Vorrang vor HKEYS, um lange laufende Befehle zu vermeiden. HDEL, HGETALL, HMGET, HMSET und HVALS sind Befehle, die bei großen Hashes mit Vorsicht verwendet werden sollten.
- Andere Big-Data-Strukturen: Neben Hashes können andere Datenstrukturen CPU-intensiv sein. Auch Sets, Listen, Sortierte Sets und Hyperloglogs können abhängig von ihrer Größe und den verwendeten Befehlen viel Zeit in Anspruch nehmen. Weitere Informationen zu diesen Befehlen finden Sie unter [Valkey- und Redis](#) OSS-Befehle.

Netzwerkverbindbarkeit

Nach der Überprüfung der Netzwerkkonfigurationen in Bezug auf DNS-Auflösung, Sicherheitsgruppen ACLs, Netzwerk- und Routingtabellen kann die Verbindbarkeit mit dem VPC Reachability Analyzer und den Systemtools überprüft werden.

Reachability Analyzer testet die Netzwerkkonnektivität und bestätigt, ob alle Anforderungen und Berechtigungen erfüllt sind. Für die folgenden Tests benötigen Sie die ENI-ID (Elastic Network Interface Identification) eines der in Ihrer VPC verfügbaren ElastiCache Knoten. Gehen Sie dazu wie folgt vor:

1. Zu <https://console.aws.amazon.com/ec2/v2/home gehen? #NIC:>
2. Filtern Sie die Schnittstellenliste nach Ihrem ElastiCache Clusternamen oder der IP-Adresse, die Sie zuvor bei den DNS-Validierungen erhalten haben.
3. Notieren oder auf andere Weise speichern Sie die ENI ID. Wenn mehrere Schnittstellen angezeigt werden, überprüfen Sie die Beschreibung, um sicherzustellen, dass sie zum richtigen ElastiCache Cluster gehören, und wählen Sie eine davon aus.
4. Fahren Sie mit dem nächsten Schritt fort.
5. Zu <https://console.aws.amazon.com/vpc/Hause einen Analysepfad erstellen? #> ReachabilityAnalyzer und wählen Sie die folgenden Optionen:
 - Quelltyp: Wählen Sie die Instance, wenn Ihr ElastiCache Client auf einer EC2 Amazon-Instance läuft, oder eine Netzwerkschnittstelle (falls er einen anderen Service verwendet, z. B. AWS Fargate Amazon ECS mit awsvpc-Netzwerk usw.) AWS Lambda, und die entsprechende Ressourcen-ID (EC2 Instance- oder ENI-ID);
 - Typ Ziel: Wählen Sie ausNetzwerkschnittstelleund wählen Sie das KontrollkästchenElasticache ENIaus der Liste.
 - Zielport: Geben Sie 6379 für Redis OSS oder 11211 ElastiCache für Memcached an. ElastiCache Dies sind die Ports, die mit der Standardkonfiguration definiert sind, und in diesem Beispiel wird davon ausgegangen, dass sie nicht geändert werden.
 - Protocol (Protokoll): TCP

Erstellen Sie den Analyse-Pfad und warten Sie ein paar Augenblicke auf das Ergebnis. Wenn der Status nicht erreichbar ist, öffnen Sie die Analysedetails und überprüfen Sie dieAnalyse-Explorerrfür Details, in denen die Anfragen blockiert wurden.

Wenn die Erreichbarkeitstests bestanden haben, fahren Sie mit der Überprüfung auf Betriebssystemebene fort.

Um die TCP-Konnektivität auf dem ElastiCache Service-Port zu überprüfen: Auf Amazon Linux Nping ist es im Paket enthalten nmap und kann die TCP-Konnektivität auf dem ElastiCache Port testen sowie die Netzwerk-Round-Trip-Zeit für den Verbindungsaufbau bereitstellen. Verwenden Sie diese Option, um die Netzwerkkonnektivität und die aktuelle Latenz zum ElastiCache Cluster zu überprüfen, wie im Folgenden dargestellt:

```
$ sudo nping --tcp -p 6379 example.xxxxxx.ng.0001.use1.cache.amazonaws.com

Starting Nping 0.6.40 ( http://nmap.org/nping ) at 2020-12-30 16:48 UTC
SENT (0.0495s) TCP ...
(Output suppressed )

Max rtt: 0.937ms | Min rtt: 0.318ms | Avg rtt: 0.449ms
Raw packets sent: 5 (200B) | Rcvd: 5 (220B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.08 seconds
```

In der Standardeinstellung nping sendet 5 Sonden mit einer Verzögerung von 1 Sekunde zwischen ihnen. Sie können die Option „-c“ verwenden, um die Anzahl der Prüfpunkte zu erhöhen und „—delay“, um die Zeit für das Senden eines neuen Tests zu ändern.

Wenn die Tests mit nping schlägt fehl und die VPC Reachability Analyzer-Tests bestanden haben, bitten Sie Ihren Systemadministrator, mögliche hostbasierte Firewall-Regeln, asymmetrische Routingregeln oder andere mögliche Einschränkungen auf Betriebssystemebene zu überprüfen.

Überprüfen Sie auf der ElastiCache Konsole, ob die Verschlüsselung während der Übertragung in Ihren ElastiCache Clusterdetails aktiviert ist. Wenn die Verschlüsselung bei der Übertragung aktiviert ist, bestätigen Sie, ob die TLS-Sitzung mit folgendem Befehl eingerichtet werden kann:

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

Eine umfangreiche Ausgabe wird erwartet, wenn die Verbindung und TLS-Verhandlung erfolgreich sind. Überprüfen Sie den in der letzten Zeile verfügbaren Rückgabecode, der Wert muss 0 (ok) sein. [Wenn openssl etwas anderes zurückgibt, überprüfen Sie den Grund für den Fehler unter https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS.](https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS)

Wenn alle Infrastruktur- und Betriebssystemtests bestanden wurden, Ihre Anwendung aber immer noch keine Verbindung herstellen kann ElastiCache, überprüfen Sie, ob die Anwendungskonfigurationen den ElastiCache Einstellungen entsprechen. Häufige Fehler sind:

- Ihre Anwendung unterstützt den ElastiCache Clustermodus nicht und der Clustermodus ElastiCache ist aktiviert;
- Ihre Anwendung unterstützt TLS/SSL nicht und die Verschlüsselung bei der Übertragung ElastiCache ist aktiviert;
- Die Anwendung unterstützt TLS/SSL , verfügt aber nicht über die richtigen Konfigurationsflags oder vertrauenswürdigen Zertifizierungsstellen;

Netzwerkbezogene Grenzen

- Maximale Anzahl von Verbindungen: Es gibt harte Grenzen für gleichzeitige Verbindungen. Jeder ElastiCache Knoten ermöglicht bis zu 65.000 gleichzeitige Verbindungen zwischen allen Clients. Dieses Limit kann mithilfe der eingeschalteten `CurrConnections` Metriken überwacht werden. CloudWatch Clients haben jedoch auch ihre Grenzen für ausgehende Verbindungen. Überprüfen Sie unter Linux den zulässigen flüchtigen Portbereich mit folgendem Befehl:

```
# sysctl net.ipv4.ip_local_port_range
net.ipv4.ip_local_port_range = 32768 60999
```

Im vorherigen Beispiel sind 28231 Verbindungen von derselben Quelle zu derselben Ziel-IP (ElastiCache Knoten) und demselben Zielport zulässig. Der folgende Befehl zeigt, wie viele Verbindungen für einen bestimmten ElastiCache Knoten (IP 1.2.3.4) bestehen:

```
ss --numeric --tcp state connected "dst 1.2.3.4 and dport == 6379" | grep -vE
'^State' | wc -l
```

Wenn die Zahl zu hoch ist, wird Ihr System möglicherweise überlastet und versucht, die Verbindungsanforderungen zu verarbeiten. Es ist ratsam, Techniken wie Verbindungspooling oder persistente Verbindungen zu implementieren, um die Verbindungen besser zu handhaben. Wenn möglich, konfigurieren Sie den Verbindungspool so, dass die maximale Anzahl von Verbindungen auf einige hundert begrenzt wird. Außerdem wäre eine Back-Off-Logik zur Behandlung von Timeouts oder anderen Verbindungsausnahmen ratsam, um im Falle von Problemen eine Verbindungsabwanderung zu vermeiden.

- Grenzwerte für den Netzwerkverkehr: Überprüfen Sie die folgenden [CloudWatch Metriken für Redis OSS](#), um mögliche Netzwerklimits zu identifizieren, die auf dem Knoten erreicht werden: ElastiCache
 - `NetworkBandwidthInAllowanceExceeded/NetworkBandwidthOutAllowanceExceeded`: Netzwerkpakete geformt, weil der Durchsatz das aggregierte Bandbreitenlimit überschritten hat.

Es ist wichtig zu beachten, dass jedes Byte, das auf den primären Knoten geschrieben wird, auf N Replikate repliziert wird, wobei N die Anzahl der Replikate ist. Cluster mit kleinen Knotentypen, mehreren Replikaten und intensiven Schreib Anforderungen können den Replikationsrückstand möglicherweise nicht bewältigen. In solchen Fällen ist es eine bewährte Methode, hochzuskalieren (Knoten-Typ ändern), aufzuskalieren (Shards in Cluster mit aktiviertem Cluster-Modus hinzufügen), die Anzahl der Replikate zu reduzieren oder die Anzahl der Schreibvorgänge zu minimieren.

- `NetworkConntrackAllowanceExceeded`: Pakete geformt, weil die maximale Anzahl von Verbindungen, die über alle dem Knoten zugewiesenen Sicherheitsgruppen nachverfolgt werden, überschritten wurde. Neue Verbindungen werden in diesem Zeitraum wahrscheinlich fehlschlagen.
- `NetworkPackets PerSecondAllowanceExceeded`: Maximale Anzahl von Paketen pro Sekunde überschritten. Arbeitslasten, die auf einer hohen Rate von sehr kleinen Anforderungen basieren, können diese Grenze vor der maximalen Bandbreite erreichen.

Die oben genannten Metriken sind der ideale Weg, um zu bestätigen, dass Knoten ihre Netzwerklimits erreichen. Grenzwerte können jedoch auch von Plateaus in Netzwerkmetriken identifiziert werden.

Wenn die Plateaus über längere Zeiträume beobachtet werden, folgen sie wahrscheinlich Replikationsverzögerung, Zunahme der Bytes, die für den Cache verwendet werden, Drop auf freien Speicher, hohe Swap und CPU-Auslastung. EC2 Amazon-Instances haben auch Netzwerklimits, die anhand von [ENA-Treibermetriken](#) verfolgt werden können. Linux-Instances mit erweiterter Netzwerkunterstützung und ENA-Treibern 2.2.10 oder höher können die Limitindikatoren mit folgendem Befehl überprüfen:

```
# ethtool -S eth0 | grep "allowance_exceeded"
```


CPU-Verwendung

Die Metrik zur CPU-Auslastung ist der Ausgangspunkt der Untersuchung, und die folgenden Punkte können helfen, mögliche Probleme ElastiCache nebenbei einzugrenzen:

- **Redis OSS SlowLogs:** In der ElastiCache Standardkonfiguration werden die letzten 128 Befehle beibehalten, deren Ausführung mehr als 10 Millisekunden gedauert hat. Der Verlauf der langsamen Befehle wird während der Motorlaufzeit beibehalten und geht im Falle eines Fehlers oder eines Neustarts verloren. Wenn die Liste 128 Einträge erreicht, werden alte Ereignisse entfernt, um Raum für neue zu öffnen. Die Größe der Liste der langsamen Ereignisse und die Ausführungszeit, die als langsam betrachtet wird, kann durch die Parameter `slowlog-max-len` und `slowlog-log-slower-than` in einer [Benutzerdefinierte Parametergruppe](#) angepasst werden. Die `slowlogs`-Liste kann abgerufen werden, indem `SLOWLOG GET 128` auf dem Motor, 128 sind die letzten 128 langsamen Befehle gemeldet. Jeder Spielereintrag hat folgende Felder:

```
1) 1) (integer) 1 -----> Sequential ID
   2) (integer) 1609010767 --> Timestamp (Unix epoch time) of the Event
   3) (integer) 4823378 -----> Time in microseconds to complete the command.
   4) 1) "keys" -----> Command
      2) "*" -----> Arguments
   5) "1.2.3.4:57004" -> Source
```

Das obige Ereignis geschah am 26. Dezember um 19:26:07 UTC, dauerte 4,8 Sekunden (4,823 ms) und wurde durch die `KEYS` Befehl vom Client 1.2.3.4 angefordert.

Unter Linux kann der Zeitstempel mit dem Befehlsdatum konvertiert werden:

```
$ date --date='@1609010767'
Sat Dec 26 19:26:07 UTC 2020
```

Mit Python:

```
>>> from datetime import datetime
>>> datetime.fromtimestamp(1609010767)
datetime.datetime(2020, 12, 26, 19, 26, 7)
```

Oder unter Windows mit: PowerShell

```
PS D:\Users\user> [datetimeoffset]::FromUnixTimeSeconds('1609010767')
```

```
DateTime      : 12/26/2020 7:26:07 PM
UtcDateTime   : 12/26/2020 7:26:07 PM
LocalDateTime : 12/26/2020 2:26:07 PM
Date          : 12/26/2020 12:00:00 AM
Day           : 26
DayOfWeek     : Saturday
DayOfYear     : 361
Hour          : 19
Millisecond    : 0
Minute        : 26
Month         : 12
Offset        : 00:00:00Ticks           : 637446075670000000
UtcTicks      : 637446075670000000
TimeOfDay     : 19:26:07
Year          : 2020
```

Viele langsame Befehle in kurzer Zeit (gleiche Minute oder weniger) sind ein Grund zur Besorgnis. Überprüfen Sie die Art der Befehle und wie sie optimiert werden können (siehe vorangegangene Beispiele). Wenn Befehle mit O (1) -Zeitkomplexität häufig gemeldet werden, überprüfen Sie die anderen Faktoren für eine hohe CPU-Auslastung, die zuvor erwähnt wurden.

- **Latenzmetriken:** ElastiCache Für Redis bietet OSS CloudWatch Metriken zur Überwachung der durchschnittlichen Latenz für verschiedene Befehlsklassen. Der Datenpunkt wird berechnet, indem die Gesamtzahl der Ausführungen von Befehlen in der Kategorie durch die gesamte Ausführungszeit in der Periode dividiert wird. Es ist wichtig zu verstehen, dass Latenzmetrikergebnisse ein Aggregat mehrerer Befehle sind. Ein einzelner Befehl kann unerwartete Ergebnisse wie Timeouts verursachen, ohne signifikante Auswirkungen auf die Metriken zu haben. In solchen Fällen wären die Slowlog-Ereignisse eine genauere Informationsquelle. Die folgende Liste enthält die verfügbaren Latenzmetriken und die entsprechenden Befehle, die sie betreffen.
 - **EvalBasedCmdsLatency:** bezieht sich auf Lua-Script-Befehle,; eval evalsha
 - **GeoSpatialBasedCmdsLatency:** geodist, geohash, geopos, georadius, georadiusbymember, geoaddd;
 - **GetTypeCmdsLatency:** Befehle lesen, unabhängig vom Datentyp;

- **HashBasedCmdsLatency:** hexists, hget, hgetall, hkeys, hlen, hmget, hvals, hstrlen, hdel, hincrby, hincrbyfloat, hset, hsetnx;
- **HyperLogLogBasedCmdsLatency:** pfselftest, pfcount, pfdebug, pfadd, pfmerge;
- **KeyBasedCmdsLatency:** Befehle, die auf verschiedene Datentypen einwirken können: dump, exists, keys, object, pttl, randomkey, ttl, type, del, expire, expireat, move, persist;
- **ListBasedCmdsLatency:** lindex, len, lrange, blpop, brpop, brpoplpush, linsert, lpop, push, pushx, lrem, let, ltrim, rpop, rpoplpush, rpush, rpushx;
- **PubSubBasedCmdsLatency:** abonnieren, veröffentlichen, pubsub, abbestellen, abonnieren, abbestellen;
- **SetBasedCmdsLatency:** scard, sdiff, sinter, sismember, smembers, srandmember, sunion, sadd, sdiffstore, sinterstore, smove, spop, srem, sunionstore;
- **SetTypeCmdsLatency:** Befehle schreiben, unabhängig vom Datentyp;
- **SortedSetBasedCmdsLatency:** zcard, zcount, zrange, zrangebyscore, zrank, zrevrange, zrevrangebyscore, zrevrank, zscore, zrangebylex, zrevrangebylex, zlexcount, zadd, zincrby, zinterstore, zrem, zremrangebyrank, zremrangebyscore, zunionstore, zremrangebylex, zpopmax, zpopmin, bzpopmin, bzpopmax;
- **StringBasedCmdsLatency:** bitcount, get, getbit, getrange, mget, strlen, substr, bitpos, append, bitop, bitfield, decr, decrby, getset, incr, incrby, incrbyfloat, mset, msetnx, psetex, set, setbit, setex, setnx, setrange;
- **StreamBasedCmdsLatency:** xrange, xrevrange, xlen, xread, xpending, xinfo, xadd, xgroup, readgroup, xack, xclaim, xdel, xtrim, xsetid;
- **Redis OSS-Laufzeitbefehle:**
 - **info commandstats:** Stellt eine Liste der Befehle bereit, die seit dem Start der Engine ausgeführt wurden, ihre kumulative Anzahl der Ausführungen, die Gesamtausführungszeit und die durchschnittliche Ausführungszeit pro Befehl;
 - **Client-Liste:** Bietet eine Liste der aktuell verbundenen Clients und relevante Informationen wie Pufferverwendung, zuletzt ausgeführter Befehl usw.;
- **Backup und Replikation:** Verwenden Sie ElastiCache für Redis OSS-Versionen vor 2.8.22 einen Fork-Prozess, um Backups zu erstellen und vollständige Synchronisationen mit den Replikaten durchzuführen. Diese Methode kann in erheblichem Speicheraufwand für schreibintensive Anwendungsfälle auftreten.

Beginnend mit ElastiCache Redis OSS 2.8.22 wurde eine Sicherungs- und Replikationsmethode ohne Forkless eingeführt. AWS Die neue Methode kann Schreibvorgänge verzögern, um Fehler zu vermeiden. Beide Methoden können Perioden höherer CPU-Auslastung verursachen, zu höheren Reaktionszeiten führen und somit zu Client-Timeouts während ihrer Ausführung führen. Überprüfen Sie immer, ob die Client-Fehler während des Backup-Fensters oder der `SaveInProgress`-Metrik war 1 in der Periode. Es ist ratsam, das Backup-Fenster für Zeiten geringer Auslastung zu planen, um die Möglichkeit von Problemen mit Clients oder Backup-Fehlern zu minimieren.

Verbindungen, die von der Serverseite beendet werden

Die Standardeinstellung ElastiCache für die Redis OSS-Konfiguration sorgt dafür, dass die Client-Verbindungen auf unbestimmte Zeit hergestellt werden. In einigen Fällen kann eine Verbindungsbeendigung jedoch wünschenswert sein. Zum Beispiel:

- Fehler in der Client-Anwendung können dazu führen, dass Verbindungen vergessen und im Leerlauf gehalten werden. Dies wird als „Verbindungsleck“ bezeichnet und die Folge ist eine stetige Zunahme der Anzahl etablierter Verbindungen, die auf der `CurrentConnections`-Metrik beobachtet werden. Dieses Verhalten kann zu einer Überlastung des Clients oder ElastiCache der Seite führen. Wenn eine sofortige Korrektur von der Client-Seite aus nicht möglich ist, legen einige Administratoren in ihrer ElastiCache Parametergruppe einen Wert für das „Timeout“ fest. Das Timeout ist die Zeit in Sekunden, die erlaubt ist, dass Verbindungen im Leerlauf bestehen bleiben. Wenn der Client innerhalb dieses Zeitraums keine Anfrage einreicht, beendet die Engine die Verbindung, sobald die Verbindung den Timeout-Wert erreicht. Kleine Zeitüberschreitungswerte können zu unnötigen Trennungen führen, und Clients müssen sie ordnungsgemäß behandeln und eine erneute Verbindung herstellen, was zu Verzögerungen führt.
- Der Speicher, der zum Speichern von Schlüsseln verwendet wird, wird für Clientpuffer freigegeben. Langsame Clients mit großen Anfragen oder Antworten benötigen möglicherweise eine beträchtliche Menge an Speicher, um ihre Puffer zu verarbeiten. Die Standardeinstellung ElastiCache für Redis OSS-Konfigurationen schränkt die Größe der regulären Client-Ausgabepuffer nicht ein. Wenn das `SymbolMemory-Limit` erreicht wird, versucht die Engine, Elemente zu vertreiben, um die Pufferverwendung zu erfüllen. Bei extrem wenig Arbeitsspeicher kann sich OSS ElastiCache für Redis dazu entscheiden, die Verbindung von Clients zu trennen, die große Client-Ausgabepuffer verbrauchen, um Speicherplatz freizugeben und den Zustand des Clusters zu erhalten.

Es ist möglich, die Größe von Clientpuffern mit benutzerdefinierten Konfigurationen zu begrenzen, und Clients, die das Limit erreichen, werden getrennt. Clients sollten jedoch in der Lage sein, unerwartete Trennungen zu verarbeiten. Die Parameter für die Puffergröße für reguläre Clients sind die folgenden:

- `client-query-buffer-limit`: Maximale Größe einer einzelnen Eingabeanforderung;
- `client-output-buffer-limit-normal-soft-limit`: Soft-Limit für Client-Verbindungen. Die Verbindung wird beendet, wenn sie länger als die für `client-output-buffer-limit` - definierte Zeit in Sekunden über dem Soft-Limit bleibt `normal-soft-seconds` oder wenn sie das Hard-Limit erreicht;
- `client-output-buffer-limit-normal-soft-seconds`: Zulässige Zeit für Verbindungen, die den Wert von `client-output-buffer-limit` - überschreiten `normal-soft-limit`;
- `client-output-buffer-limit-normal-hard-limit`: Eine Verbindung, die dieses Limit erreicht, wird sofort beendet.

Neben den regulären Client-Puffern steuern die folgenden Optionen den Puffer für Replikatknoten und Clients Pub/Sub (Publish/Subscribe):

- `client-output-buffer-limit-replica-hard-limit`;
- `client-output-buffer-limit-replica-soft-seconds`;
- `client-output-buffer-limit-replica-hard-limit`;
- `client-output-buffer-limit-pubsub-soft-limit`;
- `client-output-buffer-limit-pubsub-soft-seconds`;
- `client-output-buffer-limit-pubsub-hard-limit`;

Clientseitige Fehlerbehebung für Amazon-Instances EC2

Die Auslastung und Reaktionsfähigkeit auf der Client-Seite können sich auch auf die Anfragen auswirken. ElastiCache EC2 Bei der Behebung von zeitweiligen Verbindungs- oder Timeout-Problemen müssen die Beschränkungen für Instanzen und Betriebssysteme sorgfältig geprüft werden. Einige wichtige Punkte zu beachten:

- CPU:
 - EC2 CPU-Auslastung der Instanz: Stellen Sie sicher, dass die CPU nicht ausgelastet ist oder fast zu 100 Prozent ausgelastet ist. Die historische Analyse kann wie folgt durchgeführt werden. Beachten Sie jedoch CloudWatch, dass die Granularität der Datenpunkte entweder 1 Minute (bei aktivierter detaillierter Überwachung) oder 5 Minuten beträgt;

- Wenn Sie [EC2 Burstable-Instances](#) verwenden, stellen Sie sicher, dass ihr CPU-Guthaben nicht aufgebraucht ist. Diese Informationen sind in der Metrik verfügbar. CPUCreditBalance CloudWatch
- Kurze Perioden mit hoher CPU-Auslastung können zu Timeouts führen, ohne dass eine 100-prozentige Auslastung berücksichtigt wird. CloudWatch Solche Fälle erfordern eine Echtzeitüberwachung mit Betriebssystem-Tools wie `top`, `ps` und `mpstat`.
- Netzwerk
 - Überprüfen Sie, ob der Netzwerkdurchsatz gemäß den Instance-Funktionen unter akzeptablen Werten liegt. Weitere Informationen finden Sie unter [EC2 Amazon-Instance-Typen](#)
 - Bei Instances mit Erweiterten ena-Netzwerktreiber, überprüfen Sie die [EN-Statistiken](#) für Timeouts oder überschritten Limits. Die folgenden Statistiken sind nützlich, um die Sättigung der Netzwerklimits zu bestätigen:
 - `bw_in_allowance_exceeded/bw_out_allowance_exceeded`: Anzahl der Pakete, die durch übermäßigen eingehenden oder ausgehenden Durchsatz geformt werden;
 - `conntrack_allowance_exceeded`: Anzahl der Pakete, die aufgrund von Sicherheitsgruppen [Verbindungsverfolgung von Grenzen](#) gelöscht wurden. Neue Verbindungen werden fehlschlagen, wenn diese Grenze gesättigt ist;
 - `linklocal_allowance_exceeded`: Anzahl der Pakete, die aufgrund übermäßiger Anfragen an Instance-Meta-Daten gelöscht wurden, NTP über VPC DNS. Das Limit beträgt 1024 Pakete pro Sekunde für alle Dienste;
 - `pps_allowance_exceeded`: Anzahl der Pakete, die aufgrund übermäßiger Pakete pro Sekunde verworfen wurden. Das PPS-Limit kann erreicht werden, wenn der Netzwerkverkehr aus Tausenden oder Millionen sehr kleiner Anfragen pro Sekunde besteht. ElastiCache Der Datenverkehr kann optimiert werden, um Netzwerkpakete über Pipelines oder Befehle, die beispielsweise MGET mehrere Operationen gleichzeitig ausführen, besser zu nutzen. GET

Aufschlüsselung der Zeit, die zum Abschließen einer einzelnen Anfrage benötigt wird

- Im Netzwerk sind: `Tcpdump` und `Wireshark` (`tshark` auf der Befehlszeile) praktische Tools, um zu verstehen, wie viel Zeit die Anfrage benötigt hat, um das Netzwerk zu erreichen, die ElastiCache Engine zu starten und eine Antwort zu erhalten. Im folgenden Beispiel wird eine einzelne Anforderung hervorgehoben, die mit dem folgenden Befehl erstellt wurde:

```
$ echo ping | nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com 6379
```

```
+PONG
```

Parallel zum obigen Befehl wurde tcpdump ausgeführt und zurückgegeben:

```
$ sudo tcpdump -i any -nn port 6379 -tt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
1609428918.917869 IP 172.31.11.142.40966
    > 172.31.11.247.6379: Flags [S], seq 177032944, win 26883, options [mss
    8961,sackOK,TS val 27819440 ecr 0,nop,wscale 7], length 0
1609428918.918071 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [S.], seq
    53962565, ack 177032945, win
    28960, options [mss 1460,sackOK,TS val 3788576332 ecr 27819440,nop,wscale 7],
    length 0
1609428918.918091 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 1, win
    211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918122
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [P.], seq 1:6, ack 1, win 211,
    options [nop,nop,TS val 27819440 ecr 3788576332], length 5: RESP "ping"
1609428918.918132 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [F.], seq 6, ack
    1, win 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918240 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [.], ack 6, win
    227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918295
    IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [P.], seq 1:8, ack 7, win 227,
    options [nop,nop,TS val 3788576332 ecr 27819440], length 7: RESP "PONG"
1609428918.918300 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 8, win
    211, options [nop,nop,TS val 27819441 ecr 3788576332], length 0
1609428918.918302 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [F.], seq 8, ack
    7, win 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918307
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 9, win 211, options
    [nop,nop,TS val 27819441 ecr 3788576332], length 0
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

Aus der obigen Ausgabe können wir bestätigen, dass der TCP-3-Wege-Handshake in 222 Mikrosekunden (918091 - 917869) abgeschlossen wurde und der Ping-Befehl in 173 Mikrosekunden (918295 - 918122) zurückgegeben wurde.

Es dauerte 438 Mikrosekunden (918307 - 917869) vom Anfordern bis zum Schließen der Verbindung. Diese Ergebnisse würden bestätigen, dass Netz- und Triebwerksreaktionszeiten gut sind und sich die Untersuchung auf andere Komponenten konzentrieren kann.

- Auf dem Betriebssystem: Strace kann helfen, Zeitlücken auf Betriebssystemebene zu identifizieren. Die Analyse der tatsächlichen Anwendungen wäre viel umfangreicher und spezialisierte Anwendungsprofiler oder Debugger sind ratsam. Das folgende Beispiel zeigt nur, ob die Basisbetriebssystemkomponenten wie erwartet funktionieren, andernfalls können weitere Untersuchungen erforderlich sein. Wenn wir denselben Redis PING OSS-Befehl verwenden, erhalten `strace` wir:

```
$ echo ping | strace -f -tttt -r -e trace=execve,socket,open,recvfrom,sendto
nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com (http://
example.xxxxxx.ng.0001.use1.cache.amazonaws.com/)
6379
1609430221.697712 (+ 0.000000) execve("/usr/bin/nc", ["nc",
"example.xxxxxx.ng.0001.use1.cache.amazonaws.com", "6379"], 0x7ffede7cc38 /* 22 vars */) = 0
1609430221.708955 (+ 0.011231) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|
SOCK_NONBLOCK, 0) = 3
1609430221.709084
(+ 0.000124) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 3
1609430221.709258 (+ 0.000173) open("/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709637 (+ 0.000378) open("/etc/host.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709923
(+ 0.000286) open("/etc/resolv.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.711365 (+ 0.001443) open("/etc/hosts", O_RDONLY|O_CLOEXEC) = 3
1609430221.713293 (+ 0.001928) socket(AF_INET, SOCK_DGRAM|SOCK_CLOEXEC|SOCK_NONBLOCK,
IPPROTO_IP) = 3
1609430221.717419
(+ 0.004126) recvfrom(3, "\362|
\201\200\0\1\0\2\0\0\0\0\0\notls20201224\6tihew"... , 2048, 0, {sa_family=AF_INET,
sin_port=htons(53), sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 155
1609430221.717890 (+ 0.000469) recvfrom(3,
"\204\207\201\200\0\1\0\1\0\0\0\0\0\notls20201224\6tihew"... ,
65536, 0, {sa_family=AF_INET, sin_port=htons(53),
sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 139
1609430221.745659 (+ 0.027772) socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) = 3
1609430221.747548 (+ 0.001887) recvfrom(0, 0x7ffcf2f2ca50, 8192,
0, 0x7ffcf2f2c9d0, [128]) = -1 ENOTSOCK (Socket operation on non-socket)
1609430221.747858 (+ 0.000308) sendto(3, "ping\n", 5, 0, NULL, 0) = 5
1609430221.748048 (+ 0.000188) recvfrom(0, 0x7ffcf2f2ca50, 8192, 0, 0x7ffcf2f2c9d0,
[128]) = -1 ENOTSOCK
```



```
(Socket operation on non-socket)
1609430221.748330 (+ 0.000282) recvfrom(3, "+PONG\r\n", 8192, 0, 0x7ffcf2f2c9d0,
[128->0]) = 7
+PONG
1609430221.748543 (+ 0.000213) recvfrom(3, "", 8192, 0, 0x7ffcf2f2c9d0, [128->0]) = 0
1609430221.752110
(+ 0.003569) +++ exited with 0 +++
```

Im obigen Beispiel dauerte der Befehl etwas mehr als 54 Millisekunden ($752110 - 697712 = 54398$ Mikrosekunden).

Eine beträchtliche Zeit, etwa 20 ms, wurde benötigt, um nc zu instanziiieren und die Namensauflösung durchzuführen (von 697712 bis 717890), danach waren 2 ms erforderlich, um den TCP-Socket zu erstellen (745659 bis 747858) und 0,4 ms (747858 bis 748330), um die Antwort für die Anfrage zu senden und zu erhalten.

Verwandte Themen

- [the section called “Bewährte Methoden und Caching-Strategien”](#)

Sicherheit bei Amazon ElastiCache

Cloud-Sicherheit AWS hat höchste 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. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- **Sicherheit der Cloud** — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS -Compliance-Programme](#) regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für Amazon gelten ElastiCache, finden Sie unter [AWS Services in Scope by Compliance Program](#).
- **Sicherheit in der Cloud** — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung von Amazon anwenden können ElastiCache. In den folgenden Themen erfahren Sie, wie Sie Amazon konfigurieren ElastiCache , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Ihnen helfen, Ihre ElastiCache Amazon-Ressourcen zu überwachen und zu sichern.

Themen

- [Datenschutz bei Amazon ElastiCache](#)
- [Richtlinie für den Datenverkehr zwischen Netzwerken](#)
- [Identity and Access Management für Amazon ElastiCache](#)
- [Konformitätsvalidierung für Amazon ElastiCache](#)
- [Resilienz bei Amazon ElastiCache](#)
- [Infrastruktursicherheit in AWS ElastiCache](#)
- [Service-Updates in ElastiCache](#)
- [Common Vulnerabilities and Exposures \(CVE\): Sicherheitslücken, die behoben wurden in ElastiCache](#)

Datenschutz bei Amazon ElastiCache

Das [Modell der AWS gemeinsamen Verantwortung](#) und geteilter Verantwortung gilt für den Datenschutz in AWS ElastiCache (ElastiCache). Wie in diesem Modell beschrieben, AWS ist es für den Schutz der globalen Infrastruktur verantwortlich, auf der die gesamte AWS Cloud betrieben wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Dieser Inhalt umfasst die Sicherheitskonfiguration und die Verwaltungsaufgaben für die AWS Dienste, die Sie verwenden. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#).

Aus Datenschutzgründen empfehlen wir, die AWS Kontoanmeldeinformationen zu schützen und individuelle Konten mit AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem sollten Sie die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie TLS, um mit AWS Ressourcen zu kommunizieren.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen innerhalb der AWS Dienste.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu sichern.

Wir empfehlen dringend, in Freitextfeldern wie z. B. im Feld Name keine sensiblen, identifizierenden Informationen wie Kontonummern von Kunden einzugeben. Dazu gehört auch, wenn Sie mit ElastiCache oder anderen AWS Diensten arbeiten, die die Konsole, die API oder verwenden AWS SDKs. AWS CLI Alle Daten, die Sie in ElastiCache oder andere Services eingeben, werden möglicherweise in Diagnoseprotokolle aufgenommen. Wenn Sie eine URL für einen externen Server bereitstellen, schließen Sie keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL ein.

Themen

- [Datensicherheit bei Amazon ElastiCache](#)

Datensicherheit bei Amazon ElastiCache

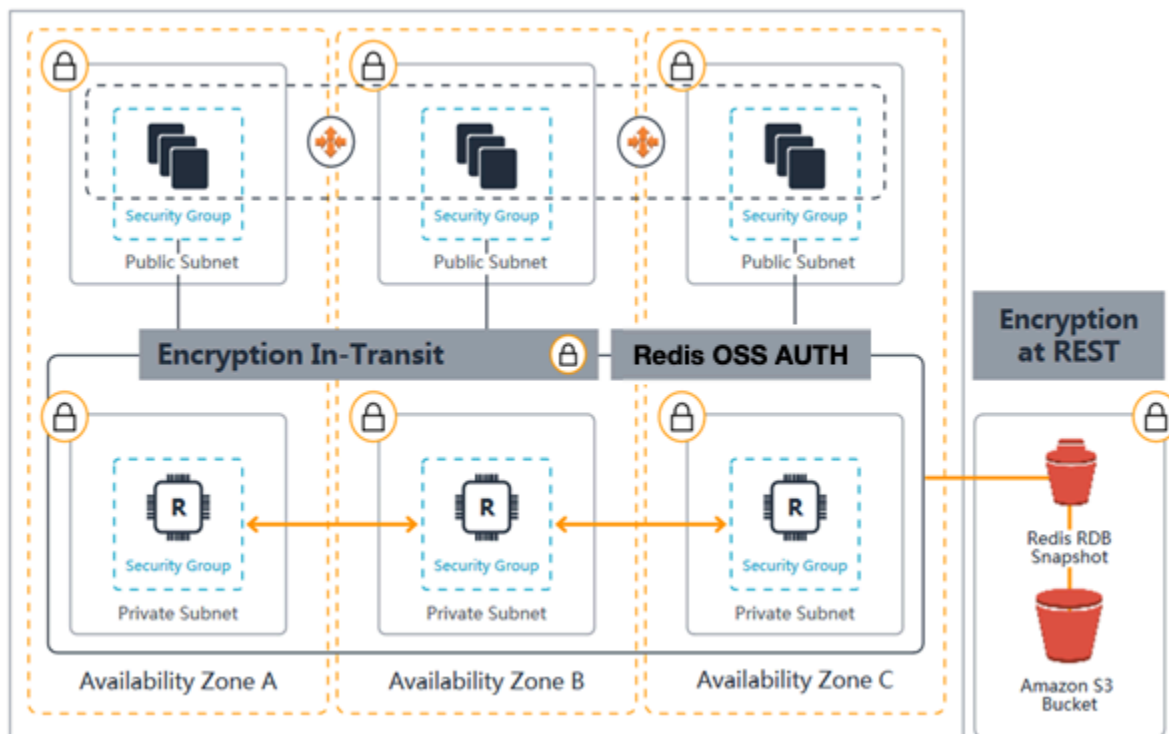
Um Ihre Daten zu schützen, EC2 bieten Amazon ElastiCache und Amazon Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten auf dem Server.

Amazon ElastiCache for Memcached bietet Verschlüsselungsfunktionen für Daten in Caches, auf denen Memcached-Versionen 1.6.12 oder höher ausgeführt werden.

Amazon ElastiCache mit Valkey und Redis OSS bietet Verschlüsselungsfunktionen für Daten in Caches, auf denen Valkey 7.2 oder höher ausgeführt wird, und Redis OSS-Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das [Ende des Lebenszyklus der Redis OSS-Versionen](#)), 4.0.10 oder höher. Amazon unterstützt ElastiCache auch die Authentifizierung von Benutzern mit IAM oder Valkey und Redis OSS AUTH sowie die Autorisierung von Benutzervorgängen mithilfe von Role-Based Access Control (RBAC).

- Bei der Verschlüsselung während der Übertragung werden Ihre Daten bei der Verschiebung von einem Ort an den anderen, z. B. zwischen Knoten in Ihrem Cluster oder zwischen einem Cache und Ihrer Anwendung, verschlüsselt.
- Bei der Verschlüsselung im Ruhezustand werden Ihre Daten auf dem Laufwerk bei Synchronisierungs- und Sicherungsoperationen verschlüsselt.

ElastiCache unterstützt die Authentifizierung von Benutzern mithilfe von IAM und den Befehlen Valkey und Redis OSS AUTH sowie die Autorisierung von Benutzervorgängen mithilfe von Role-Based Access Control (RBAC).



ElastiCache für das OSS-Sicherheitsdiagramm von Valkey und Redis

Themen

- [ElastiCache Verschlüsselung bei der Übertragung \(TLS\)](#)
- [Verschlüsselung im Ruhezustand in ElastiCache](#)
- [Authentifizierung und Autorisierung](#)

ElastiCache Verschlüsselung bei der Übertragung (TLS)

Um Ihre Daten zu schützen, EC2 bieten Amazon ElastiCache und Amazon Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten auf dem Server. Durch die Bereitstellung von Verschlüsselungsfunktionen während der Übertragung ElastiCache erhalten Sie ein Tool, mit dem Sie Ihre Daten schützen können, wenn sie von einem Ort zum anderen übertragen werden.

In allen serverlosen Valkey- oder Redis OSS-Caches ist die Verschlüsselung bei der Übertragung aktiviert. Für knotenbasierte Cluster können Sie die Verschlüsselung während der Übertragung für eine Replikationsgruppe aktivieren, indem Sie den Parameter `TransitEncryptionEnabled` auf `true` (CLI: `--transit-encryption-enabled`) setzen, wenn Sie die Replikationsgruppe erstellen. Sie können dies unabhängig davon tun, ob Sie die Replikationsgruppe mithilfe der AWS-Managementkonsole, der oder der API AWS CLI erstellen. ElastiCache

Für alle Serverless-Caches ist die Verschlüsselung während der Übertragung aktiviert. Für knotenbasierte Cluster können Sie die Verschlüsselung während der Übertragung auf einem Cluster aktivieren, indem Sie den Parameter `TransitEncryptionEnabled` auf `true` (CLI: `--transit-encryption-enabled`) setzen, wenn Sie den Cluster mit dem Vorgang `CreateCacheCluster` (CLI: `create-cache-cluster`) erstellen.

Themen

- [Übersicht über die Verschlüsselung während der Übertragung](#)
- [Verschlüsselungsbedingungen bei der Übertragung \(Valkey und Redis OSS\)](#)
- [Verschlüsselungsbedingungen bei der Übertragung \(Memcached\)](#)
- [Bewährte Methoden für die Verschlüsselung während der Übertragung](#)
- [Weitere OSS-Optionen von Valkey und Redis](#)
- [Aktivierung der Verschlüsselung bei der Übertragung für Memcached](#)
- [Aktivieren von Verschlüsselung während der Übertragung](#)
- [Verbindung zu ElastiCache \(Valkey\) oder Amazon ElastiCache for Redis OSS mit Verschlüsselung während der Übertragung mithilfe von Valkey-CLI herstellen](#)
- [Aktivierung der Verschlüsselung während der Übertragung auf einem knotenbasierten Redis-OSS-Cluster mithilfe von Python](#)
- [Bewährte Methoden für die Aktivierung der Verschlüsselung während der Übertragung](#)
- [Mit Openssl \(Memcached\) wird eine Verbindung zu Knoten hergestellt, für die Verschlüsselung während der Übertragung aktiviert ist](#)
- [Erstellen eines TLS-Memcached-Clients mit Java](#)
- [Erstellen eines TLS-Memcached-Clients mit PHP](#)

Übersicht über die Verschlüsselung während der Übertragung

Die ElastiCache Amazon-Verschlüsselung bei der Übertragung ist eine Funktion, mit der Sie die Sicherheit Ihrer Daten an den anfälligsten Stellen erhöhen können — wenn sie von einem Ort zum anderen übertragen werden. Da für die Verschlüsselung und Entschlüsselung der Daten an den Endpunkten eine gewisse Verarbeitung erforderlich ist, kann die Aktivierung der Verschlüsselung während der Übertragung einige Performance-Einbußen zufolge haben. Sie sollten Benchmark-Tests Ihrer Daten mit und ohne Verschlüsselung während der Übertragung durchführen, um die Performance-Einbußen bei Ihren Anwendungsfällen zu bestimmen.

ElastiCache Die Verschlüsselung bei der Übertragung implementiert die folgenden Funktionen:

- Verschlüsselte Client-Verbindungen – Client-Verbindungen mit Cache-Knoten sind TLS-verschlüsselt.
- Verschlüsselte Serververbindungen – Daten, die zwischen Knoten in einem Cluster übertragen werden, sind verschlüsselt.
- Serverauthentifizierung – Clients können die Verbindung zum richtigen Server authentifizieren.
- Client-Authentifizierung — Mithilfe der Funktionen Valkey und Redis OSS AUTH kann der Server die Clients authentifizieren.

Verschlüsselungsbedingungen bei der Übertragung (Valkey und Redis OSS)

Bei der Planung Ihrer knotenbasierten ElastiCache Cluster-Implementierung sollten Sie die folgenden Einschränkungen der In-Transit-Verschlüsselung von Amazon berücksichtigen:

- Die Verschlüsselung während der Übertragung wird für Replikationsgruppen unterstützt, auf denen Valkey 7.2 und höher sowie die Redis OSS-Versionen 3.2.6, 4.0.10 und höher ausgeführt werden.
- Das Ändern der Verschlüsselungseinstellung bei der Übertragung für einen vorhandenen Cluster wird für Replikationsgruppen unterstützt, auf denen Valkey 7.2 und höher sowie Redis OSS Version 7 und höher ausgeführt wird.
- Die Verschlüsselung während der Übertragung wird nur für Replikationsgruppen unterstützt, die in einer Amazon VPC ausgeführt werden.
- Die Verschlüsselung während der Übertragung wird für Replikationsgruppen, auf denen die folgenden Knotentypen ausgeführt werden, nicht unterstützt: M1, M2.

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Verschlüsselung während der Übertragung wird durch die Festlegung des `TransitEncryptionEnabled`-Parameters auf `true` ausdrücklich aktiviert.
- Stellen Sie sicher, dass Ihr Caching-Client TLS-Konnektivität unterstützt und dass Sie diese in der Client-Konfiguration aktiviert haben.
- Ab dem 26. Januar 2026 AWS wird die unterstützte Mindestversion von TLS ElastiCache für Valkey Version 7.2 und höher sowie ElastiCache für Redis OSS Version 6 und höher auf 1.2 aktualisiert. Kunden müssen ihre Client-Software vor diesem Datum aktualisieren. Dieses Update hilft Ihnen dabei, Sicherheits-, Compliance- und behördliche Anforderungen zu erfüllen.

Verschlüsselungsbedingungen bei der Übertragung (Memcached)

Bei der Planung Ihrer knotenbasierten ElastiCache Cluster-Implementierung sollten Sie die folgenden Einschränkungen der In-Transit-Verschlüsselung von Amazon berücksichtigen:

- Verschlüsselung während der Übertragung wird auf Clustern mit der Memcached-Version 1.6.12 und höher unterstützt.
- Verschlüsselung während der Übertragung unterstützt die Transport-Layer-Security(TLS)-Versionen 1.2 und 1.3.
- Die Verschlüsselung während der Übertragung wird nur für Cluster unterstützt, die in einer Amazon VPC ausgeführt werden.
- Die Verschlüsselung während der Übertragung wird für Replikationsgruppen, auf denen die folgenden Knotentypen ausgeführt werden, nicht unterstützt: M1, M2, M3, R3, T2.

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Verschlüsselung während der Übertragung wird durch die Festlegung des `TransitEncryptionEnabled`-Parameters auf `true` ausdrücklich aktiviert.
- Sie können die Verschlüsselung während der Übertragung nur für einen Cluster aktivieren, wenn Sie den Cluster erstellen. Sie können die Verschlüsselung während der Übertragung nicht aktivieren und deaktivieren, indem Sie einen Cluster ändern.
- Stellen Sie sicher, dass Ihr Caching-Client TLS-Konnektivität unterstützt und dass Sie diese in der Client-Konfiguration aktiviert haben.

Bewährte Methoden für die Verschlüsselung während der Übertragung

- Aufgrund der für die Verschlüsselung und Entschlüsselung der Daten an den Endpunkten erforderliche Verarbeitung, kann die Implementierung der Verschlüsselung während der Übertragung eine Verringerung der Performance zur Folge haben. Führen Sie einen Benchmark-Test der Verschlüsselung während der Übertragung im Vergleich zu keiner Verschlüsselung Ihrer eigenen Daten durch, um deren Auswirkungen auf die Performance Ihrer Implementierung zu ermitteln.
- Da der Aufbau neuer Verbindungen kostspielig sein kann, können Sie die Auswirkungen der Verschlüsselung während der Übertragung auf die Leistung verringern, indem Sie Ihre TLS-Verbindungen aufrechterhalten.

Weitere OSS-Optionen von Valkey und Redis

Weitere Informationen zu den für Valkey und Redis OSS verfügbaren Optionen finden Sie unter den folgenden Links.

- [Verschlüsselung im Ruhezustand in ElastiCache](#)
- [Authentifizierung mit dem Befehl Valkey und Redis OSS AUTH](#)
- [Rollenbasierte Zugriffskontrolle \(RBAC\)](#)
- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Identity and Access Management für Amazon ElastiCache](#)

Aktivierung der Verschlüsselung bei der Übertragung für Memcached

Treffen Sie zum Aktivieren der Verschlüsselung während der Übertragung beim Erstellen eines Memcached-Clusters mit der AWS-Managementkonsole die folgende Auswahl:

- Wählen Sie Memcached als Ihre Engine.
- Wählen Sie die Engine-Version 1.6.12 und höher.
- Wählen Sie unter Encryption in transit (Verschlüsselung während der Übertragung) Enable (Aktivieren) aus.

Informationen zum step-by-step Vorgang finden Sie unter. [Einen Cluster für Valkey oder Redis OSS erstellen](#)

Aktivieren von Verschlüsselung während der Übertragung

Für alle Serverless-Caches ist die Verschlüsselung während der Übertragung aktiviert. In einem knotenbasierten Cluster können Sie die Verschlüsselung während der Übertragung mithilfe der AWS-Managementkonsole, AWS CLI oder der API aktivieren. ElastiCache

Aktivieren der Verschlüsselung bei der Übertragung mithilfe der AWS-Managementkonsole

Aktivieren der Verschlüsselung bei der Übertragung für einen neuen knotenbasierten Cluster mithilfe der AWS-Managementkonsole

Wenn Sie einen eigenen Cluster entwerfen, ist bei den Konfigurationen „Dev/Test“ und „Production“ mit der Methode „Easy Create“ die Verschlüsselung während der Übertragung aktiviert. Legen Sie die folgenden Einstellungen fest, wenn Sie die Konfiguration selbst auswählen:

- Wählen Sie die Engine-Version 3.2.6, 4.0.10 und höher.
- Aktivieren Sie für die Option Verschlüsselung während der Übertragung das Kontrollkästchen neben Aktivieren.

Informationen zum step-by-step Prozess finden Sie im Folgenden:

- [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Aktivieren der Verschlüsselung bei der Übertragung für einen vorhandenen knotenbasierten Cluster mithilfe der AWS-Managementkonsole

Die Aktivierung der Verschlüsselung während der Übertragung ist ein zweistufiger Vorgang. Sie müssen zuerst den Modus der Übertragungsverschlüsselung auf `preferred` festlegen. In diesem Modus können Ihre Valkey- oder Redis OSS-Clients eine Verbindung sowohl über verschlüsselte als auch über unverschlüsselte Verbindungen herstellen. Nachdem Sie alle Ihre Valkey- oder Redis OSS-Clients auf verschlüsselte Verbindungen migriert haben, können Sie Ihre Clusterkonfiguration ändern, um den Übertragungsverschlüsselungsmodus auf `required` einzustellen. Wenn Sie den Modus der Verschlüsselung während der Übertragung auf `required` festlegen, werden alle unverschlüsselten Verbindungen gelöscht und nur verschlüsselte Verbindungen werden zugelassen.

Stellen Sie Ihren Transit-Verschlüsselungsmodus auf Bevorzugt ein

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie Valkey-Caches oder Redis OSS-Caches aus den ElastiCache Ressourcen, die im Navigationsbereich auf der linken Seite angezeigt werden.
3. Wählen Sie den Cache aus, den Sie aktualisieren möchten.
4. Wählen Sie das Dropdown-Menü Actions (Aktionen) und dann Modify (Ändern) aus.
5. Wählen Sie im Bereich Sicherheit unter Verschlüsselung während der Übertragung die Option Aktivieren aus.
6. Wählen Sie Preferred (Bevorzugt) als den Transit encryption mode (Modus der Verschlüsselung während der Übertragung) aus.
7. Wählen Sie Preview changes (Änderungsvorschau anzeigen) aus und speichern Sie die Änderungen.

Nachdem Sie alle Ihre Valkey- oder Redis OSS-Clients migriert haben, um verschlüsselte Verbindungen zu verwenden:

Stellen Sie Ihren Transit-Verschlüsselungsmodus auf Erforderlich

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie Valkey-Caches oder Redis OSS-Caches aus den ElastiCache Ressourcen, die im Navigationsbereich auf der linken Seite angezeigt werden.
3. Wählen Sie den Cache aus, den Sie aktualisieren möchten.
4. Wählen Sie das Dropdown-Menü Actions (Aktionen) und dann Modify (Ändern) aus.
5. Wählen Sie im Bereich Security (Sicherheit) die Option Required (Erforderlich) als Transit encryption mode (Modus der Verschlüsselung während der Übertragung) aus.
6. Wählen Sie Preview changes (Änderungsvorschau anzeigen) aus und speichern Sie die Änderungen.

Aktivieren Sie die Verschlüsselung bei der Übertragung mit dem AWS CLI

Verwenden Sie den Parameter, um die Verschlüsselung während der Übertragung zu aktivieren, wenn Sie eine Valkey- oder Redis OSS-Replikationsgruppe mithilfe von erstellen. `AWS CLI transit-encryption-enabled`

Aktivierung der Verschlüsselung während der Übertragung auf einem neuen knotenbasierten Cluster für Valkey oder Redis OSS (Clustermodus deaktiviert) (CLI)

Verwenden Sie den AWS CLI Vorgang `create-replication-group` und die folgenden Parameter, um eine Valkey- oder Redis OSS-Replikationsgruppe mit Replikaten zu erstellen, für die die Verschlüsselung bei der Übertragung aktiviert ist:

Hauptparameter:

- **--engine**— Muss oder sein. `valkey` `redis`
- **--engine-version**—Wenn es sich bei der Engine um Redis OSS handelt, muss es sich um 3.2.6, 4.0.10 oder höher handeln.
- **--transit-encryption-enabled**—Erforderlich. Wenn Sie die Verschlüsselung während der Übertragung aktivieren, müssen Sie zudem einen Wert für den `--cache-subnet-group`-Parameter angeben.

- **--num-cache-clusters**—Muss mindestens 1 sein. Der Höchstwert für diesen Parameter ist 6.

Weitere Informationen finden Sie hier:

- [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Clustermodus deaktiviert\) von Grund auf neu erstellen \(\)AWS CLI](#)
- [create-replication-group](#)

Aktivierung der Verschlüsselung während der Übertragung auf einem neuen knotenbasierten Cluster für Valkey oder Redis OSS (Cluster Mode Enabled) (CLI)

Verwenden Sie den AWS CLI Vorgang `create-replication-group` und die folgenden Parameter, um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) zu erstellen, für die die Verschlüsselung bei der Übertragung aktiviert ist:

Hauptparameter:

- **--engine**— Muss oder sein. `valkey redis`
- **--engine-version**—Wenn es sich bei der Engine um Redis OSS handelt, muss es sich um 3.2.6, 4.0.10 oder höher handeln.
- **--transit-encryption-enabled**—Erforderlich. Wenn Sie die Verschlüsselung während der Übertragung aktivieren, müssen Sie zudem einen Wert für den `--cache-subnet-group`-Parameter angeben.
- Verwenden Sie einen der folgenden Parametersätze, um die Konfiguration der Knotengruppen der Replikationsgruppe anzugeben.
 - **--num-node-groups**—Zur Angabe der Anzahl der Shards (Knoten-Gruppen) in dieser Replikationsgruppe. Der Höchstwert für diesen Parameter ist 500.
 - **--replicas-per-node-group**—Definiert die Anzahl der Replikat-Knoten in jeder Knotengruppe. Der hier angegebene Wert wird auf allen Shards in dieser Replikationsgruppe angewendet. Der Höchstwert für diesen Parameter ist 5.
 - **--node-group-configuration**—Spezifiziert die Konfiguration jedes Shards unabhängig.

Weitere Informationen finden Sie hier:

- [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Cluster Mode Enabled\) von Grund auf neu erstellen \(\)AWS CLI](#)
- [create-replication-group](#)

Aktivieren der Verschlüsselung während der Übertragung für einen vorhandenen Cluster mithilfe der AWS CLI

Die Aktivierung der Verschlüsselung während der Übertragung ist ein zweistufiger Vorgang. Sie müssen zuerst den Modus der Übertragungsverschlüsselung auf `preferred` festlegen. In diesem Modus können Ihre Valkey- oder Redis OSS-Clients Verbindungen sowohl über verschlüsselte als auch über unverschlüsselte Verbindungen herstellen. Nachdem Sie alle Ihre Valkey- oder Redis OSS-Clients auf verschlüsselte Verbindungen migriert haben, können Sie Ihre Clusterkonfiguration ändern, um den Übertragungsverschlüsselungsmodus auf `required` einzustellen. Wenn Sie den Modus der Verschlüsselung während der Übertragung auf `required` festlegen, werden alle unverschlüsselten Verbindungen gelöscht und nur verschlüsselte Verbindungen werden zugelassen.

Verwenden Sie den AWS CLI Vorgang `modify-replication-group` und die folgenden Parameter, um eine Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) zu aktualisieren, für die die Verschlüsselung während der Übertragung deaktiviert ist.

So aktivieren Sie die Verschlüsselung während der Übertragung

1. Stellen Sie unter Verwendung `transit-encryption-mode` der `preferred` folgenden Parameter auf ein
 - **--transit-encryption-enabled**—Erforderlich.
 - **--transit-encryption-mode** – muss auf `preferred` festgelegt sein.
2. Stellen `transit-encryption-mode` Sie `required` unter Verwendung der folgenden Parameter auf ein:
 - **--transit-encryption-enabled**—Erforderlich.
 - **--transit-encryption-mode** – muss auf `required` festgelegt sein.

Verbindung zu ElastiCache (Valkey) oder Amazon ElastiCache for Redis OSS mit Verschlüsselung während der Übertragung mithilfe von Valkey-CLI herstellen

Um auf Daten von Redis OSS-Caches zuzugreifen, ElastiCache für die Verschlüsselung während der Übertragung aktiviert ist, verwenden Sie Clients, die mit Secure Socket Layer (SSL) arbeiten.

Sie können valkey-cli auch unter Amazon Linux und TLS/SSL Amazon Linux 2 verwenden. Wenn Ihr Client TLS nicht unterstützt, können Sie den `stunnel` Befehl auf Ihrem Client-Host verwenden, um einen SSL-Tunnel zu den Redis-OSS-Knoten zu erstellen.

Verschlüsselte Verbindung mit Linux

Gehen Sie wie folgt vor, um mit Valkey-CLI eine Verbindung zu einem Valkey- oder Redis-OSS-Cluster herzustellen, für den die Verschlüsselung während der Übertragung auf Amazon Linux 2 oder Amazon Linux aktiviert ist.

1. Laden Sie das Valkey-Cli-Hilfsprogramm herunter und kompilieren Sie es. Dieses Hilfsprogramm ist in der Valkey-Softwaredistribution enthalten.
2. Geben Sie an der Befehlszeile Ihrer EC2 Instanz die entsprechenden Befehle für die von Ihnen verwendete Linux-Version ein.

Amazon Linux 2

Wenn Sie Amazon Linux 2 verwenden, geben Sie Folgendes ein:

```
sudo yum -y install openssl-devel gcc
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make distclean
make valkey-cli BUILD_TLS=yes
sudo install -m 755 src/valkey-cli /usr/local/bin/
```

Amazon Linux

Wenn Sie Amazon Linux verwenden, geben Sie Folgendes ein:

```
sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel clang wget
wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
tar xvzf valkey-8.0.0.tar.gz
cd valkey-8.0.0
make valkey-cli CC=clang BUILD_TLS=yes
sudo install -m 755 src/valkey-cli /usr/local/bin/
```

Unter Amazon Linux müssen Sie möglicherweise auch die folgenden zusätzlichen Schritte ausführen:

```
sudo yum install clang
CC=clang make
sudo make install
```

3. Nachdem Sie das valkey-cli Utility heruntergeladen und installiert haben, wird empfohlen, den optionalen Befehl auszuführen. `make-test`
4. Um eine Verbindung zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung herzustellen, geben Sie diesen Befehl ein:

```
valkey-cli -h Primary or Configuration Endpoint --tls -a 'your-password' -p 6379
```

Note

Wenn Sie redis6 auf Amazon Linux 2023 installieren, können Sie jetzt den folgenden Befehl verwenden: `redis6-cli: valkey-cli`

```
redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
```

Verschlüsselte Verbindung mit Stunnel

Gehen Sie wie folgt vor, um mit Valkey-CLI eine Verbindung zu einem Redis-OSS-Cluster herzustellen, für den die Verschlüsselung während der Übertragung mithilfe von Stunnel aktiviert ist.

1. Verwenden Sie SSH, um eine Verbindung mit Ihrem Client herzustellen und installieren Sie stunnel.

```
sudo yum install stunnel
```

2. Führen Sie den folgenden Befehl aus, um eine Datei `/etc/stunnel/valkey-cli.conf` gleichzeitig zu erstellen und zu bearbeiten, um einem oder mehreren Verbindungsparametern einen Endpunkt ElastiCache für den Redis OSS-Cluster hinzuzufügen. Verwenden Sie dabei die unten angegebene Ausgabe als Vorlage.

```
vi /etc/stunnel/valkey-cli.conf

fips = no
```

```
setuid = root
setgid = root
pid = /var/run/stunnel.pid
debug = 7
delay = yes
options = NO_SSLv2
options = NO_SSLv3
[valkey-cli]
    client = yes
    accept = 127.0.0.1:6379
    connect = primary.ssltest.wif01h.use1.cache.amazonaws.com:6379
[valkey-cli-replica]
    client = yes
    accept = 127.0.0.1:6380
    connect = ssltest-02.ssltest.wif01h.use1.cache.amazonaws.com:6379
```

In diesem Beispiel hat die Konfigurationsdatei zwei Verbindungen: `valkey-cli` und `valkey-cli-replica`. Dabei werden die Parameter folgendermaßen eingestellt:

- Für `client` wird "yes" eingestellt, um anzugeben, dass diese `stunnel`-Instance ein Client ist.
- `accept` wird die Client-IP zugewiesen. In diesem Beispiel ist der primäre Wert auf den Redis OSS-Standard 127.0.0.1 auf Port 6379 gesetzt. Das Replikat muss einen anderen Port aufrufen und auf „6380“ festgelegt sein. Sie können die temporären Ports 1024-65535 verwenden. Weitere Informationen finden Sie unter [Temporäre Ports](#) im Benutzerhandbuch zu Amazon VPC.
- `connect` ist auf den Redis OSS-Serverendpunkt eingestellt. Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).

3. Starten `stunnel`.

```
sudo stunnel /etc/stunnel/valkey-cli.conf
```

Verwenden Sie den Befehl `netstat`, um zu bestätigen, dass der Tunnel gestartet wurde.

```
sudo netstat -tulnp | grep -i stunnel

tcp        0      0 127.0.0.1:6379        0.0.0.0:*              LISTEN
3189/stunnel
tcp        0      0 127.0.0.1:6380        0.0.0.0:*              LISTEN
3189/stunnel
```


4. Stellen Sie über den lokalen Endpunkt des Tunnels eine Verbindung zum verschlüsselten Redis OSS-Knoten her.
 - Wenn bei ElastiCache der Erstellung des Redis OSS-Clusters kein AUTH-Passwort verwendet wurde, verwendet dieses Beispiel die Valkey-CLI, um eine Verbindung zum ElastiCache for Redis OSS-Server unter Verwendung des vollständigen Pfads für Valkey-CLI unter Amazon Linux herzustellen:

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379
```

Wenn das AUTH-Passwort bei der Erstellung des Redis OSS-Clusters verwendet wurde, verwendet dieses Beispiel valkey-cli, um eine Verbindung zum Redis OSS-Server herzustellen, wobei der vollständige Pfad für valkey-cli auf Amazon Linux verwendet wird:

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

ODER

- Ändern Sie das Verzeichnis auf redis-7.2.5 und gehen Sie wie folgt vor:

Wenn bei ElastiCache der Erstellung des Redis OSS-Clusters kein AUTH-Passwort verwendet wurde, verwendet dieses Beispiel die Valkey-CLI, um eine Verbindung zum ElastiCache for Redis OSS-Server unter Verwendung des vollständigen Pfads für Valkey-CLI unter Amazon Linux herzustellen:

```
src/valkey-cli -h localhost -p 6379
```

Wenn das AUTH-Passwort bei der Erstellung des Redis OSS-Clusters verwendet wurde, verwendet dieses Beispiel valkey-cli, um eine Verbindung zum Valkey- oder Redis OSS-Server herzustellen, wobei der vollständige Pfad für valkey-cli auf Amazon Linux verwendet wird:

```
src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

In diesem Beispiel wird Telnet verwendet, um eine Verbindung zum Valkey Redis OSS-Server herzustellen.

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

5. Um SSL-Tunnel zu stoppen und zu schließen, verwenden Sie `kill` für den `stunnel`-Prozess.

```
sudo kill stunnel
```

Aktivierung der Verschlüsselung während der Übertragung auf einem knotenbasierten Redis-OSS-Cluster mithilfe von Python

Die folgende Anleitung zeigt, wie die Verschlüsselung bei der Übertragung auf einem Redis OSS 7.0-Cluster aktiviert wird, der ursprünglich mit deaktivierter Verschlüsselung während der Übertragung erstellt wurde. TCP- und TLS-Clients kommunizieren während dieses Vorgangs ohne Ausfallzeiten weiter mit dem Cluster.

Boto3 ruft die benötigten Anmeldeinformationen

(`aws_access_key_id`, `aws_secret_access_key`, und `aws_session_token`) aus den Umgebungsvariablen ab. Diese Anmeldeinformationen werden vorab in dasselbe Bash-Terminal eingefügt, in dem wir `python3` ausführen, um den in diesem Handbuch gezeigten Python-Code zu verarbeiten. Der Code im folgenden Beispiel wurde von einer EC2 Instance verarbeitet, die in derselben VPC gestartet wurde, mit der der ElastiCache Redis OSS-Cluster darin erstellt wird.

Note

- In den folgenden Beispielen werden das boto3-SDK für ElastiCache Verwaltungsvorgänge (Cluster- oder Benutzererstellung) und `redis-py-cluster redis-py/` für die Datenverarbeitung verwendet.
- Sie müssen mindestens boto3-Version (`=~`) 1.26.39 verwenden, um die Online-TLS-Migration mit der Cluster-Änderungs-API verwenden zu können.

- ElastiCache unterstützt die Online-TLS-Migration nur für Cluster mit Valkey Version 7.2 und höher oder Redis OSS Version 7.0 oder höher. Wenn Sie also einen Cluster haben, auf dem eine Redis OSS-Version vor 7.0 ausgeführt wird, müssen Sie die Redis OSS-Version Ihres Clusters aktualisieren. Weitere Informationen zu Versionsunterschieden finden Sie unter [Wesentliche Unterschiede zwischen dem Verhalten und der Kompatibilität der Engine-Versionen mit Redis OSS](#).

Themen

- [Definieren Sie die Zeichenkettenkonstanten, die den ElastiCache Valkey- oder Redis OSS-Cluster starten](#)
- [Definieren der Klassen für die Clusterkonfiguration](#)
- [Definieren einer Klasse, die den eigentlichen Cluster darstellt](#)
- [\(Optional\) Erstellen Sie eine Wrapper-Klasse, um die Client-Verbindung zum Valkey- oder Redis-OSS-Cluster zu demonstrieren](#)
- [Erstellen der Hauptfunktion, die den Änderungsprozess der Konfiguration der Verschlüsselung während der Übertragung demonstriert](#)

Definieren Sie die Zeichenkettenkonstanten, die den ElastiCache Valkey- oder Redis OSS-Cluster starten

Definieren wir zunächst einige einfache Python-String-Konstanten, die die Namen der AWS Entitäten enthalten, die zur Erstellung des ElastiCache Clusters erforderlich sind: `security-group`, `elasticache-subnet-group`, und `default-parameter-group`. Alle diese AWS Entitäten müssen im Voraus in Ihrem AWS Konto in der Region erstellt werden, die Sie verwenden möchten.

```
#Constants definitions
SECURITY_GROUP = "sg-0492aa0a29c558427"
CLUSTER_DESCRIPTION = "This cluster has been launched as part of the online TLS
migration user guide"
EC_SUBNET_GROUP = "client-testing"
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED = "default.redis7.cluster.on"
```

Definieren der Klassen für die Clusterkonfiguration

Lassen Sie uns nun einige einfache Python-Klassen definieren, die eine Konfiguration eines Clusters darstellen und Metadaten über den Cluster wie die Valkey- oder Redis-OSS-Version, den Instanztyp und ob die In-Transit-Verschlüsselung (TLS) aktiviert oder deaktiviert ist, enthalten.

```
#Class definitions

class Config:
    def __init__(
        self,
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
        multi_az: bool = True,
        TLS: bool = True,
        name: str = None,
    ):
        self.instance_type = instance_type
        self.version = version
        self.multi_az = multi_az
        self.TLS = TLS
        self.name = name or f"tls-test"

    def create_base_launch_request(self):
        return {
            "ReplicationGroupId": self.name,
            "TransitEncryptionEnabled": self.TLS,
            "MultiAZEnabled": self.multi_az,
            "CacheNodeType": self.instance_type,
            "Engine": "redis",
            "EngineVersion": self.version,
            "CacheSubnetGroupName": EC_SUBNET_GROUP ,
            "CacheParameterGroupName":
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED ,
            "ReplicationGroupDescription": CLUSTER_DESCRIPTION,
            "SecurityGroupIds": [SECURITY_GROUP],
        }

class ConfigCME(Config):
    def __init__(
        self,
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
```

```

        multi_az: bool = True,
        TLS: bool = True,
        name: str = None,
        num_shards: int = 2,
        num_replicas_per_shard: int = 1,
    ):
        super().__init__(instance_type, version, multi_az, TLS, name)
        self.num_shards = num_shards
        self.num_replicas_per_shard = num_replicas_per_shard

    def create_launch_request(self) -> dict:
        launch_request = self.create_base_launch_request()
        launch_request["NumNodeGroups"] = self.num_shards
        launch_request["ReplicasPerNodeGroup"] = self.num_replicas_per_shard
        return launch_request

```

Definieren einer Klasse, die den eigentlichen Cluster darstellt

Lassen Sie uns nun einige einfache Python-Klassen definieren, die den ElastiCache Valkey- oder Redis-OSS-Cluster selbst darstellen. Diese Klasse wird über ein Client-Feld verfügen, das einen Boto3-Client für ElastiCache Verwaltungsvorgänge wie das Erstellen des Clusters und das Abfragen der API enthält. ElastiCache

```

import botocore.config
import boto3

# Create boto3 client
def init_client(region: str = "us-east-1"):
    config = botocore.config.Config(retries={"max_attempts": 10, "mode": "standard"})
    init_request = dict()
    init_request["config"] = config
    init_request["service_name"] = "elasticache"
    init_request["region_name"] = region
    return boto3.client(**init_request)

class ElastiCacheClusterBase:
    def __init__(self, name: str):
        self.name = name
        self.elasticache_client = init_client()

    def get_first_replication_group(self):
        return self.elasticache_client.describe_replication_groups(

```

```
        ReplicationGroupId=self.name
    )["ReplicationGroups"][0]

def get_status(self) -> str:
    return self.get_first_replication_group()["Status"]

def get_transit_encryption_enabled(self) -> bool:
    return self.get_first_replication_group()["TransitEncryptionEnabled"]

def is_available(self) -> bool:
    return self.get_status() == "available"

def is_modifying(self) -> bool:
    return self.get_status() == "modifying"

def wait_for_available(self):
    while True:
        if self.is_available():
            break
        else:
            time.sleep(5)

def wait_for_modifying(self):
    while True:
        if self.is_modifying():
            break
        else:
            time.sleep(5)

def delete_cluster(self) -> bool:
    self.elasticache_client.delete_replication_group(
        ReplicationGroupId=self.name, RetainPrimaryCluster=False
    )

def modify_transit_encryption_mode(self, new_transit_encryption_mode: str):
    # generate api call to migrate the cluster to TLS preferred or to TLS required
    self.elasticache_client.modify_replication_group(
        ReplicationGroupId=self.name,
        TransitEncryptionMode=new_transit_encryption_mode,
        TransitEncryptionEnabled=True,
        ApplyImmediately=True,
    )
    self.wait_for_modifying()
```

```

class ElastiCacheClusterCME(ElastiCacheClusterBase):
    def __init__(self, name: str):
        super().__init__(name)

    @classmethod
    def launch(cls, config: ConfigCME = None) -> ElastiCacheClusterCME:
        config = config or ConfigCME()
        print(config)
        new_cluster = ElastiCacheClusterCME(config.name)
        launch_request = config.create_launch_request()
        new_cluster.elasticache_client.create_replication_group(**launch_request)
        new_cluster.wait_for_available()
        return new_cluster

    def get_configuration_endpoint(self) -> str:
        return self.get_first_replication_group()["ConfigurationEndpoint"]["Address"]

#Since the code can throw exceptions, we define this class to make the code more
#readable and
#so we won't forget to delete the cluster
class ElastiCacheCMEManager:
    def __init__(self, config: ConfigCME = None):
        self.config = config or ConfigCME()

    def __enter__(self) -> ElastiCacheClusterCME:
        self.cluster = ElastiCacheClusterCME.launch(self.config)
        return self.cluster

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.cluster.delete_cluster()

```

(Optional) Erstellen Sie eine Wrapper-Klasse, um die Client-Verbindung zum Valkey- oder Redis-OSS-Cluster zu demonstrieren

Jetzt erstellen wir eine Wrapper-Klasse für den `redis-py-cluster`-Client. Diese Wrapper-Klasse unterstützt das Vorbefüllen des Clusters mit einigen Schlüsseln und das anschließende Ausführen zufälliger wiederholter `get`-Befehle.

Note

Dies ist ein optionaler Schritt, der jedoch den Code der Hauptfunktion vereinfacht, die in einem späteren Schritt erstellt wird.

```
import redis
import random
from time import perf_counter_ns, time

class DowntimeTestClient:
    def __init__(self, client):
        self.client = client

        # num of keys prefilled
        self.prefilled = 0
        # percent of get above prefilled
        self.percent_get_above_prefilled = 10 # nil result expected when get hit above
prefilled
        # total downtime in nano seconds
        self.downtime_ns = 0
        # num of success and fail operations
        self.success_ops = 0
        self.fail_ops = 0
        self.connection_errors = 0
        self.timeout_errors = 0

    def replace_client(self, client):
        self.client = client

    def prefill_data(self, timelimit_sec=60):
        end_time = time() + timelimit_sec
        while time() < end_time:
            self.client.set(self.prefilled, self.prefilled)
            self.prefilled += 1

    # unsuccessful operations throw exceptions
    def _exec(self, func):
        try:
            start_ns = perf_counter_ns()
            func()
            self.success_ops += 1
            elapsed_ms = (perf_counter_ns() - start_ns) // 10 ** 6
            # upon succesful execution of func
            # reset random_key to None so that the next command
            # will use a new random key
            self.random_key = None
```



```

except Exception as e:
    elapsed_ns = perf_counter_ns() - start_ns
    self.downtime_ns += elapsed_ns
    # in case of failure- increment the relevant counters so that we will keep
track
    # of how many connection issues we had while trying to communicate with
    # the cluster.
    self.fail_ops += 1
    if e.__class__ is redis.exceptions.ConnectionError:
        self.connection_errors += 1
    if e.__class__ is redis.exceptions.TimeoutError:
        self.timeout_errors += 1

def _repeat_exec(self, func, seconds):
    end_time = time() + seconds
    while time() < end_time:
        self._exec(func)

def _new_random_key_if_needed(self, percent_above_prefilled):
    if self.random_key is None:
        max = int((self.prefilled * (100 + percent_above_prefilled)) / 100)
        return random.randint(0, max)
    return self.random_key

def _random_get(self):
    key = self._new_random_key_if_needed(self.percent_get_above_prefilled)
    result = self.client.get(key)
    # we know the key was set for sure only in the case key < self.prefilled
    if key < self.prefilled:
        assert result.decode("UTF-8") == str(key)

def repeat_get(self, seconds=60):
    self._repeat_exec(self._random_get, seconds)

def get_downtime_ms(self) -> int:
    return self.downtime_ns // 10 ** 6

def do_get_until(self, cond_check):
    while not cond_check():
        self.repeat_get()
    # do one more get cycle once condition is met

```

```
self.repeat_get()
```

Erstellen der Hauptfunktion, die den Änderungsprozess der Konfiguration der Verschlüsselung während der Übertragung demonstriert

Definieren wir nun die Hauptfunktion wie folgt:

1. Erstellen Sie den Cluster mit dem Boto3-Client. ElastiCache
2. Initialisieren Sie den `redis-py-cluster`-Client, der eine Verbindung mit dem Cluster mit einer guten TCP-Verbindung ohne TLS herstellt.
3. Der `redis-py-cluster`-Client füllt den Cluster vorab mit einigen Daten.
4. Der boto3-Client löst die TLS-Migration von „no-TLS“ zu „TLS Preferred“ aus.
5. Während der Cluster zu TLS Preferred migriert wird, sendet der `redis-py-cluster-TCP`-Client wiederholte `get`-Operationen an den Cluster, bis die Migration abgeschlossen ist.
6. Nachdem die Migration zu TLS Preferred abgeschlossen ist, überprüfen wir, dass der Cluster die Verschlüsselung während der Übertragung unterstützt. Anschließend erstellen wir einen `redis-py-cluster`-Client, der mit TLS eine Verbindung mit dem Cluster herstellt.
7. Wir senden einige `get`-Befehle mit dem neuen TLS-Client und dem alten TCP-Client.
8. Der boto3-Client löst die TLS-Migration von „TLS Preferred“ zu „TLS required“ aus.
9. Während der Migration des Clusters auf TLS erforderlich, sendet der `redis-py-cluster TLS`-Client wiederholte `get` Operationen an den Cluster, bis die Migration abgeschlossen ist.

```
import redis

def init_cluster_client(
    cluster: ElastiCacheClusterCME, prefill_data: bool, TLS: bool = True) ->
DowntimeTestClient:
    # we must use for the host name the cluster configuration endpoint.
    redis_client = redis.RedisCluster(
        host=cluster.get_configuration_endpoint(), ssl=TLS, socket_timeout=0.25,
        socket_connect_timeout=0.1
    )
    test_client = DowntimeTestClient(redis_client)
    if prefill_data:
        test_client.prefill_data()
    return test_client

if __name__ == '__main__':
```

```
config = ConfigCME(TLS=False, instance_type="cache.m5.large")

with ElastiCacheCMEManager(config) as cluster:
    # create a client that will connect to the cluster with clear tcp connection
    test_client_tcp = init_cluster_client(cluster, prefill_data=True, TLS=False)

    # migrate the cluster to TLS Preferred
    cluster.modify_transit_encryption_mode(new_transit_encryption_mode="preferred")

    # do repeated get commands until the cluster finishes the migration to TLS
    Preferred
    test_client_tcp.do_get_until(cluster.is_available)

    # verify that in transit encryption is enabled so that clients will be able to
    connect to the cluster with TLS
    assert cluster.get_transit_encryption_enabled() == True

    # create a client that will connect to the cluster with TLS connection.
    # we must first make sure that the cluster indeed supports TLS
    test_client_tls = init_cluster_client(cluster, prefill_data=True, TLS=True)

    # by doing get commands with the tcp client for 60 more seconds
    # we can verify that the existing tcp connection to the cluster still works
    test_client_tcp.repeat_get(seconds=60)

    # do get commands with the new TLS client for 60 more seconds
    test_client_tcp.repeat_get(seconds=60)

    # migrate the cluster to TLS required
    cluster.modify_transit_encryption_mode(new_transit_encryption_mode="required")

    # from this point the tcp clients will be disconnected and we must not use them
    anymore.
    # do get commands with the TLS client until the cluster finishes migration to
    TLS required mode.
    test_client_tls.do_get_until(cluster.is_available)
```

Bewährte Methoden für die Aktivierung der Verschlüsselung während der Übertragung

Vor dem Aktivieren der Verschlüsselung während der Übertragung: sicherstellen, dass die DNS-Einträge ordnungsgemäß verarbeitet werden

Note

Während dieses Prozesses ändern und löschen wir alte Endpunkte. Eine falsche Verwendung der Endpunkte kann dazu führen, dass der Valkey- oder Redis-OSS-Client alte und gelöschte Endpunkte verwendet, sodass er keine Verbindung zum Cluster herstellen kann.

Während der Migration des Clusters von ohne TLS auf TLS Preferred wird der alte DNS-Eintrag für den Endpunkt der Clusterkonfiguration beibehalten, und die neuen DNS-Einträge für den Clusterkonfigurationsendpunkt werden in einem anderen Format generiert. TLS-fähige Cluster verwenden ein anderes Format von DNS-Einträgen als TLS-deaktivierte Cluster. ElastiCache behält beide DNS-Einträge bei, wenn ein Cluster konfiguriert ist, `encryption mode: Preferred` sodass Anwendungen und andere Valkey- oder Redis OSS-Clients zwischen ihnen wechseln können. Die folgenden Änderungen in den DNS-Einträgen finden während des TLS-Migrationsprozesses statt:

Beschreibung der Änderungen an den DNS-Einträgen, die bei der Aktivierung der Verschlüsselung während der Übertragung vorgenommen werden

Für CME-Cluster

Wenn ein Cluster auf „Übertragungs-Verschlüsselungsmodus: bevorzugt“ eingestellt ist:

- Der ursprüngliche Endpunkt der Clusterkonfiguration für einen No-TLS-Cluster bleibt aktiv. Es wird keine Ausfallzeit geben, wenn der Cluster vom TLS-Verschlüsselungsmodus „none“ auf „preferred“ umkonfiguriert wird.
- Neue TLS Valkey- oder Redis OSS-Endpunkte werden generiert, wenn der Cluster auf den TLS-Preferred-Modus gesetzt wird. Diese neuen Endpunkte werden genauso aufgelöst IPs wie die alten Endpunkte (ohne TLS).
- Der neue TLS Valkey- oder Redis OSS-Konfigurationsendpunkt wird in der ElastiCache Konsole und in der Antwort auf die API verfügbar gemacht. `describe-replication-group`

Wenn ein Cluster auf „Übertragungs-Verschlüsselungsmodus: erforderlich“ eingestellt ist:

- Alte Endpunkte, die nicht TLS-fähig sind, werden gelöscht. Es wird keine Ausfallzeiten der TLS-Cluster-Endpunkte geben.
- Sie können einen neuen `cluster-configuration-endpoint` von der ElastiCache Konsole oder von der `describe-replication-group` API abrufen.

Für CMD-Cluster mit aktiviertem automatischem Failover oder deaktiviertem automatischen Failover

Wenn die Replikationsgruppe auf „Übertragungs-Verschlüsselungsmodus: bevorzugt“ eingestellt ist:

- Der ursprüngliche primäre Endpunkt und der Reader-Endpunkt für nicht TLS-fähige Cluster bleiben aktiv.
- Neue primäre und Reader-Endpunkte mit TLS werden generiert, wenn der Cluster auf den Modus „TLS Preferred“ eingestellt ist. Diese neuen Endpunkte werden auf dieselben IPs wie die alten (ohne TLS) aufgelöst.
- Der neue primäre Endpunkt und der Reader-Endpunkt werden in der ElastiCache Konsole und in der Antwort auf die `describe-replication-group` API angezeigt.

Wenn die Replikationsgruppe auf „Übertragungs-Verschlüsselungsmodus: erforderlich“ eingestellt ist:

- Alte primäre und Reader-Endpunkte ohne TLS werden gelöscht. Es wird keine Ausfallzeiten der TLS-Cluster-Endpunkte geben.
- Sie können neue primäre Endpunkte und Leser-Endpunkte von der ElastiCache Konsole oder von der `describe-replication-group` API abrufen.

Die empfohlene Verwendung der DNS-Einträge

Für CME-Cluster

- Verwenden Sie den Endpunkt der Clusterkonfiguration anstelle von DNS-Einträgen pro Knoten in Ihrem Anwendungscode. Die direkte Verwendung von DNS-Namen pro Knoten wird nicht empfohlen, da sie sich während der Migration ändern und der Anwendungscode die Verbindung zum Cluster unterbricht.
- Codieren Sie einen Cluster-Konfigurationsendpunkt in Ihrer Anwendung nicht fest, da er sich während dieses Vorgangs ändern wird.
- Es ist eine schlechte Praxis, den Cluster-Konfigurationsendpunkt in Ihrer Anwendung fest zu codieren, da er während dieses Vorgangs geändert werden kann. Nachdem die Verschlüsselung

während der Übertragung abgeschlossen ist, fragen Sie den Endpunkt der Cluster-Konfiguration mit der `describe-replication-group` API ab (wie oben dargestellt (fett gedruckt)) und verwenden Sie ab diesem Zeitpunkt den DNS, den Sie als Antwort erhalten.

Für CMD-Cluster mit aktiviertem automatischen Failover

- Verwenden Sie in Ihrem Anwendungscode den primären Endpunkt und den Reader-Endpoint anstelle der DNS-Namen pro Knoten, da die alten DNS-Namen pro Knoten gelöscht und neue generiert werden, wenn der Cluster von „no-TLS“ zu „TLS Preferred“ migriert wird. Es wird nicht empfohlen, DNS-Namen pro Knoten direkt zu verwenden, da Sie Ihrem Cluster später möglicherweise Replikate hinzufügen. Wenn Automatic Failover aktiviert ist, werden außerdem die Rollen des primären Clusters und der Replikate automatisch vom ElastiCache Dienst geändert. Es wird empfohlen, den primären Endpunkt und den Leser-Endpoint zu verwenden, damit Sie diese Änderungen verfolgen können. Schließlich hilft Ihnen die Verwendung des Reader-Endpunkts dabei, Ihre Lesevorgänge aus den Replikaten gleichmäßig auf die Replikate im Cluster zu verteilen.
- Es wird nicht empfohlen, den primären Endpunkt und den Reader-Endpoint in Ihrer Anwendung fest zu kodieren, da diese während des TLS-Migrationsprozesses geändert werden können. Nachdem die Umstellung auf TLS-Preferred abgeschlossen ist, fragen Sie den primären Endpunkt und den Endpunkt des Lesegeräts mit der `describe-replication-group` API ab und verwenden Sie ab diesem Zeitpunkt den DNS, den Sie als Antwort erhalten. Auf diese Weise können Sie Änderungen an Endpunkten dynamisch verfolgen.

Für CMD-Cluster mit deaktiviertem automatischen Failover

- Verwenden Sie den primären Endpunkt und den Reader-Endpoint anstelle der DNS-Namen pro Knoten in Ihrem Anwendungscode. Wenn Automatic Failover deaktiviert ist, werden Skalierung, Patching, Failover und andere Verfahren, die automatisch vom ElastiCache Dienst verwaltet werden, wenn Automatic Failover aktiviert ist, stattdessen von Ihnen durchgeführt. Das erleichtert es Ihnen, den Überblick über die verschiedenen Endpunkte zu behalten. Da bei der Migration des Clusters von „no-TLS“ zu „TLS Preferred“ die alten DNS-Namen pro Knoten gelöscht und neue generiert werden, sollten Sie die DNS-Namen pro Knoten nicht direkt verwenden. Dies ist erforderlich, damit sich die Clients während der TLS-Migration mit dem Cluster verbinden können. Außerdem profitieren Sie von einer gleichmäßigen Verteilung der Lesevorgänge zwischen den Replikaten, wenn Sie den Reader-Endpoint verwenden, und behalten den Überblick über die DNS-Einträge, wenn Sie dem Cluster Replikate hinzufügen oder löschen.

- Eine feste Kodierung des Endpunkts der Clusterkonfiguration in Ihrer Anwendung wird nicht empfohlen, da er während der TLS-Migration geändert werden kann.

Bei der Verschlüsselung während der Übertragung: darauf achten, wann der Migrationsprozess abgeschlossen ist

Die Änderung des Verschlüsselungsmodus während der Übertragung erfolgt nicht sofort und kann einige Zeit in Anspruch nehmen. Dies gilt insbesondere für große Cluster. Erst wenn der Cluster die Migration zu „TLS Preferred“ abgeschlossen hat, kann er sowohl TCP- als auch TLS-Verbindungen akzeptieren und bereitstellen. Daher sollten Sie keine Clients erstellen, die versuchen, TLS-Verbindungen mit dem Cluster herzustellen, bis die Verschlüsselung während der Übertragung abgeschlossen ist.

Es gibt mehrere Möglichkeiten, sich benachrichtigen zu lassen, wenn die Verschlüsselung während der Übertragung erfolgreich abgeschlossen wurde oder fehlschlägt (im obigen Codebeispiel nicht dargestellt):

- Verwenden des SNS-Services, um benachrichtigt zu werden, wenn die Verschlüsselung abgeschlossen ist
- Verwenden der `describe-events`-API, die ein Ereignis ausgibt, wenn die Verschlüsselung abgeschlossen ist
- In der ElastiCache Konsole wird eine Meldung angezeigt, dass die Verschlüsselung abgeschlossen ist

Sie können in Ihrer Anwendung auch Logik implementieren, um festzustellen, ob die Verschlüsselung abgeschlossen ist. Im obigen Beispiel haben wir mehrere Möglichkeiten kennengelernt, wie wir sicherstellen können, dass der Cluster die Migration abschließt:

- Warten, bis der Migrationsprozess beginnt (der Clusterstatus wechselt zu „Wird geändert“) und warten, bis die Änderung abgeschlossen ist (der Clusterstatus wechselt wieder zu „Verfügbar“)
- Bestätigen Sie, dass für den Cluster `transit_encryption_enabled` auf „Wahr“ festgelegt ist, indem Sie die `describe-replication-group`-API abfragen.

Nach dem Aktivieren der Verschlüsselung während der Übertragung: sicherstellen, dass die von Ihnen verwendeten Clients ordnungsgemäß konfiguriert sind

Während sich der Cluster im Modus „TLS Preferred“ befindet, sollte Ihre Anwendung TLS-Verbindungen mit dem Cluster herstellen und nur diese Verbindungen verwenden. Auf diese Weise kommt es bei Ihrer Anwendung nicht zu Ausfallzeiten, wenn die Verschlüsselung während der Übertragung aktiviert wird. Mit dem Befehl `info` im Abschnitt `SSL` können Sie sicherstellen, dass keine klareren TCP-Verbindungen zur Valkey- oder Redis-OSS-Engine bestehen.

```
# SSL
ssl_enabled:yes
ssl_current_certificate_not_before_date:Mar 20 23:27:07 2017 GMT
ssl_current_certificate_not_after_date:Feb 24 23:27:07 2117 GMT
ssl_current_certificate_serial:D8C7DEA91E684163
tls_mode_connected_tcp_clients:0    (should be zero)
tls_mode_connected_tls_clients:100
```

Mit Openssl (Memcached) wird eine Verbindung zu Knoten hergestellt, für die Verschlüsselung während der Übertragung aktiviert ist

Um auf Daten von Memcached-Knoten zuzugreifen, ElastiCache für die Verschlüsselung bei der Übertragung aktiviert ist, müssen Sie Clients verwenden, die mit Secure Socket Layer (SSL) arbeiten. Sie können Openssl `s_client` auch auf Amazon Linux und Amazon Linux 2 verwenden.

Verwenden Sie Openssl `s_client`, um sich mit einem Memcached-Cluster zu verbinden, der mit Verschlüsselung während der Übertragung auf Amazon Linux 2 oder Amazon Linux aktiviert ist, wie folgt:

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

Erstellen eines TLS-Memcached-Clients mit Java

Um einen Client im TLS-Modus zu erstellen, gehen Sie wie folgt vor, um den Client mit dem entsprechenden zu initialisieren: `SSLContext`

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.MemcachedClient;
```



```

public class TLSDemo {
    public static void main(String[] args) throws Exception {
        ConnectionFactoryBuilder connectionFactoryBuilder = new
ConnectionFactoryBuilder();
        // Build SSLContext
        TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init((KeyStore) null);
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, tmf.getTrustManagers(), null);
        // Create the client in TLS mode
        connectionFactoryBuilder.setSSLContext(sslContext);
        MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
AddrUtil.getAddresses("mycluster.fnjyzo.cfg.use1.cache.amazonaws.com:11211"));

        // Store a data item for an hour.
        client.set("theKey", 3600, "This is the data value");
    }
}

```

Erstellen eines TLS-Memcached-Clients mit PHP

Gehen Sie wie folgt vor, um einen Client im TLS-Modus zu erstellen, um den Client mit dem entsprechenden zu initialisieren: SSLContext

```

<?php

/**
 * Sample PHP code to show how to create a TLS Memcached client. In this example we
 * will use the Amazon ElastiCache Auto Discovery feature, but TLS can also be
 * used with a Static mode client.
 * See Using the ElastiCache Cluster Client for PHP (https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/AutoDiscovery.Using.ModifyApp.PHP.html) for more
 * information
 * about Auto Discovery and persistent-id.
 */

/* Configuration endpoint to use to initialize memcached client.
 * this is only an example */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the cluster.
 * This is only an example */

```

```
$server_port = 11211;

/* Initialize a persistent Memcached client and configure it with the Dynamic client
mode */
$tls_client = new Memcached('persistent-id');
$tls_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);

/* Add the memcached's cluster server/s */
$tls_client->addServer($server_endpoint, $server_port);

/* Configure the client to use TLS */
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastError_message(), "\n";
    exit(1);
}

/* Set your TLS context configurations values.
 * See MemcachedTLSTextContextConfig in memcached-api.php for all configurations */
$tls_config = new MemcachedTLSTextContextConfig();
$tls_config->hostname = '*.mycluster.fnjyzo.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;

/* Use the created TLS context configuration object to create OpenSSL's SSL_CTX and set
it to your client.
 * Note: These TLS context configurations will be applied to all the servers connected
to this client. */
$tls_client->createAndSetTLSContext((array)$tls_config);

/* test the TLS connection with set-get scenario: */

/* store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
if($tls_client->set('key', 'value', 60)) {
    print "Successfully stored key\n";
} else {
    echo "Failed to set key: ", $tls_client->getLastError_message(), "\n";
    exit(1);
}

/* retrieve the key */
if ($tls_client->get('key') === 'value') {
    print "Successfully retrieved key\n";
}
```

```
} else {  
    echo "Failed to get key: ", $tls_client->getLastError(), "\n";  
    exit(1);  
}
```

Weitere Informationen zur Verwendung des PHP-Client finden Sie unter [Installation des ElastiCache Cluster-Clients für PHP](#).

Verschlüsselung im Ruhezustand in ElastiCache

Um Ihre Daten zu schützen, bieten Amazon ElastiCache und Amazon S3 verschiedene Möglichkeiten, den Zugriff auf Daten in Ihrem Cache einzuschränken. Weitere Informationen erhalten Sie unter [Amazon VPCs und ElastiCache Sicherheit](#) und [Identity and Access Management für Amazon ElastiCache](#).

ElastiCache Die Verschlüsselung im Ruhezustand ist eine Funktion zur Erhöhung der Datensicherheit durch Verschlüsselung von Daten auf der Festplatte. In einem Serverless-Cache ist sie immer aktiviert. Wenn die Funktion aktiviert ist, werden folgenden Aspekte verschlüsselt:

- Festplatte bei Synchronisierung, Backup und Austausch
- In Amazon S3 gespeicherte Backups

Daten, die auf SSDs (Solid-State-Laufwerken) in Clustern mit aktivierter Datenklassifizierung gespeichert sind, sind immer verschlüsselt.

ElastiCache bietet standardmäßige (vom Service verwaltete) Verschlüsselung im Ruhezustand sowie die Möglichkeit, Ihre eigenen symmetrischen, vom Kunden verwalteten AWS KMS-Schlüssel im [AWS Key Management Service](#) (KMS) zu verwenden. Wählen Sie bei der Sicherung des Caches unter Verschlüsselungsoptionen aus, ob Sie den Standardverschlüsselungsschlüssel oder einen vom Kunden verwalteten Schlüssel verwenden möchten. Weitere Informationen finden Sie unter [Aktivieren der Verschlüsselung im Ruhezustand](#).

Important

Die Aktivierung der At-Rest-Verschlüsselung auf einem vorhandenen knotenbasierten Valkey- oder Redis-OSS-Cluster beinhaltet das Löschen Ihrer vorhandenen Replikationsgruppe, nachdem die Sicherung und Wiederherstellung für die Replikationsgruppe ausgeführt wurde.

Die Verschlüsselung im Ruhezustand kann nur für einen Cache aktiviert werden, wenn dieser erstellt wird. Da zum Verschlüsseln und Entschlüsseln der Daten ein gewisses Maß an Verarbeitung erforderlich ist, kann die Aktivierung der Verschlüsselung der Daten im Ruhezustand bei diesen Vorgängen zu Leistungseinbußen führen. Sie sollten Ihre Daten mit und ohne Verschlüsselung im Ruhezustand vergleichen, um die Auswirkungen auf die Leistung für Ihre Anwendungsfälle zu ermitteln.

Themen

- [Bedingungen für die Verschlüsselung im Ruhezustand](#)
- [Verwendung von vom Kunden verwalteten Schlüsseln von KMS AWS](#)
- [Aktivieren der Verschlüsselung im Ruhezustand](#)
- [Weitere Informationen finden Sie unter:](#)

Bedingungen für die Verschlüsselung im Ruhezustand

Bei der Planung der Implementierung ElastiCache der Verschlüsselung im Ruhezustand sollten Sie die folgenden Einschränkungen bei der ElastiCache Verschlüsselung im Ruhezustand berücksichtigen:

- Die Verschlüsselung im Ruhezustand wird für Replikationsgruppen unterstützt, auf denen Valkey 7.2 und höher ausgeführt wird, sowie auf Redis OSS-Versionen (3.2.6 ist für EOL geplant, siehe Zeitplan für das [Ende des Lebenszyklus der Redis OSS-Versionen](#)), 4.0.10 oder höher.
- Die Verschlüsselung der Daten im Ruhezustand wird nur für Replikationsgruppen unterstützt, die in einer Amazon VPC. ausgeführt werden.
- Die Verschlüsselung im Ruhezustand wird nur für Replikationsgruppen unterstützt, die die folgenden Knotentypen ausführen.
 - R7g, R6gd, R6g, R5, R4, R3
 - M7 g, M6 g, M5, M4, M3
 - T4g, T3, T2
 - C7gn

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Die Verschlüsselung im Ruhezustand wird durch die explizite Festlegung des `AtRestEncryptionEnabled`-Parameters auf `true` aktiviert.
- Sie können die Verschlüsselung im Ruhezustand nur für eine Replikationsgruppe aktivieren, wenn Sie die Replikationsgruppe erstellen. Sie können die Verschlüsselung während der Übertragung nicht aktivieren und deaktivieren, indem Sie eine Replikationsgruppe ändern. Informationen zur Implementierung der Verschlüsselung im Ruhezustand bei vorhandenen Replikationsgruppen finden Sie unter [Aktivieren der Verschlüsselung im Ruhezustand](#).
- Wenn ein Cluster einen Knotentyp aus der R6gd-Familie verwendet, werden auf SSDs gespeicherte Daten unabhängig davon verschlüsselt, ob die Verschlüsselung im Ruhezustand aktiviert ist oder nicht.

- Die Option, vom Kunden verwaltete Schlüssel für die Verschlüsselung im Ruhezustand zu verwenden, ist in den Regionen AWS GovCloud (us-gov-east-1 und -1) nicht verfügbar. us-gov-west
- Wenn ein Cluster einen Knotentyp aus der R6GD-Familie verwendet, werden die auf der SSD gespeicherten Daten mit dem ausgewählten, vom Kunden verwalteten AWS KMS-Schlüssel (oder der vom Service verwalteten Verschlüsselung in AWS GovCloud Regionen) verschlüsselt.
- Mit Memcached wird die Verschlüsselung im Ruhezustand nur in serverlosen Caches unterstützt.
- Bei Verwendung von Memcached ist die Option, vom Kunden verwaltete Schlüssel für die Verschlüsselung im Ruhezustand zu verwenden, in AWS GovCloud den Regionen (-1 und -1) nicht verfügbar. us-gov-east us-gov-west

Durch die Implementierung der Verschlüsselung der Daten im Ruhezustand kann die Leistung während der Sicherungs- und Knoten-Synchronisierungsvorgänge reduziert sein. Vergleichen Sie die Verschlüsselung der Daten im Ruhezustand mit Ihren eigenen unverschlüsselten Daten, um die Auswirkungen auf die Leistung Ihrer Implementierung zu ermitteln.

Verwendung von vom Kunden verwalteten Schlüsseln von KMS AWS

ElastiCache unterstützt symmetrische, vom Kunden verwaltete AWS KMS-Schlüssel (KMS-Schlüssel) für die Verschlüsselung im Ruhezustand. Kundenverwaltete KMS-Schlüssel sind Verschlüsselungsschlüssel, die Sie in Ihrem AWS Konto erstellen, besitzen und verwalten. Weitere Informationen finden Sie unter [AWS-KMS-Schlüssel](#) im Entwicklerhandbuch zum AWS-Schlüsselverwaltungsdienst. Die Schlüssel müssen in AWS KMS erstellt werden, bevor sie mit ElastiCache verwendet werden können.

Informationen zum Erstellen von AWS KMS-Stammschlüsseln finden Sie unter [Creating Keys](#) im AWS Key Management Service Developer Guide.

ElastiCache ermöglicht Ihnen die Integration mit AWS KMS. Weitere Informationen finden Sie unter [Verwendung von Berechtigungen](#) im Entwicklerhandbuch zum AWS-Schlüsselverwaltungsdienst. Es sind keine Kundenaktionen erforderlich, um die ElastiCache Amazon-Integration mit AWS KMS zu aktivieren.

Der `kms:ViaService` Bedingungsschlüssel beschränkt die Verwendung eines AWS KMS-Schlüssels (KMS-Schlüssel) auf Anfragen von bestimmten AWS Diensten. Um `kms:ViaService` mit zu verwenden ElastiCache, schließen Sie beide `ViaService` Namen

in den Wert des Bedingungsschlüssels ein: `elasticache.AWS_region.amazonaws.com`
`unddax.AWS_region.amazonaws.com`. Weitere Informationen finden Sie unter [kms: ViaService](#).

Sie können [AWS CloudTrail](#) damit die Anfragen verfolgen, an die Amazon in AWS Key Management Service Ihrem Namen ElastiCache sendet. Alle API-Aufrufe, die sich auf vom Kunden verwaltete Schlüssel AWS Key Management Service beziehen, haben entsprechende CloudTrail Protokolle. Sie können sich auch die Zuschüsse ansehen, die ElastiCache generiert werden, wenn Sie den [ListGrantsKMS-API-Aufruf](#) aufrufen.

Ist eine Replikationsgruppe mit einem vom Kunden verwalteten Schlüssel verschlüsselt, werden alle Backups für die Replikationsgruppe wie folgt verschlüsselt:

- Tägliche automatische Backups werden mit dem vom Kunden verwalteten Schlüssel verschlüsselt, der dem Cluster zugeordnet ist.
- Die letzte Sicherung, die beim Löschen der Replikationsgruppe erstellt wird, wird ebenfalls mit dem kundenverwalteten Schlüssel verschlüsselt, der der Replikationsgruppe zugeordnet ist.
- Standardmäßig werden manuell erstellte Sicherungen mit dem der Replikationsgruppe zugeordneten KMS-Schlüssel verschlüsselt. Mit einem anderen kundenverwalteten Schlüssel können Sie dies außer Kraft setzen.
- Als Standardeinstellung wird beim Kopieren einer Sicherung ein vom Kunden verwalteter Schlüssel verwendet, der mit der Quellsicherung verknüpft ist. Mit einem anderen kundenverwalteten Schlüssel können Sie dies außer Kraft setzen.

Note

- Kundenverwaltete Schlüssel können beim Exportieren von Sicherungen in den von Ihnen ausgewählten Amazon S3-Bucket nicht verwendet werden. Alle in Amazon S3 exportierten Sicherungen werden jedoch mit [serverseitiger Verschlüsselung](#) verschlüsselt. Sie können wahlweise die Sicherungsdatei in ein neues S3-Objekt kopieren und mit einem vom Kunden verwalteten KMS-Schlüssel verschlüsseln, die Datei in einen anderen S3-Bucket kopieren, der mit Standardverschlüsselung unter Verwendung eines KMS-Schlüssels eingerichtet ist, oder eine Verschlüsselungsoption in der Datei selbst ändern.
- Außerdem können Sie vom Kunden verwaltete Schlüssel verwenden, um manuell erstellte Backups für Replikationsgruppen zu verschlüsseln, die keine vom Kunden verwalteten Schlüssel für die Verschlüsselung verwenden. Durch diese Option wird die in Amazon

S3 gespeicherte Sicherungsdatei mit einem KMS-Schlüssel verschlüsselt, auch wenn die Daten in der ursprünglichen Replikationsgruppe nicht verschlüsselt sind.

Bei der Wiederherstellung von einer Sicherung können Sie aus verschiedenen Verschlüsselungsoptionen wählen, ähnlich den verfügbaren Verschlüsselungsoptionen beim Erstellen einer neuen Replikationsgruppe.

- Wenn Sie den Schlüssel löschen oder [deaktivieren](#) und für den Schlüssel, den Sie zur Verschlüsselung eines Caches verwendet haben, [Genehmigungen zurückziehen](#), kann der Cache nicht wiederhergestellt werden. Mit anderen Worten, es kann nach einem Hardwarefehler nicht geändert oder wiederhergestellt werden. AWS KMS löscht Root-Schlüssel erst nach einer Wartezeit von mindestens sieben Tagen. Nach dem Löschen des Schlüssels können Sie einen anderen vom Kunden verwalteten Schlüssel verwenden, um eine Sicherungskopie für Archivierungszwecke zu erstellen.
- Bei der automatischen Schlüsselrotation bleiben die Eigenschaften Ihrer AWS KMS-Root-Schlüssel erhalten, sodass die Rotation keine Auswirkungen auf Ihre Fähigkeit hat, auf Ihre Daten zuzugreifen. ElastiCache Verschlüsselte ElastiCache Amazon-Caches unterstützen keine manuelle Schlüsselrotation, bei der ein neuer Root-Schlüssel erstellt und alle Verweise auf den alten Schlüssel aktualisiert werden. Weitere Informationen finden Sie unter [Rotation von AWS KMS-Schlüsseln](#) im AWS Key Management Service Developer Guide.
- Für die Verschlüsselung eines ElastiCache Caches mithilfe eines KMS-Schlüssels ist ein Grant pro Cache erforderlich. Diese Genehmigung gilt für die Lebensdauer des Caches. Zusätzlich wird eine Genehmigung pro Backup bei der Backup-Erstellung verwendet. Diese Genehmigung wird zurückgezogen, sobald das Backup erstellt wurde.
- Weitere Informationen zu AWS KMS-Zuschüssen und -Limits finden Sie unter [Limits](#) im AWS Key Management Service Developer Guide.

Aktivieren der Verschlüsselung im Ruhezustand

Für alle Serverless-Caches ist die Verschlüsselung im Ruhezustand aktiviert.

Wenn Sie einen knotenbasierten Cluster erstellen, können Sie die Verschlüsselung im Ruhezustand aktivieren, indem Sie den Parameter auf `AtRestEncryptionEnabled true` setzen. Sie können die Verschlüsselung der Daten im Ruhezustand für vorhandene Replikationsgruppen nicht aktivieren.

Sie können die Verschlüsselung im Ruhezustand aktivieren, wenn Sie einen Cache erstellen. ElastiCache Sie können dies mit der AWS-Managementkonsole, der AWS CLI, oder der ElastiCache API tun.

Beim Erstellen eines Caches können Sie eine der folgenden Optionen auswählen:

- Standard – Diese Option verwendet serviceverwaltete Verschlüsselung im Ruhezustand.
- Vom Kunden verwalteter Schlüssel — Mit dieser Option können Sie den Schlüssel ID/ARN von AWS KMS für die Verschlüsselung im Ruhezustand bereitstellen.

Informationen zum Erstellen von AWS KMS-Root-Schlüsseln finden Sie unter [Create Keys](#) im AWS Key Management Service Developer Guide

Inhalt

- [Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS-Managementkonsole](#)
- [Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS CLI](#)

Aktivierung der At-Rest-Verschlüsselung auf einem vorhandenen knotenbasierten Valkey- oder Redis OSS-Cluster

Sie können die Verschlüsselung im Ruhezustand nur aktivieren, wenn Sie eine Valkey- oder Redis OSS-Replikationsgruppe erstellen. Gehen Sie folgendermaßen vor, wenn Sie über eine vorhandene Replikationsgruppe verfügen, für die Sie die Verschlüsselung der Daten im Ruhezustand aktivieren möchten.

Aktivieren der Verschlüsselung der Daten im Ruhezustand für eine vorhandene Replikationsgruppe

1. Erstellen Sie eine manuelle Sicherung Ihrer vorhandenen Replikationsgruppe. Weitere Informationen finden Sie unter [Erstellen manueller Backups](#).
2. Erstellen Sie eine neue Replikationsgruppe, indem Sie sie aus der Sicherung wiederherstellen. Aktivieren Sie in der neuen Replikationsgruppe die Verschlüsselung der Daten im Ruhezustand. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).
3. Aktualisieren Sie die Endpunkte in Ihrer Anwendung so, dass sie auf die neue Replikationsgruppe verweisen.
4. Löschen Sie die alte Replikationsgruppe. Für weitere Informationen siehe [Löschen eines Clusters in ElastiCache](#) oder [Löschen einer Replikationsgruppe](#).

Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS-Managementkonsole

Aktivieren der Verschlüsselung im Ruhezustand für einen Serverless-Cache (Konsole)

Für alle Serverless-Caches ist die Verschlüsselung im Ruhezustand aktiviert. Standardmäßig wird ein KMS-Schlüssel, der AWS-Eigentümer ist, zum Verschlüsseln von Daten verwendet. Um Ihren eigenen AWS KMS Schlüssel auszuwählen, treffen Sie die folgenden Auswahlen:

- Erweitern Sie den Abschnitt Standardeinstellungen.
- Wählen Sie im Abschnitt Standardeinstellungen die Option Standardeinstellungen anpassen aus.
- Wählen Sie im Abschnitt Sicherheit die Option Sicherheitseinstellungen anpassen aus.
- Wählen Sie unter der Einstellung Verschlüsselungsschlüssels die Option Kundenverwalteter CMK aus.
- Wählen Sie unter der Einstellung AWS KMS-Schlüssel einen Schlüssel aus.

Aktivierung der Verschlüsselung im Ruhezustand auf einem knotenbasierten Cluster (Konsole)

Wenn Sie einen eigenen Cache entwerfen, ist bei den Konfigurationen „Dev/Test“ und „Production“ mit der Methode „Easy Create“ die Verschlüsselung im Ruhezustand mithilfe des Schlüssels Standard aktiviert. Legen Sie die folgenden Einstellungen fest, wenn Sie die Konfiguration selbst auswählen:

- Wählen Sie Version 3.2.6, 4.0.10 oder höher als Ihre Engine-Version.
- Aktivieren Sie für die Option Verschlüsselung im Ruhezustand das Kontrollkästchen neben Aktivieren.
- Wählen Sie entweder die Option Standardschlüssel oder Kundenverwalteter CMK.

Das step-by-step Verfahren finden Sie im Folgenden:

- [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS CLI

Um die Verschlüsselung im Ruhezustand zu aktivieren, wenn Sie einen Valkey- oder Redis-OSS-Cluster mithilfe von erstellen AWS CLI, verwenden Sie beim Erstellen einer Replikationsgruppe den `at-rest-encryption-enabled` Parameter --.

Aktivierung der At-Rest-Verschlüsselung auf einem Valkey- oder Redis OSS-Cluster (Cluster-Modus deaktiviert) (CLI)

Der folgende Vorgang erstellt die Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus deaktiviert) `my-classic-rg` mit drei Knoten (`-- num-cache-clusters`), einem primären und zwei Read Replicas. Die Verschlüsselung im Ruhezustand ist für diese Replikationsgruppe aktiviert (`--. at-rest-encryption-enabled`).

Die folgenden Parameter und ihre Werte sind erforderlich, um die Verschlüsselung für diese Replikationsgruppe zu aktivieren:

Hauptparameter

- **--engine**— Muss oder sein. `valkey` oder `redis`
- **--engine-version**—Wenn es sich bei der Engine um Redis OSS handelt, muss es sich um 3.2.6, 4.0.10 oder höher handeln.
- **--at-rest-encryption-enabled**—Erforderlich, um die Verschlüsselung von Daten im Ruhezustand zu aktivieren.

Example 1: Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) mit Replikaten

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id my-classic-rg \  
  --replication-group-description "3 node replication group" \  
  --cache-node-type cache.m4.large \  
  --engine redis \  
  --at-rest-encryption-enabled \  
  --num-cache-clusters 3
```

Für Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id my-classic-rg ^  
  --replication-group-description "3 node replication group" ^  
  --cache-node-type cache.m4.large ^  
  --engine redis ^  
  --at-rest-encryption-enabled ^  
  --num-cache-clusters 3 ^
```

Weitere Informationen finden Sie unter:

- [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Clustermodus deaktiviert\) von Grund auf neu erstellen \(\)AWS CLI](#)
- [create-replication-group](#)

Aktivierung der At-Rest-Verschlüsselung auf einem Cluster für Valkey oder Redis OSS (Cluster-Modus aktiviert) (CLI)

Der folgende Vorgang erstellt die Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) *my-clustered-rg* mit drei Knotengruppen oder Shards (--). num-node-groups Jeder hat drei Knoten, einen primären und zwei Read Replicas (--). replicas-per-node-group Die Verschlüsselung im Ruhezustand ist für diese Replikationsgruppe aktiviert (-- at-rest-encryption-enabled).

Die folgenden Parameter und ihre Werte sind erforderlich, um die Verschlüsselung für diese Replikationsgruppe zu aktivieren:

Hauptparameter

- **--engine**— Muss oder sein. valkey redis
- **--engine-version**—Wenn es sich bei der Engine um Redis OSS handelt, muss es sich um 4.0.10 oder höher handeln.
- **--at-rest-encryption-enabled**—Erforderlich, um die Verschlüsselung von Daten im Ruhezustand zu aktivieren.
- **--cache-parameter-group**—Muss default-redis4.0.cluster.on oder eine davon abgeleitete sein, um diese zu einer Cluster-Modus-fähigen Replikationsgruppe zu machen.

Example 2: Ein Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert)

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id my-clustered-rg \  
  --replication-group-description "redis clustered cluster" \  
  --cache-node-type cache.m3.large \  
  --num-node-groups 3 \  
  --at-rest-encryption-enabled
```

```
--replicas-per-node-group 2 \  
--engine redis \  
--engine-version 6.2 \  
--at-rest-encryption-enabled \  
--cache-parameter-group default.redis6.x.cluster.on
```

Für Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id my-clustered-rg ^  
  --replication-group-description "redis clustered cluster" ^  
  --cache-node-type cache.m3.large ^  
  --num-node-groups 3 ^  
  --replicas-per-node-group 2 ^  
  --engine redis ^  
  --engine-version 6.2 ^  
  --at-rest-encryption-enabled ^  
  --cache-parameter-group default.redis6.x.cluster.on
```

Weitere Informationen finden Sie unter:

- [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Cluster Mode Enabled\) von Grund auf neu erstellen \(\)AWS CLI](#)
- [create-replication-group](#)

Weitere Informationen finden Sie unter:

- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Identity and Access Management für Amazon ElastiCache](#)

Authentifizierung und Autorisierung

AWS Identity and Access Management (IAM) ist ein Webservice, mit dem Sie den Zugriff auf AWS Ressourcen sicher kontrollieren können. ElastiCache unterstützt die Authentifizierung von Benutzern mithilfe von IAM und den Befehlen Valkey und Redis OSS AUTH sowie die Autorisierung von Benutzervorgängen mithilfe von Role-Based Access Control (RBAC).

Themen

- [Rollenbasierte Zugriffskontrolle \(RBAC\)](#)

- [Authentifizierung mit dem Befehl Valkey und Redis OSS AUTH](#)
- [Deaktivieren der Zugriffskontrolle auf einem ElastiCache Valkey- oder Redis-OSS-Cache](#)

Rollenbasierte Zugriffskontrolle (RBAC)

Mit den Befehlen Valkey und Redis OSS AUTH, wie unter beschrieben, [Authentifizierung mit dem Befehl Valkey und Redis OSS AUTH](#) können Sie Role-Based Access Control (RBAC) verwenden. RBAC ist auch die einzige Möglichkeit, den Zugriff auf Serverless-Caches zu kontrollieren. Dies ist für Valkey 7.2 und höher sowie für Redis OSS 6.0 bis 7.2 verfügbar.

RBAC ermöglicht Ihnen:

- Steuern Sie den Cache-Zugriff über Benutzergruppen. Diese Benutzergruppen dienen dazu, den Zugriff auf Caches zu organisieren.
- Verwenden Sie bei AuthN Passwörter pro Benutzer und nicht pro Cluster-Authentifizierungstoken.
- Mit AuthZ verfügen Sie über fein abgestufte Benutzerberechtigungen.
- Basieren Sie Ihren Clusterzugriff auf ACLs

Im Gegensatz zu Valkey und Redis OSS AUTH, bei denen alle authentifizierten Clients vollen Cache-Zugriff haben, wenn ihr Token authentifiziert ist, können Sie mit RBAC Benutzer Gruppen zuweisen, die von den gewünschten Rollen der Benutzer abhängen. Diese Sets dienen dazu, den Zugriff auf Caches zu organisieren.

Mithilfe von RBAC erstellen Sie Benutzer und weisen ihnen bestimmte Berechtigungen zu, indem Sie eine Zugriffszeichenfolge verwenden, wie im Folgenden beschrieben. Sie weisen den Benutzern Gruppen zu, die auf eine bestimmte Rolle zugeschnitten sind (Administratoren, Personalabteilung), die dann in einem oder mehreren ElastiCache Caches bereitgestellt werden. Auf diese Weise können Sie Sicherheitsgrenzen zwischen Clients einrichten, die denselben Valkey- oder Redis-OSS-Cache oder dieselben Caches verwenden, und verhindern, dass Clients gegenseitig auf die Daten zugreifen.

RBAC wurde entwickelt, um die Einführung von [ACL](#) in Redis OSS 6 zu unterstützen. Wenn Sie RBAC mit Ihrem ElastiCache Valkey- oder Redis OSS-Cache verwenden, gibt es einige Einschränkungen:

- Eine für die „VALKEY“-Engine konfigurierte Benutzergruppe kann nur Benutzer enthalten, die einen Authentifizierungsmechanismus (entweder Passwort oder IAM) verwenden. Das bedeutet, dass alle Benutzer mit der Engine „VALKEY“ und alle anderen Benutzer mit der Engine

„Redis“, deren Setup für die Authentifizierung mit Passwort oder IAM konfiguriert ist, zu dieser Benutzergruppe gehören können.

- Bei der Verwendung von RBAC mit Valkey-Clustern können sowohl Benutzergruppen mit der Engine „VALKEY“ als auch mit der Engine „REDIS“ verwendet werden.
- Bei der Verwendung von RBAC mit Redis OSS-Clustern können nur Benutzergruppen mit der Engine „REDIS“ verwendet werden.
- Sie können keine Passwörter in einer Zugriffsfolge angeben. Sie legen Passwörter mit oder Anrufen fest. [CreateUserModifyUser](#)
- Bei Benutzerrechten aktivieren oder deaktivieren Sie Benutzer mit dem on und off als Teil der Zugriffszeichenfolge. Wenn in der Zugriffszeichenfolge keine angegeben ist, wird dem Benutzer off zugewiesen und er hat keine Zugriffsrechte für den Cache.
- Sie können keine verbotenen und umbenannten Befehle als Teil der Zugriffszeichenfolge verwenden. Gibt man einen verbotenen oder einen umbenannten Befehl an, wird eine Ausnahme ausgelöst. Wenn Sie Zugriffskontrolllisten (ACLs) für einen umbenannten Befehl verwenden möchten, geben Sie den ursprünglichen Namen des Befehls an, d. h. den Namen des Befehls vor der Umbenennung.
- Sie können den reset-Befehl als Teil einer Zugriffszeichenfolge nicht benutzen. Sie geben Passwörter mit API-Parametern an, und ElastiCache für Valkey und Redis verwaltet OSS Passwörter. Daher können Sie reset nicht nutzen, da es alle Kennwörter für einen Benutzer entfernen würde.
- Redis OSS 6 führt den Befehl [ACL](#) LIST ein. Dieser Befehl gibt eine Liste von Benutzern zusammen mit den für jeden Benutzer geltenden ACL-Regeln zurück. ElastiCache unterstützt den ACL LIST Befehl, bietet jedoch keine Unterstützung für Passwort-Hashes wie Redis OSS. Mit können Sie den [DescribeUsers](#)Vorgang verwenden ElastiCache, um ähnliche Informationen abzurufen, einschließlich der in der Zugriffszeichenfolge enthaltenen Regeln. Ruft jedoch [DescribeUsers](#)kein Benutzerkennwort ab.
- [Andere schreibgeschützte Befehle, die von ElastiCache for Valkey und Redis OSS unterstützt werden, umfassen ACL WHOAMI, ACL USERS und ACL CAT.](#) ElastiCache für Valkey und Redis unterstützt OSS keine anderen schreibbasierten ACL-Befehle.
- Es gelten die folgenden Limits:

Ressource	Maximum erlaubt
Benutzer pro Benutzergruppe	100

Ressource	Maximum erlaubt
Anzahl Benutzer	1000
Anzahl der Benutzergruppen	100

RBAC mit Valkey

Bei der Verwendung von Role Based Access Control mit Valkey werden Benutzer und Benutzergruppen mit dem Engine-Typ „VALKEY“ erstellt. Dies wird empfohlen, da Valkey mit RBAC standardmäßig eine höhere Sicherheit im Vergleich zu Redis OSS bietet. Sowohl bereitgestellte als auch serverlose Valkey-Cluster unterstützen VALKEY-Benutzer- und Benutzergruppenuordnungen.

Zu den wichtigsten Funktionen von Valkey Access Control gehören:

- Valkey-Benutzer sind nur auf Valkey-Benutzergruppenuordnungen beschränkt.
- Valkey-Benutzergruppen können Valkey-Benutzer und Redis OSS-Benutzer enthalten, die entweder kennwortgeschützt oder IAM-Authentifizierung aktiviert sind.
- Valkey-Benutzer müssen entweder einen Kennwortschutz oder eine IAM-Authentifizierung verwenden.
- VALKEY-Benutzergruppen können nur VALKEY-Clustern zugeordnet werden
- Es gibt keine Standardbenutzeranforderung. Wenn die Valkey-Benutzergruppe Clustern zugeordnet ist, wird die Standardbenutzeranforderung automatisch deaktiviert. Kunden werden sehen, dass der Standardbenutzer ausgeschaltet ist, wenn sie den Befehl ACL LIST verwenden.

Weitere Informationen zur Verwendung von RBAC mit ElastiCache für Valkey und Redis OSS folgen.

Themen

- [Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge](#)
- [Anwenden von RBAC auf einen Cache ElastiCache für Valkey oder Redis OSS](#)
- [Migration von AUTH zu RBAC](#)
- [Migration von RBAC zu AUTH](#)
- [Automatisches Rotieren von Passwörtern für Benutzer](#)
- [Authentifizieren mit IAM](#)

Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge

Um Berechtigungen für einen ElastiCache Valkey- oder Redis OSS-Cache anzugeben, erstellen Sie eine Zugriffszeichenfolge und weisen sie einem Benutzer entweder über oder zu.AWS CLI/AWS-Managementkonsole

Zugriffszeichenfolgen werden als eine Liste von durch Leerzeichen getrennten Regeln definiert, die für den Benutzer angewendet werden. Sie definieren, welche Befehle ein Benutzer ausführen kann und welche Schlüssel ein Benutzer benutzen kann. Um einen Befehl auszuführen, muss ein Benutzer Zugriff auf den ausgeführten Befehl und alle Schlüssel haben, auf die mit dem Befehl zugegriffen wird. Regeln werden von links nach rechts kumulativ angewendet und anstelle der angegebenen Zeichenfolge kann eine einfachere Zeichenfolge verwendet werden, wenn Redundanzen in der angegebenen Zeichenfolge vorhanden sind.

Weitere Informationen zur Syntax der ACL-Regeln finden Sie unter [ACL](#).

Im folgenden Beispiel wird durch die Zugriffszeichenfolge ein aktiver Benutzer dargestellt, der Zugriff auf alle verfügbaren Schlüssel und Befehle hat.

```
on ~* +@all
```

Die Syntax der Zugriffszeichenfolge wird wie folgt verteilt:

- `on` – Der Benutzer ist ein aktiver Benutzer.
- `~*` – Zugriff auf alle verfügbaren Schlüssel ist erlaubt.
- `+@all` – Zugriff auf alle verfügbaren Befehle ist erlaubt.

Die vorgenannten Einstellungen sind am wenigsten restriktiv. Sie können diese Einstellungen ändern und sie sicherer zu machen.

Das folgende Beispiel zeigt einen Benutzer, der nur Lesezugriff auf Schlüssel hat, die mit dem Schlüsselbereich "app:." beginnen

```
on ~app:.* -@all +@read
```

Sie können diese Berechtigungen weiter verfeinern, indem Sie die Befehle auflisten, auf die der Benutzer zugreifen kann:

`+command1` – Der Zugriff des Benutzers auf Befehle ist auf `command1` beschränkt.

+@category – Der Zugriff des Benutzers auf Befehle ist auf eine Kategorie von Befehlen beschränkt.

Informationen zum Zuweisen einer Zugriffszeichenfolge zu einem Benutzer finden Sie unter [Erstellen von Benutzern und Benutzergruppen mit der Konsole und dem CLI](#).

Wenn Sie einen vorhandenen Workload zu migrieren ElastiCache, können Sie die Zugriffszeichenfolge durch einen Aufruf abrufen `ACL LIST`, wobei der Benutzer und alle Kennwort-Hashes ausgenommen sind.

Für Redis OSS Version 6.2 und höher wird auch die folgende Syntax für die Zugriffszeichenfolge unterstützt:

- `&*` – Zugriff auf alle verfügbaren Kanäle ist erlaubt.

Für Redis OSS Version 7.0 und höher wird auch die folgende Syntax für Zugriffszeichenfolgen unterstützt:

- `|` – kann zum Blockieren von Unterbefehlen verwendet werden (z. B. „-config|set“).
- `%R~<pattern>` – fügt das angegebene Leseschlüsselmuster hinzu. Dies verhält sich ähnlich wie das reguläre Schlüsselmuster, gewährt jedoch nur die Erlaubnis, aus Schlüsseln zu lesen, die dem angegebenen Muster entsprechen. Weitere Informationen finden Sie unter [Wichtigen Berechtigungen](#).
- `%W~<pattern>` – fügt das angegebene Schreibschlüsselmuster hinzu. Dies verhält sich ähnlich wie das reguläre Schlüsselmuster, gewährt jedoch nur die Erlaubnis, in Schlüssel zu schreiben, die dem angegebenen Muster entsprechen. Weitere Informationen finden Sie unter [ACL-Schlüsselberechtigungen](#).
- `%RW~<pattern>` – Alias für `~<pattern>`.
- `(<rule list>)` – erstellt einen neuen Selektor, mit dem die Regeln verglichen werden sollen. Selektoren werden nach den Benutzerberechtigungen und in der Reihenfolge bewertet, in der sie definiert sind. Wenn ein Befehl entweder den Benutzerberechtigungen oder einem beliebigen Selektor entspricht, ist er zulässig. Weitere Informationen finden Sie unter [ACL-Selektoren](#).
- `clearselectors` – löscht alle an den Benutzer angehängten Selektoren.

Anwenden von RBAC auf einen Cache ElastiCache für Valkey oder Redis OSS

Um RBAC ElastiCache für Valkey oder Redis OSS zu verwenden, gehen Sie wie folgt vor:

1. Erstellung eines oder mehrerer Benutzer.

2. Erstellen Sie eine Benutzergruppe und fügen Sie der Gruppe Benutzer hinzu.
3. Weisen Sie die Benutzergruppe einen Cache zu, bei dem die Verschlüsselung während der Übertragung aktiviert ist.

Diese Schritte werden im Folgenden detailliert beschrieben.

Themen

- [Erstellen von Benutzern und Benutzergruppen mit der Konsole und dem CLI](#)
- [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#)
- [Zuweisen von Benutzergruppen zu Serverless-Caches](#)
- [Zuweisung von Benutzergruppen zu Replikationsgruppen](#)

Erstellen von Benutzern und Benutzergruppen mit der Konsole und dem CLI

Die Benutzerinformationen für RBAC-Benutzer sind eine Benutzer-ID, ein Benutzername, sowie optional ein Kennwort und eine Zugriffszeichenfolge. Die Zugriffszeichenfolge stellt die Berechtigungsstufe für Schlüssel und Befehle bereit. Die Benutzer-ID ist eindeutig für den Benutzer und der Benutzername wird an die Engine übergeben.

Vergewissern Sie sich, dass die von Ihnen bereitgestellten Benutzerberechtigungen für den beabsichtigten Zweck der Benutzergruppe sinnvoll sind. Wenn Sie beispielsweise eine Benutzergruppe mit dem Namen `Administrators` erstellen, sollte jeder Benutzer, den Sie dieser Gruppe hinzufügen, seine Zugriffszeichenfolge auf vollen Zugriff auf Schlüssel und Befehle gesetzt haben. Für Benutzer in einer `e-commerce`-Benutzergruppe können Sie die Zugriffszeichenfolge auf Nur-Lese-Zugriff setzen.

ElastiCache konfiguriert automatisch einen Standardbenutzer mit Benutzer-ID und Benutzername `"default"` und fügt ihn allen Benutzergruppen hinzu. Sie können diesen Benutzer nicht löschen oder ändern. Dieser Benutzer ist aus Kompatibilitätsgründen mit dem Standardverhalten früherer Redis OSS-Versionen vorgesehen und verfügt über eine Zugriffszeichenfolge, die es ihm ermöglicht, alle Befehle aufzurufen und auf alle Schlüssel zuzugreifen.

Wenn Sie einem Cache eine angemessene Zugriffskontrolle hinzufügen möchten, ersetzen Sie diesen Standardbenutzer durch einen neuen, der nicht aktiviert ist oder ein sicheres Passwort verwendet. Um den Standardbenutzer zu ändern, erstellen Sie einen neuen Benutzer, dessen Benutzername auf `default` gesetzt ist. Sie können ihn dann mit dem ursprünglichen Standardbenutzer austauschen.

Führen Sie die folgenden Schritte aus, wie Sie den ursprünglichen default-Benutzer mit einem anderen default-Benutzer mit einer geänderten Zugriffszeichenfolge getauscht haben.

Ändern Sie den Standardbenutzer in der Konsole wie folgt:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus dem Navigationsbereich Benutzergruppenverwaltung aus.
3. Wählen Sie für Benutzergruppen-ID die ID aus, die Sie ändern möchten. Stellen Sie sicher, dass Sie den Link und nicht das Kontrollkästchen ausgewählt haben.
4. Wählen Sie Ändern aus.
5. Wählen Sie im Fenster Ändern die Option Verwalten aus. Wählen Sie unter „Wählen Sie den gewünschten Benutzer aus“ den Benutzer mit dem Standardbenutzernamen aus.
6. Wählen Sie Choose (Auswählen) aus.
7. Wählen Sie Ändern aus. Wenn Sie dies tun, werden alle vorhandenen Verbindungen mit einem Cache, die der ursprüngliche Standardbenutzer hatte, beendet.

Um den Standardbenutzer mit dem zu ändern AWS CLI

1. Erstellen Sie einen neuen Benutzer mit dem Benutzernamen default unter Verwendung der folgenden Befehle.

Für Linux, macOS oder Unix:

```
aws elasticache create-user \  
  --user-id "new-default-user" \  
  --user-name "default" \  
  --engine "VALKEY" \  
  --passwords "a-strong-password" \  
  --access-string "off +get ~keys*"
```

Für Windows:

```
aws elasticache create-user ^  
  --user-id "new-default-user" ^  
  --user-name "default" ^  
  --engine "VALKEY" ^  
  --passwords "a-strong-password" ^
```

```
--access-string "off +get ~keys*"
```

- Erstellen Sie eine Benutzergruppe und fügen Sie den von Ihnen zuvor erstellten Benutzer hinzu.

Für Linux, macOS oder Unix:

```
aws elasticache create-user-group \  
  --user-group-id "new-group-2" \  
  --engine "VALKEY" \  
  --user-ids "new-default-user"
```

Für Windows:

```
aws elasticache create-user-group ^  
  --user-group-id "new-group-2" ^  
  --engine "VALKEY" ^  
  --user-ids "new-default-user"
```

Beim Erstellen eines Benutzers können Sie bis zu zwei Passwörter einrichten. Wenn Sie ein Passwort ändern, werden alle vorhandenen Verbindungen mit Caches beibehalten.

Beachten Sie insbesondere die folgenden Einschränkungen bei Benutzerkennwörtern, wenn Sie RBAC for ElastiCache für Valkey und Redis OSS verwenden:

- Passwörter müssen 16-128 druckbare Zeichen enthalten.
- Folgende nicht-alphanumerische Zeichen sind nicht zulässig: , " " / @.

Verwalten von Benutzern mit der Konsole und dem CLI

Gehen Sie wie unten beschrieben vor, um die Benutzer in der Konsole zu verwalten.

Verwalten von Benutzern in der Konsole

- Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
- Wählen Sie im ElastiCache Amazon-Dashboard Benutzerverwaltung aus. Die folgenden Optionen sind verfügbar:

- **Benutzer erstellen** – Beim Erstellen eines Benutzers geben Sie eine Benutzer-ID, einen Benutzernamen, einen Authentifizierungsmodus und eine Zugriffszeichenfolge ein. Die Zugriffszeichenfolge legt die Berechtigungsstufe fest, welche Schlüssel und Befehle für den Benutzer erlaubt sind.

Beim Erstellen eines Benutzers können Sie bis zu zwei Passwörter einrichten. Wenn Sie ein Passwort ändern, werden alle vorhandenen Verbindungen mit Caches beibehalten.

- **Benutzer ändern** – Ermöglicht Ihnen, die Authentifizierungseinstellungen eines Benutzers zu aktualisieren oder dessen Zugriffszeichenfolge zu ändern.
- **Benutzer löschen** – Das Konto wird aus allen Benutzergruppen entfernt, zu denen es gehört.

Gehen Sie wie unten beschrieben vor, um die Benutzer im AWS CLI zu verwalten.

So ändern Sie einen Benutzer mit dem CLI

- Verwenden des `modify-user`-Befehls, um das Passwort oder die Passwörter eines Benutzers zu aktualisieren oder die Zugriffsberechtigungen eines Benutzers zu ändern.

Wenn ein Benutzer geändert wird, werden die Benutzergruppen, die dem Benutzer zugeordnet sind, gemeinsam mit allen Caches aktualisiert, die der Benutzergruppe zugeordnet sind. Alle vorhandenen Verbindungen werden gewartet. Im Folgenden sind einige Beispiele aufgeführt.

Für Linux, macOS oder Unix:

```
aws elasticache modify-user \  
  --user-id user-id-1 \  
  --access-string "~objects:* ~items:* ~public:*" \  
  --authentication-mode Type=iam
```

Für Windows:

```
aws elasticache modify-user ^  
  --user-id user-id-1 ^  
  --access-string "~objects:* ~items:* ~public:*" ^  
  --authentication-mode Type=iam
```

 Note

Wir raten davon ab, die `nopass`-Option zu nutzen. Wenn Sie dies tun, empfehlen wir, die Berechtigungen des Benutzers auf Nur-Lese-Zugriff und auf einen begrenzten Satz von Schlüsseln festzulegen.

So löschen Sie einen Benutzer mit dem CLI

- Verwenden Sie den Befehl `delete-user`, um einen Benutzer zu löschen. Das Konto wird gelöscht und aus allen zugehörigen Benutzergruppen entfernt. Im Folgenden wird ein -Beispiel gezeigt.

Für Linux, macOS oder Unix:

```
aws elasticache delete-user \  
--user-id user-id-2
```

Für Windows:

```
aws elasticache delete-user ^  
--user-id user-id-2
```

Um eine Liste der Benutzer anzuzeigen, rufen Sie die [describe-user](#)-Operation auf.

```
aws elasticache describe-users
```

Verwalten von Benutzergruppen mit der Konsole und dem CLI

Sie können Benutzergruppen erstellen, um den Zugriff von Benutzern auf einen oder mehrere Caches zu organisieren und zu steuern, wie im Folgenden dargestellt.

Gehen Sie wie unten beschrieben vor, um die Benutzergruppen in der Konsole zu verwalten.

So verwalten Sie Benutzergruppen mithilfe der Konsole

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Amazon-Dashboard die Option Benutzergruppenverwaltung aus.

Die folgenden Operationen sind verfügbar, um neue Benutzergruppen zu erstellen:

- Erstellen – Wenn Sie eine Benutzergruppe erstellen, fügen Sie Benutzer hinzu und ordnen die Benutzergruppen dann Caches zu. So können Sie beispielsweise eine Admin-Benutzergruppe für Benutzer erstellen, die über Administratorrollen in einem Cache verfügen.

 Important

Wenn Sie keine Valkey- oder Redis OSS-Benutzergruppe verwenden, müssen Sie beim Erstellen einer Benutzergruppe einen Standardbenutzer angeben.

- Benutzer hinzufügen – Fügen Sie der Benutzergruppe Benutzer hinzu.
- Entfernen von Benutzern – Entfernt Benutzer aus der Benutzergruppe. Wenn Benutzer aus einer Benutzergruppe entfernt werden, werden alle vorhandenen Verbindungen, die diese mit einem Cache haben, beendet.
- Löschen – Verwenden Sie diese Option, um eine Benutzergruppe zu löschen. Beachten Sie, dass nur die Benutzergruppe selbst und nicht die Benutzer, die zu dieser Gruppe gehören, gelöscht wird.

Für bestehende Benutzergruppen können Sie die folgenden Aktionen ausführen:

- Benutzer hinzufügen – Fügen Sie der Benutzergruppe bereits vorhandene Benutzer hinzu.
- Benutzer löschen – Entfernt vorhandene Benutzer aus der Benutzergruppe.

 Note

Benutzer werden von der Benutzergruppe entfernt, aber nicht aus dem System gelöscht.

Gehen Sie wie unten beschrieben vor, um die Benutzergruppen im CLI zu verwalten.

So erstellen Sie eine neue Benutzergruppe und fügen einen Benutzer mittels der CLI hinzu

- Geben Sie wie nachfolgend gezeigt den `create-user-group`-Befehl ein.

Für Linux, macOS oder Unix:


```
aws elasticache create-user-group \  
  --user-group-id "new-group-1" \  
  --engine "VALKEY" \  
  --user-ids user-id-1, user-id-2
```

Für Windows:

```
aws elasticache create-user-group ^  
  --user-group-id "new-group-1" ^  
  --engine "VALKEY" ^  
  --user-ids user-id-1, user-id-2
```

So ändern Sie eine Benutzergruppe durch Hinzufügen neuer Benutzer oder Entfernen aktueller Mitglieder mit der CLI

- Geben Sie wie nachfolgend gezeigt den `modify-user-group`-Befehl ein.

Für Linux, macOS oder Unix:

```
aws elasticache modify-user-group --user-group-id new-group-1 \  
  --user-ids-to-add user-id-3 \  
  --user-ids-to-remove user-id-2
```

Für Windows:

```
aws elasticache modify-user-group --user-group-id new-group-1 ^  
  --user-ids-to-add user-id-3 ^  
  --user-ids-to-remove user-id-2
```

Note

Alle offenen Verbindungen, die zu einem Benutzer gehören, der aus einer Benutzergruppe entfernt wurde, werden mit diesem Befehl beendet.

So löschen Sie eine Benutzergruppe mit dem CLI

- Geben Sie wie nachfolgend gezeigt den `delete-user-group`-Befehl ein. Die Benutzergruppe selbst, nicht die zur Gruppe gehörenden Benutzer, wird gelöscht.

Für Linux, macOS oder Unix:

```
aws elasticache delete-user-group /  
  --user-group-id
```

Für Windows:

```
aws elasticache delete-user-group ^  
  --user-group-id
```

Um eine Liste der Benutzergruppen zu sehen, können Sie den [describe-user-groups](#) Vorgang aufrufen.

```
aws elasticache describe-user-groups \  
  --user-group-id test-group
```

Zuweisen von Benutzergruppen zu Serverless-Caches

Nachdem Sie eine Benutzergruppe erstellt und Benutzer hinzugefügt haben, besteht der letzte Schritt bei der Implementierung von RBAC darin, diese Benutzergruppe einem Serverless-Cache zuzuweisen.

Zuweisen von Benutzergruppen zu Serverless-Caches mithilfe der Konsole

Gehen Sie wie folgt vor, um einem serverlosen Cache mithilfe von eine Benutzergruppe hinzuzufügen: AWS-Managementkonsole

- Informationen zum deaktivierten Cluster-Modus finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- Informationen zum aktivierten Cluster-Modus finden Sie unter [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Zuweisen von Benutzergruppen zu serverlosen Caches mithilfe von AWS CLI

Der folgende AWS CLI Vorgang erstellt einen serverlosen Cache unter Verwendung des `user-group-id` Parameters mit dem Wert. *my-user-group-id* Ersetzen Sie die Subnetzgruppe `sng-test` durch eine vorhandene Subnetzgruppe.

Hauptparameter

- **--engine**— Muss VALKEY oder REDIS sein.
- **--user-group-id** – Dieser Wert gibt die ID der Benutzergruppe an, die aus Benutzern mit bestimmten Zugriffsberechtigungen für den Cache besteht.

Für Linux, macOS oder Unix:

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name "new-serverless-cache" \  
  --description "new-serverless-cache" \  
  --engine "VALKEY" \  
  --user-group-id "new-group-1"
```

Für Windows:

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name "new-serverless-cache" ^  
  --description "new-serverless-cache" ^  
  --engine "VALKEY" ^  
  --user-group-id "new-group-1"
```

Die folgende AWS CLI Operation ändert einen serverlosen Cache mit dem `user-group-id` Parameter mit dem Wert. *my-user-group-id*

Für Linux, macOS oder Unix:

```
aws elasticache modify-serverless-cache \  
  --serverless-cache-name serverless-cache-1 \  
  --user-group-id "new-group-2"
```

Für Windows:

```
aws elasticache modify-serverless-cache ^
```

```
--serverless-cache-name serverless-cache-1 ^  
--user-group-id "new-group-2"
```

Beachten Sie, dass jegliche an einem Cache vorgenommenen Änderungen asynchron aktualisiert werden. Sie können den Fortschritt durch Ansicht der Ereignismeldungen überwachen. Weitere Informationen finden Sie unter [ElastiCache Ereignisse anzeigen](#).

Zuweisung von Benutzergruppen zu Replikationsgruppen

Nachdem Sie eine Benutzergruppe erstellt und Benutzer hinzugefügt haben, besteht der letzte Schritt bei der Implementierung von RBAC darin, diese Benutzergruppe einer Replikationsgruppe zuzuweisen.

Zuweisen von Benutzergruppen zu Replikationsgruppen mit der Konsole

Gehen Sie wie folgt vor, um einer Replikation mithilfe von eine Benutzergruppe hinzuzufügen:AWS-Managementkonsole

- Informationen zum deaktivierten Cluster-Modus finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- Informationen zum aktivierten Cluster-Modus finden Sie unter [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Zuweisen von Benutzergruppen zu Replikationsgruppen mithilfe von AWS CLI

Der folgende AWS CLI Vorgang erstellt eine Replikationsgruppe mit aktivierter Verschlüsselung bei der Übertragung (TLS) und dem user-group-ids Parameter mit dem Wert *my-user-group-id*. Ersetzen Sie die Subnetzgruppe sng-test durch eine vorhandene Subnetzgruppe.

Hauptparameter

- **--engine**— Muss valkey oder seinredis.
- **--engine-version** – Muss 6.0 oder höher sein.
- **--transit-encryption-enabled** – Erforderlich für die Authentifizierung und zum Zuordnen einer Benutzergruppe.
- **--user-group-ids** – Dieser Wert gibt die ID der Benutzergruppe an, die aus Benutzern mit bestimmten Zugriffsberechtigungen für den Cache besteht.
- **--cache-subnet-group** – Erforderlich für die Zuordnung einer Benutzergruppe.

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id "new-replication-group" \  
  --replication-group-description "new-replication-group" \  
  --engine "VALKEY" \  
  --cache-node-type cache.m5.large \  
  --transit-encryption-enabled \  
  --user-group-ids "new-group-1" \  
  --cache-subnet-group "cache-subnet-group"
```

Für Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id "new-replication-group" ^  
  --replication-group-description "new-replication-group" ^  
  --engine "VALKEY" ^  
  --cache-node-type cache.m5.large ^  
  --transit-encryption-enabled ^  
  --user-group-ids "new-group-1" ^  
  --cache-subnet-group "cache-subnet-group"
```

Der folgende AWS CLI Vorgang ändert eine Replikationsgruppe mit aktivierter Verschlüsselung bei der Übertragung (TLS) und dem user-group-ids Parameter mit dem Wert *my-user-group-id*.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id replication-group-1 \  
  --user-group-ids-to-remove "new-group-1" \  
  --user-group-ids-to-add "new-group-2"
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id replication-group-1 ^  
  --user-group-ids-to-remove "new-group-1" ^  
  --user-group-ids-to-add "new-group-2"
```

Beachten Sie die PendingChanges in der Antwort. Jegliche an einem Cache vorgenommenen Änderungen werden asynchron aktualisiert. Sie können den Fortschritt durch Ansicht der

Ereignismeldungen überwachen. Weitere Informationen finden Sie unter [ElastiCache Ereignisse anzeigen](#).

Migration von AUTH zu RBAC

Wenn Sie AUTH wie unter beschrieben verwenden [Authentifizierung mit dem Befehl Valkey und Redis OSS AUTH](#) und zu RBAC migrieren möchten, gehen Sie wie folgt vor.

Gehen Sie wie folgt vor, um mithilfe der Konsole von AUTH zu RBAC zu migrieren.

Um mithilfe der Konsole von Valkey oder Redis OSS AUTH zu RBAC zu migrieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der sich der Cache befindet, den Sie ändern möchten.
3. Wählen Sie im Navigationsbereich die Engine aus, die auf dem Cache ausgeführt wird, den Sie ändern möchten.

Es wird eine Liste der Caches der ausgewählten Engine angezeigt.

4. Wählen Sie in der Liste der Caches den Namen des Caches aus, den Sie ändern möchten.
5. Wählen Sie für Actions (Aktionen) die Option Modify (Ändern) aus.

Das Fenster Ändern wird angezeigt.

6. Wählen Sie für Zugriffskontroll die Option Zugriffskontrollliste für Benutzergruppen aus.
7. Wählen Sie für Zugriffskontrollliste für Benutzergruppen eine Benutzergruppe aus.
8. Wählen Sie Änderungen in der Vorschau anzeigen und dann auf dem nächsten Bildschirm Ändern aus.

Verwenden Sie das folgende Verfahren, um mithilfe der CLI von Valkey oder Redis OSS AUTH zu RBAC zu migrieren.

So migrieren Sie mit der CLI von AUTH zu RBAC

- Geben Sie wie nachfolgend gezeigt den `modify-replication-group`-Befehl ein.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group --replication-group-id test \
```

```
--auth-token-update-strategy DELETE \  
--user-group-ids-to-add user-group-1
```

Für Windows:

```
aws elasticache modify-replication-group --replication-group-id test ^  
--auth-token-update-strategy DELETE ^  
--user-group-ids-to-add user-group-1
```

Migration von RBAC zu AUTH

Wenn Sie RBAC verwenden und zu Redis OSS AUTH migrieren möchten, finden Sie weitere Informationen unter [Migration von RBAC zu AUTH](#)

Note

Wenn Sie die Zugriffskontrolle für einen ElastiCache Cache deaktivieren müssen, müssen Sie dies über den tun.AWS CLI Weitere Informationen finden Sie unter [the section called "Deaktivieren der Zugriffskontrolle auf einem ElastiCache Valkey- oder Redis-OSS-Cache"](#).

Automatisches Rotieren von Passwörtern für Benutzer

Mit AWS Secrets Manager können Sie hartcodierte Anmeldeinformationen in Ihrem Code (einschließlich Kennwörtern) durch einen API-Aufruf an Secrets Manager ersetzen, um das Geheimnis programmgesteuert abzurufen. Dadurch wird sichergestellt, dass das Secret nicht kompromittiert werden kann, wenn jemand Ihren Code durchsucht, da es sich gar nicht dort befindet. Außerdem können Sie Secrets Manager so konfigurieren, dass er das Secret automatisch nach einem von Ihnen festgelegten Zeitplan rotiert. So können Sie Secrets mit langer Einsatzdauer durch Secrets mit kurzer Einsatzdauer ersetzen und damit das Risiko einer Kompromittierung erheblich verringern.

Mit Secrets Manager können Sie Ihre OSS-Passwörter (d. h. Secrets) ElastiCache für Redis mithilfe einer von Secrets Manager AWS Lambda bereitgestellten Funktion automatisch rotieren.

Weitere Informationen zu finden Sie AWS Secrets Manager unter [Was ist AWS Secrets Manager?](#)

Wie ElastiCache verwendet Secrets

Valkey 7.2 und höher verfügen über einen gleichwertigen Funktionsumfang wie Redis OSS 7.0. In Redis OSS 6, ElastiCache eingeführt, [Rollenbasierte Zugriffskontrolle \(RBAC\)](#) um den Valkey- oder Redis OSS-Cluster zu sichern. Mit dieser Funktion können bestimmte Verbindungen bezüglich der ausführbaren Befehle und der Tasten, auf die zugegriffen werden kann, eingeschränkt werden. Bei RBAC erstellt der Kunde zwar einen Benutzer mit Passwörtern, die Passwortwerte müssen jedoch manuell im Klartext eingegeben werden und sind für den Bediener sichtbar.

Mit Secrets Manager wird das Passwort von Anwendungen aus Secrets Manager abgerufen, anstatt es manuell einzugeben und in der Anwendungsconfiguration zu speichern. Weitere Informationen hierzu finden Sie unter [Wie werden ElastiCache Benutzer mit dem Geheimnis verknüpft](#).

Für die Verwendung von Secrets fallen Kosten an. Preisinformationen finden Sie unter [AWS Secrets Manager – Preise](#).

Wie werden ElastiCache Benutzer mit dem Geheimnis verknüpft

Secrets Manager speichert eine Referenz für den zugehörigen Benutzer im `SecretString`-Feld des Secrets. Von der ElastiCache Seite wird es keinen Hinweis auf das Geheimnis geben.

```
{
  "password": "strongpassword",
  "username": "user1",
  "user_arn": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1" //this is the
bond between the secret and the user
}
```

Lambda-Rotationsfunktion

Zur Aktivierung der automatischen Passwortrotation von Secrets Manager erstellen Sie eine Lambda-Funktion, die mit der [modify-user](#)-API interagiert, um die Passwörter des Benutzers zu aktualisieren.

Informationen zur Funktionsweise finden Sie unter [Funktionsweise der Drehung](#).

Note

Für einige AWS Dienste AWS empfiehlt es sich, sowohl den Bedingungsschlüssel als auch den `aws:SourceAccount` globalen Bedingungsschlüssel zu verwenden, um das `aws:SourceArn` Szenario eines verwirrten Stellvertreters zu vermeiden. Wenn Sie jedoch die `aws:SourceArn`-Bedingung in Ihre Drehungsfunktions-Richtlinie einschließen, kann die

Drehungsfunktion nur verwendet werden, um das von diesem ARN angegebene Secret zu rotieren. Es wird empfohlen, nur den Kontextschlüssel `aws:SourceAccount` anzugeben, damit Sie die Drehungsfunktion für mehrere Geheimnisse verwenden können.

Informationen zu Problemen, auf die Sie möglicherweise stoßen, finden Sie unter [Fehlerbehebung bei der AWS Secrets Manager Manager-Rotation](#).

So erstellen Sie einen ElastiCache Benutzer und verknüpfen ihn mit Secrets Manager

Die folgenden Schritte veranschaulichen, wie Sie einen Benutzer erstellen und ihn mit Secrets Manager verknüpfen:

1. Einen inaktiven Benutzer erstellen

Für Linux, macOS oder Unix:

```
aws elasticache create-user \  
  --user-id user1 \  
  --user-name user1 \  
  --engine "REDIS" \  
  --no-password \ // no authentication is required  
  --access-string "*off* +get ~keys*" // this disables the user
```

Für Windows:

```
aws elasticache create-user ^  
  --user-id user1 ^  
  --user-name user1 ^  
  --engine "REDIS" ^  
  --no-password ^ // no authentication is required  
  --access-string "*off* +get ~keys*" // this disables the user
```

Es wird eine Antwort ähnlich dem folgenden Beispiel angezeigt:

```
{  
  "UserId": "user1",  
  "UserName": "user1",  
  "Status": "active",  
  "Engine": "redis",  
  "AccessString": "off ~keys* -@all +get",
```

```
"UserGroupIds": [],
"Authentication": {
  "Type": "no_password"
},
"ARN": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1"
}
```

2. Erstellen eines -Secrets

Für Linux, macOS oder Unix:

```
aws secretsmanager create-secret \
--name production/ec/user1 \
--secret-string \
'{
  "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
  "username": "user1"
}'
```

Für Windows:

```
aws secretsmanager create-secret ^
--name production/ec/user1 ^
--secret-string ^
'{
  "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
  "username": "user1"
}'
```

Es wird eine Antwort ähnlich dem folgenden Beispiel angezeigt:

```
{
  "ARN": "arn:aws:secretsmanager:us-east-1:123456xxxx:secret:production/ec/user1-
eaFois",
  "Name": "production/ec/user1",
  "VersionId": "aae5b963-1e6b-4250-91c6-ebd6c47d0d95"
}
```

3. Eine Lambda-Funktion zur Rotation Ihres Passworts konfigurieren

- a. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Lambda-Konsole unter <https://console.aws.amazon.com/lambda/>

- b. Wählen Sie im Navigationsbereich Functions (Funktionen) und dann die erstellte Funktion aus. Wählen Sie den Funktionsnamen aus, nicht das Kontrollkästchen links davon.
- c. Wählen Sie die Registerkarte Konfiguration aus.
- d. Wählen Sie unter General configuration (Allgemeine Konfiguration) die Option Edit (Bearbeiten) aus und legen Sie die Timeout (Zeitüberschreitung) anschließend auf mindestens 12 Minuten fest.
- e. Wählen Sie Speichern.
- f. Wählen Sie Environment variables (Umgebungsvariablen) aus und legen Sie dann Folgendes fest:
 - i. SECRETS_MANAGER_ENDPOINT – <https://secretsmanager.REGION.amazonaws.com>
 - ii. SERET_ARN – der Amazon-Ressourcenname (ARN) des Secrets, das Sie in Schritt 2 erstellt haben.
 - iii. USER_NAME — Benutzername des Benutzers, ElastiCache
 - iv. Wählen Sie Speichern.
- g. Wählen Sie Permissions (Berechtigungen) aus.
- h. Wählen Sie unter Execution role (Ausführungsrolle) den Namen der Lambda-Funktionsrolle aus, die auf der IAM-Konsole angezeigt werden soll.
- i. Die Lambda-Funktion benötigt die folgende Berechtigung, um die Benutzer zu ändern und das Passwort festzulegen:

ElastiCache

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:DescribeUsers",
        "elasticache:ModifyUser"
      ],
      "Resource": "arn:aws:elasticache:us-east-1:123456789012:user:user1"
    }
  ]
}
```

```
]
}
```

Secrets Manager

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:PutSecretValue",
        "secretsmanager:UpdateSecretVersionStage"
      ],
      "Resource": "arn:aws:secretsmanager:us-east-1:123456789012:secret:XXXX"
    },
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetRandomPassword",
      "Resource": "*"
    }
  ]
}
```

4. Die Secrets-Manager-Rotation einrichten

- a. Verwenden von AWS-Managementkonsole, siehe [Automatische Rotation für AWS Secrets Manager Manager-Geheimnisse mithilfe der Konsole einrichten](#)

Weitere Informationen zum Einrichten eines Rotationszeitplans finden Sie unter [Zeitplanausdrücke in Secrets-Manager-Rotation](#).

- b. Verwenden von AWS CLI, siehe [Automatische Rotation einrichten für die AWS Secrets Manager Verwendung von AWS Command Line Interface](#)

Authentifizieren mit IAM

Themen

- [-Übersicht](#)
- [Einschränkungen](#)
- [Einrichtung](#)
- [Herstellen von Verbindungen](#)

-Übersicht

Mit der IAM-Authentifizierung können Sie eine Verbindung zu Valkey oder Redis OSS mithilfe von AWS IAM-Identitäten authentifizieren, wenn Ihr Cache ElastiCache für die Verwendung von Valkey oder Redis OSS Version 7 oder höher konfiguriert ist. Auf diese Weise können Sie Ihr Sicherheitsmodell stärken und viele administrative Sicherheitsaufgaben vereinfachen. Sie können die IAM-Authentifizierung auch verwenden, um eine detaillierte Zugriffskontrolle für jeden einzelnen Cache und Benutzer zu konfigurieren, wobei die Prinzipien der geringsten Zugriffsrechte eingehalten werden. ElastiCache ElastiCache Die IAM-Authentifizierung für ElastiCache funktioniert durch die Bereitstellung eines kurzlebigen IAM-Authentifizierungstoken anstelle eines langlebigen ElastiCache Benutzerkennworts im OSS oder Befehl Valkey oder Redis. AUTH HELLO Weitere Informationen zum IAM-Authentifizierungstoken finden Sie im AWS Allgemeinen Referenzhandbuch im [Signaturprozess für Signature Version 4](#) und im folgenden Codebeispiel.

Sie können IAM-Identitäten und die zugehörigen Richtlinien verwenden, um den Zugriff auf Valkey oder Redis OSS weiter einzuschränken. Sie können Benutzern von ihren Federated Identity-Anbietern auch direkten Zugriff auf Valkey- oder Redis OSS-Caches gewähren.

Um AWS IAM mit verwenden zu können ElastiCache, müssen Sie zunächst einen ElastiCache Benutzer erstellen, dessen Authentifizierungsmodus auf IAM eingestellt ist. Anschließend können Sie eine IAM-Identität erstellen oder wiederverwenden. Für die IAM-Identität ist eine zugehörige Richtlinie erforderlich, um dem ElastiCache Cache und ElastiCache dem `elasticache:Connect` Benutzer die Aktion zu gewähren. Nach der Konfiguration können Sie ein IAM-Authentifizierungstoken mit den AWS Anmeldeinformationen des IAM-Benutzers oder der IAM-Rolle erstellen. Schließlich müssen Sie das kurzlebige IAM-Authentifizierungstoken als Passwort in Ihrem Valkey- oder Redis OSS-Client angeben, wenn Sie eine Verbindung zu Ihrem Cache herstellen. Ein Valkey- oder Redis OSS-Client mit Unterstützung für den Anbieter von Anmeldeinformationen kann die temporären Anmeldeinformationen für jede neue Verbindung automatisch generieren. ElastiCache führt die

IAM-Authentifizierung für Verbindungsanfragen von IAM-fähigen ElastiCache Benutzern durch und validiert die Verbindungsanfragen mit IAM.

Einschränkungen

Bei der Verwendung der IAM-Authentifizierung gelten die folgenden Einschränkungen:

- Die IAM-Authentifizierung ist verfügbar, wenn sie ElastiCache für Valkey 7.2 und höher oder Redis OSS Version 7.0 und höher verwendet wird.
- Für IAM-fähige ElastiCache Benutzer müssen die Eigenschaften Benutzername und Benutzer-ID identisch sein.
- Das IAM-Authentifizierungstoken ist für 15 Minuten gültig. Für langlebige Verbindungen empfehlen wir die Verwendung eines Valkey- oder Redis-OSS-Clients, der eine Schnittstelle zum Anbieter von Anmeldeinformationen unterstützt.
- Eine IAM-authentifizierte Verbindung zu Valkey oder Redis OSS wird ElastiCache nach 12 Stunden automatisch unterbrochen. Die Verbindung kann um 12 Stunden verlängert werden, indem ein AUTH- oder HELLO-Befehl mit einem neuen IAM-Authentifizierungstoken gesendet wird.
- Die IAM-Authentifizierung wird in MULTI EXEC-Befehlen nicht unterstützt.
- Derzeit unterstützt die IAM-Authentifizierung die folgenden globalen Bedingungskontextschlüssel:
 - Bei Verwendung der IAM-Authentifizierung mit Serverless-Caches werden `aws:VpcSourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:CurrentTime`, `aws:EpochTime` und `aws:ResourceTag/%s` (von zugehörigen Serverless-Caches und Benutzern) unterstützt.
 - Bei Verwendung der IAM-Authentifizierung mit Replikationsgruppen werden `aws:SourceIp` und `aws:ResourceTag/%s` (von zugehörigen Replikationsgruppen und Benutzern) unterstützt.

Weitere Informationen über globale Bedingungskontextschlüssel finden Sie unter [Globale AWS-Bedingungskontextschlüssel](#) im IAM-Benutzerhandbuch.

Einrichtung

So richten Sie die IAM-Authentifizierung ein:

1. Erstellen eines Caches

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name cache-01 \  
  --description "ElastiCache IAM auth application" \  
  --tags Key=Value
```

```
--engine redis
```

- Erstellen Sie, wie unten dargestellt, ein Dokument mit den IAM-Vertrauensrichtlinien für Ihre Rolle, sodass Ihr Konto die neue Rolle übernehmen kann. Speichern Sie die Richtlinie in einer Datei namens trust-policy.json.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
    "Action": "sts:AssumeRole"
  }
}
```

- Erstellen Sie ein IAM-Richtliniendokument wie im Folgenden dargestellt. Speichern Sie die Richtlinie in einer Datei namens policy.json.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticache:Connect"
      ],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:cache-01",
        "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"
      ]
    }
  ]
}
```

- Erstellen Sie eine IAM-Rolle.

```
aws iam create-role \  
--role-name "elasticache-iam-auth-app" \  
--assume-role-policy-document file://trust-policy.json
```

5. Erstellen Sie die IAM-Richtlinie.

```
aws iam create-policy \  
--policy-name "elasticache-allow-all" \  
--policy-document file://policy.json
```

6. Fügen Sie die IAM-Richtlinie an die Rolle an.

```
aws iam attach-role-policy \  
--role-name "elasticache-iam-auth-app" \  
--policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

7. Erstellen Sie einen neuen IAM-fähigen Benutzer.

```
aws elasticache create-user \  
--user-name iam-user-01 \  
--user-id iam-user-01 \  
--authentication-mode Type=iam \  
--engine redis \  
--access-string "on ~* +@all"
```

8. Erstellen Sie eine Benutzergruppe und fügen Sie einen Benutzer an.

```
aws elasticache create-user-group \  
--user-group-id iam-user-group-01 \  
--engine redis \  
--user-ids default iam-user-01  
  
aws elasticache modify-serverless-cache \  
--serverless-cache-name cache-01 \  
--user-group-id iam-user-group-01
```

Herstellen von Verbindungen

Verbinden mit Token als Passwort

Sie müssen zuerst das kurzlebige IAM-Authentifizierungstoken mithilfe einer [vorab signierten AWS-SigV4-Anfrage](#) generieren. Danach geben Sie das IAM-Authentifizierungstoken als Passwort an, wenn Sie eine Verbindung zu einem Valkey- oder Redis-OSS-Cache herstellen, wie im folgenden Beispiel gezeigt.

```
String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
// properties.
AWSCredentialsProvider awsCredentialsProvider = new
    DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request and signed it using the AWS credentials.
// The pre-signed request URL is used as an IAM authentication token for ElastiCache
// with Redis OSS.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
    region, isServerless);
String iamAuthToken =
    iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(userId, iamAuthToken)
    .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Im Folgenden finden Sie die Definition für IAMAuthTokenRequest.

```
public class IAMAuthTokenRequest {
    private static final HttpMethodName REQUEST_METHOD = HttpMethodName.GET;
    private static final String REQUEST_PROTOCOL = "http://";
    private static final String PARAM_ACTION = "Action";
```

```
private static final String PARAM_USER = "User";
private static final String PARAM_RESOURCE_TYPE = "ResourceType";
private static final String RESOURCE_TYPE_SERVERLESS_CACHE = "ServerlessCache";
private static final String ACTION_NAME = "connect";
private static final String SERVICE_NAME = "elasticache";
private static final long TOKEN_EXPIRY_SECONDS = 900;

private final String userId;
private final String cacheName;
private final String region;
private final boolean isServerless;

public IAMAuthTokenRequest(String userId, String cacheName, String region, boolean
isServerless) {
    this.userId = userId;
    this.cacheName = cacheName;
    this.region = region;
    this.isServerless = isServerless;
}

public String toSignedRequestUri(AWSCredentials credentials) throws
URISyntaxException {
    Request<Void> request = getSignableRequest();
    sign(request, credentials);
    return new URIBuilder(request.getEndpoint())
        .addParameters(toNamedValuePair(request.getParameters()))
        .build()
        .toString()
        .replace(REQUEST_PROTOCOL, "");
}

private <T> Request<T> getSignableRequest() {
    Request<T> request = new DefaultRequest<>(SERVICE_NAME);
    request.setHttpMethod(REQUEST_METHOD);
    request.setEndpoint(getRequestUri());
    request.addParameters(PARAM_ACTION, Collections.singletonList(ACTION_NAME));
    request.addParameters(PARAM_USER, Collections.singletonList(userId));
    if (isServerless) {
        request.addParameters(PARAM_RESOURCE_TYPE,
Collections.singletonList(RESOURCE_TYPE_SERVERLESS_CACHE));
    }
    return request;
}
```

```

private URI getRequestId() {
    return URI.create(String.format("%s%s/", REQUEST_PROTOCOL, cacheName));
}

private <T> void sign(SignableRequest<T> request, AWSCredentials credentials) {
    AWS4Signer signer = new AWS4Signer();
    signer.setRegionName(region);
    signer.setServiceName(SERVICE_NAME);

    DateTime dateTime = DateTime.now();
    dateTime = dateTime.plus(Duration.standardSeconds(TOKEN_EXPIRY_SECONDS));

    signer.presignRequest(request, credentials, dateTime.toDate());
}

private static List<NameValuePair> toNamedValuePair(Map<String, List<String>> in) {
    return in.entrySet().stream()
        .map(e -> new BasicNameValuePair(e.getKey(), e.getValue().get(0)))
        .collect(Collectors.toList());
}
}

```

Verbinden mit Anbieter von Anmeldeinformationen

Der folgende Code zeigt, wie Sie sich ElastiCache mithilfe des IAM-Anbieters für Anmeldeinformationen authentifizieren.

```

String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
// properties.
AWSCredentialsProvider awsCredentialsProvider = new
    DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request. Once this request is signed it can be
// used as an
// IAM authentication token for ElastiCache with Redis OSS.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
    region, isServerless);

```

```
// Create a Redis OSS credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
    RedisIAMAuthCredentialsProvider(
        userId, iamAuthTokenRequest, awsCredentialsProvider);

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(redisCredentialsProvider)
    .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Im Folgenden finden Sie ein Beispiel für einen Lettuce Redis OSS-Client, der den IAMAuth TokenRequest in einen Anmeldeinformationsanbieter einbindet, um bei Bedarf automatisch temporäre Anmeldeinformationen zu generieren.

```
public class RedisIAMAuthCredentialsProvider implements RedisCredentialsProvider {
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final AWSCredentialsProvider awsCredentialsProvider;
    private final String userId;
    private final IAMAuthTokenRequest iamAuthTokenRequest;
    private final Supplier<String> iamAuthTokenSupplier;

    public RedisIAMAuthCredentialsProvider(String userId,
        IAMAuthTokenRequest iamAuthTokenRequest,
        AWSCredentialsProvider awsCredentialsProvider) {
        this.userName = userId;
        this.awsCredentialsProvider = awsCredentialsProvider;
        this.iamAuthTokenRequest = iamAuthTokenRequest;
        this.iamAuthTokenSupplier =
            Suppliers.memoizeWithExpiration(this::getIamAuthToken, TOKEN_EXPIRY_SECONDS,
                TimeUnit.SECONDS);
    }

    @Override
    public Mono<RedisCredentials> resolveCredentials() {
```

```
        return Mono.just(RedisCredentials.just(userId, iamAuthTokenSupplier.get()));
    }

    private String getIamAuthToken() {
        return
        iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
    }
}
```

Authentifizierung mit dem Befehl Valkey und Redis OSS AUTH

Note

Der AUTH wurde ersetzt durch. [the section called “Rollenbasierte Zugriffskontrolle \(RBAC\)”](#)
Alle Serverless-Caches müssen RBAC für die Authentifizierung verwenden.

Valkey und Redis OSS-Authentifizierungstoken oder Passwörter ermöglichen es Valkey und Redis OSS, ein Passwort anzufordern, bevor Clients Befehle ausführen können, wodurch die Datensicherheit verbessert wird. Das AUTH ist nur für knotenbasierte Cluster verfügbar.

Themen

- [Überblick über AUTH in ElastiCache für Valkey und Redis OSS](#)
- [Anwenden der Authentifizierung auf einen ElastiCache OSS-Cluster für Valkey und Redis](#)
- [Ändern des AUTH-Tokens auf einem vorhandenen Cluster](#)
- [Migration von RBAC zu AUTH](#)

Überblick über AUTH in ElastiCache für Valkey und Redis OSS

Wenn Sie den AUTH mit Ihrem OSS-Cluster ElastiCache für Valkey und Redis verwenden, gibt es einige Verbesserungen.

Beachten Sie bei der Verwendung von AUTH insbesondere die folgenden Einschränkungen für AUTH-Token oder -Passwörter:

- Token oder Kennwörter müssen 16-128 druckbare Zeichen enthalten.
- Nicht alphanumerische Zeichen sind auf (!, &, #, \$, ^, <, >, -).

- AUTH kann nur für Valkey- oder Redis OSS-Cluster aktiviert werden, für die Verschlüsselung während der Übertragung aktiviert ist.

Um ein starkes Token einzurichten, wird empfohlen, eine strenge Passwortrichtlinie zu befolgen, z. B. die Folgende:

- Token oder Passwörter müssen mindestens drei der folgenden Zeichentypen enthalten:
 - Uppercase characters
 - Lowercase characters
 - Ziffern
 - Nicht-alphanumerische Zeichen (!, &, #, \$, ^, <, >, -)
- Token oder Passwörter dürfen kein Wörterbuchwort oder ein leicht modifiziertes Wörterbuchwort enthalten.
- Tokens oder Passwörter dürfen nicht mit einem kürzlich verwendeten Token identisch oder diesem ähnlich sein.

Anwenden der Authentifizierung auf einen ElastiCache OSS-Cluster für Valkey und Redis

Sie können verlangen, dass Benutzer ein Token (Passwort) auf einem tokengeschützten Valkey- oder Redis OSS-Server eingeben. Geben Sie dazu beim Erstellen Ihrer Replikationsgruppe oder Ihres Clusters den Parameter `--auth-token` (API: `AuthToken`) mit dem richtigen Token an. Fügen Sie ihn auch in alle nachfolgenden Befehle an die Replikationsgruppe oder den Cluster ein.

Der folgende AWS CLI Vorgang erstellt eine Replikationsgruppe mit aktivierter Verschlüsselung bei der Übertragung (TLS) und dem Token. AUTH *This-is-a-sample-token* Ersetzen Sie die Subnetzgruppe `sng-test` durch eine vorhandene Subnetzgruppe.

Hauptparameter

- **--engine**— Muss `valkey` oder `seinredis`.
- **--engine-version**— Wenn die Engine Redis OSS ist, muss es 3.2.6, 4.0.10 oder höher sein.
- **--transit-encryption-enabled** – Erforderlich für Authentifizierung und HIPAA-Berechtigung.
- **--auth-token** – Für HIPAA-Konformität erforderlich. Dieser Wert muss das richtige Token für diesen tokengeschützten Valkey- oder Redis OSS-Server sein.
- **--cache-subnet-group** – Für HIPAA-Konformität erforderlich.

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id authtestgroup \  
  --replication-group-description authtest \  
  --engine redis \  
  --cache-node-type cache.m4.large \  
  --num-node-groups 1 \  
  --replicas-per-node-group 2 \  
  --transit-encryption-enabled \  
  --auth-token This-is-a-sample-token \  
  --cache-subnet-group sng-test
```

Für Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id authtestgroup ^  
  --replication-group-description authtest ^  
  --engine redis ^  
  --cache-node-type cache.m4.large ^  
  --num-node-groups 1 ^  
  --replicas-per-node-group 2 ^  
  --transit-encryption-enabled ^  
  --auth-token This-is-a-sample-token ^  
  --cache-subnet-group sng-test
```

Ändern des AUTH-Tokens auf einem vorhandenen Cluster

Um die Aktualisierung Ihrer Authentifizierung zu vereinfachen, können Sie das in einem Cluster verwendete AUTH Token ändern. Sie können diese Änderung vornehmen, wenn die Engine-Version Valkey 7.2 oder höher oder Redis 5.0.6 oder höher ist. ElastiCache muss außerdem die Verschlüsselung bei der Übertragung aktiviert sein.

Das Ändern des Authentifizierungs-Token unterstützt zwei Strategien: ROTATE und SET. Die ROTATE-Strategie fügt dem Server ein zusätzliches AUTH-Token hinzu, wobei das vorherige Token beibehalten wird. Die SET-Strategie aktualisiert den Server so, dass er nur ein einziges AUTH-Token unterstützt. Führen Sie diese Änderungsaufrufe mit dem `--apply-immediately`-Parameter durch, um die Änderungen sofort anzuwenden.

Rotation des AUTH-Tokens

Um einen Valkey- oder Redis OSS-Server mit einem neuen AUTH-Token zu aktualisieren, rufen Sie die `ModifyReplicationGroup` API mit dem `--auth-token` Parameter als neuem AUTH-Token und dem `--auth-token-update-strategy` Wert `ROTATE` auf. Nach Abschluss der `ROTATE`-Änderung unterstützt der Cluster zusätzlich zu dem im Parameter angegebenen auch das vorherige AUTH-Token. `auth-token` Wenn vor der AUTH-Token-Rotation kein AUTH-Token für die Replikationsgruppe konfiguriert wurde, unterstützt der Cluster das im `--auth-token` Parameter angegebene AUTH-Token zusätzlich zur Unterstützung von Verbindungen ohne Authentifizierung. Informationen [Setzen des AUTH-Tokens](#) zur Aktualisierung des erforderlichen AUTH-Tokens finden Sie unter Verwendung der Aktualisierungsstrategie `SET`.

Note

Wenn Sie das AUTH-Token vorher nicht konfigurieren, unterstützt der Cluster nach Abschluss der Änderung zusätzlich zu dem AUTH-Token, das im AUTH-Token-Parameter angegeben ist, kein weiteres AUTH-Token.

Wenn diese Änderung auf einem Server durchgeführt wird, der bereits zwei AUTH-Token unterstützt, wird bei diesem Vorgang auch das älteste AUTH-Token entfernt. Auf diese Weise kann ein Server bis zu zwei aktuelle AUTH-Token gleichzeitig unterstützen.

An dieser Stelle können Sie fortfahren, indem Sie den Client so aktualisieren, dass er das neueste AUTH-Token verwendet. Nachdem die Clients aktualisiert wurden, können Sie die `SET`-Strategie für die AUTH-Token-Rotation (im folgenden Abschnitt erläutert) verwenden, um ausschließlich den neuen Token zu verwenden.

Mit dem folgenden AWS CLI Vorgang wird eine Replikationsgruppe so geändert, dass das Token rotiert wird. ***AUTH. This-is-the-rotated-token***

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
--replication-group-id authtestgroup \  
--auth-token This-is-the-rotated-token \  
--auth-token-update-strategy ROTATE \  
--apply-immediately
```

Für Windows:


```
aws elasticache modify-replication-group ^  
--replication-group-id authtestgroup ^  
--auth-token This-is-the-rotated-token ^  
--auth-token-update-strategy ROTATE ^  
--apply-immediately
```

Setzen des AUTH-Tokens

Um einen Valkey- oder Redis OSS-Server so zu aktualisieren, dass er ein einzelnes erforderliches AUTH Token unterstützt, rufen Sie den ModifyReplicationGroup API-Vorgang mit dem --auth-token Parameter auf, der denselben Wert wie das letzte AUTH-Token hat, und dem --auth-token-update-strategy Parameter mit dem Wert. SET Die SET-Strategie kann nur mit einem Cluster verwendet werden, der über 2 AUTH-Token oder 1 optionales AUTH-Token verfügt, wenn zuvor eine ROTATE-Strategie verwendet wurde. Nach Abschluss der Änderung unterstützt der Server nur das im Parameter auth-token angegebene AUTH-Token.

Mit dem folgenden AWS CLI Vorgang wird eine Replikationsgruppe geändert, auf die das AUTH-Token festgelegt werden soll. *This-is-the-set-token*

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
--replication-group-id authtestgroup \  
--auth-token This-is-the-set-token \  
--auth-token-update-strategy SET \  
--apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id authtestgroup ^  
--auth-token This-is-the-set-token ^  
--auth-token-update-strategy SET ^  
--apply-immediately
```

Aktivierung der Authentifizierung auf einem vorhandenen Cluster

Um die Authentifizierung auf einem vorhandenen Valkey- oder Redis OSS-Server zu aktivieren, rufen Sie den ModifyReplicationGroup API-Vorgang auf. Rufen Sie ModifyReplicationGroup mit

dem `--auth-token-Parameter` als neuem Token und `--auth-token-update-strategy` mit dem Wert `ROTATE` auf.

Nach Abschluss der `ROTATE`-Änderung unterstützt der Cluster das im `--auth-token` Parameter angegebene AUTH Token zusätzlich zur Unterstützung von Verbindungen ohne Authentifizierung. Sobald alle Client-Anwendungen aktualisiert wurden, um sich mit dem AUTH-Token bei Valkey oder Redis OSS zu authentifizieren, verwenden Sie die `SET`-Strategie, um das AUTH-Token als erforderlich zu markieren. Die Aktivierung der Authentifizierung wird nur auf Valkey- und Redis OSS-Servern unterstützt, auf denen Encryption in Transit (TLS) aktiviert ist.

Migration von RBAC zu AUTH

Wenn Sie Benutzer mit Valkey oder Redis OSS Role-Based Access Control (RBAC) authentifizieren, wie unter beschrieben [Rollenbasierte Zugriffskontrolle \(RBAC\)](#), und Sie zu AUTH migrieren möchten, verwenden Sie die folgenden Verfahren. Sie können entweder mit der Konsole oder mit dem CLI migrieren.

Um mithilfe der Konsole von RBAC zu AUTH zu migrieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der sich der Cluster befindet, den Sie ändern möchten.
3. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, den Sie modifizieren möchten.

Es wird eine Liste der ausgewählten Engine-Cluster angezeigt.

4. Wählen Sie in der Liste der Cluster den Namen des Clusters aus, den Sie modifizieren möchten..
5. Wählen Sie für Actions (Aktionen) die Option `Modify` (Ändern) aus.

Das Fenster `Ändern` wird angezeigt.

6. Wählen Sie für Access Control Valkey AUTH default user access oder Redis OSS AUTH default user access.
7. Legen Sie unter Valkey AUTH-Token oder Redis OSS AUTH-Token ein neues Token fest.
8. Wählen Sie Änderungen in der Vorschau anzeigen und dann auf dem nächsten Bildschirm `Ändern` aus.

Um von RBAC zu AUTH zu migrieren, verwenden Sie den AWS CLI

Verwenden Sie einen der folgenden Befehle, um ein neues optionales AUTH Token für Ihre Valkey- oder Redis OSS-Replikationsgruppe zu konfigurieren. Beachten Sie, dass ein optionales Auth-Token einen nicht authentifizierten Zugriff auf die Replikationsgruppe ermöglicht, bis das Auth-Token als erforderlich gekennzeichnet ist. Verwenden Sie dabei die Aktualisierungsstrategie im folgenden Schritt. SET

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test \  
  --remove-user-groups \  
  --auth-token This-is-a-sample-token \  
  --auth-token-update-strategy ROTATE \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test ^  
  --remove-user-groups ^  
  --auth-token This-is-a-sample-token ^  
  --auth-token-update-strategy ROTATE ^  
  --apply-immediately
```

Nachdem Sie den obigen Befehl ausgeführt haben, können Sie Ihre Valkey- oder Redis OSS-Anwendungen aktualisieren, um sich mit dem neu konfigurierten optionalen AUTH-Token bei der ElastiCache Replikationsgruppe zu authentifizieren. Verwenden Sie die Aktualisierungsstrategie SET im nachfolgenden Befehl unten, um die Auth-Token-Rotation abzuschließen. Dadurch wird das optionale AUTH-Token als erforderlich markiert. Wenn die Aktualisierung des Auth-Tokens abgeschlossen ist, wird der Status der Replikationsgruppe als angezeigt ACTIVE und für alle Verbindungen zu dieser Replikationsgruppe ist eine Authentifizierung erforderlich.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test \  
  --auth-token This-is-a-sample-token \  
  --auth-token-update-strategy SET \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test ^  
  --remove-user-groups ^  
  --auth-token This-is-a-sample-token ^  
  --auth-token-update-strategy SET ^  
  --apply-immediately
```

Weitere Informationen finden Sie unter [Authentifizierung mit dem Befehl Valkey und Redis OSS AUTH](#).

Note

Wenn Sie die Zugriffskontrolle auf einem ElastiCache Cluster deaktivieren müssen, finden Sie weitere Informationen unter [the section called “Deaktivieren der Zugriffskontrolle auf einem ElastiCache Valkey- oder Redis-OSS-Cache”](#).

Deaktivieren der Zugriffskontrolle auf einem ElastiCache Valkey- oder Redis-OSS-Cache

Folgen Sie den nachstehenden Anweisungen, um die Zugriffskontrolle für einen TLS-fähigen Valkey- oder Redis OSS-Cache zu deaktivieren. Ihr Cache wird eine von zwei verschiedenen Konfigurationstypen haben: AUTH-Standardbenutzerzugriff oder Benutzergruppen-Zugriffskontrollliste (RBAC). Wenn Ihr Cache mit der AUTH-Konfiguration erstellt wurde, müssen Sie ihn auf die RBAC-Konfiguration ändern, bevor Sie den Cache deaktivieren können, indem Sie die Benutzergruppen entfernen. Wenn Ihr Cache mit der RBAC-Konfiguration erstellt wurde, können Sie ihn direkt deaktivieren.

Um einen mit RBAC konfigurierten serverlosen Valkey- oder Redis OSS-Cache zu deaktivieren

1. Entfernen Sie die Benutzergruppen, um die Zugriffskontrolle zu deaktivieren.

```
aws elasticache modify-serverless-cache --serverless-cache-name <serverless-cache>  
  --remove-user-group
```

2. (Optional) Stellen Sie sicher, dass dem Serverless-Cache keine Benutzergruppen zugeordnet sind.

```
aws elasticache describe-serverless-caches --serverless-cache-name <serverless-cache>
{
  "...
  "UserGroupId": ""
  "...
}
```

Um einen Valkey- oder Redis OSS-Cache zu deaktivieren, der mit einem AUTH-Token konfiguriert ist

1. Ändern Sie das AUTH-Token in RBAC und geben Sie eine Benutzergruppe an, die hinzugefügt werden soll.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-id-value> --auth-token-update-strategy DELETE --user-group-ids-to-add <user-group-value>
```

2. Stellen Sie sicher, dass das AUTH-Token deaktiviert und eine Benutzergruppe hinzugefügt wurde.

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-id-value>
{
  "...
  "AuthTokenEnabled": false,
  "UserGroupIds": [
    "<user-group-value>"
  ]
  "...
}
```

3. Entfernen Sie die Benutzergruppen, um die Zugriffskontrolle zu deaktivieren.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-value> --user-group-ids-to-remove <user-group-value>
{
  "...
  "PendingModifiedValues": {
    "UserGroups": {
      "UserGroupIdsToAdd": [],
```

```

    "UserGroupIdsToRemove": [
        "<user-group-value>"
    ]
}
"..."
}

```

4. (Optional) Stellen Sie sicher, dass dem Cluster keine Benutzergruppen zugeordnet sind. Das Feld `AuthTokenEnabled` sollte auch „false“ lauten.

```

aws elasticache describe-replication-groups --replication-group-id <replication-
group-value>
"AuthTokenEnabled": false

```

Um einen mit RBAC konfigurierten Valkey- oder Redis OSS-Cluster zu deaktivieren

1. Entfernen Sie die Benutzergruppen, um die Zugriffskontrolle zu deaktivieren.

```

aws elasticache modify-replication-group --replication-group-id <replication-group-
value> --user-group-ids-to-remove <user-group-value>
{
    "..."
    "PendingModifiedValues": {
        "UserGroups": {
            "UserGroupIdsToAdd": [],
            "UserGroupIdsToRemove": [
                "<user-group-value>"
            ]
        }
    }
    "..."
}

```

2. (Optional) Stellen Sie sicher, dass dem Cluster keine Benutzergruppen zugeordnet sind. Das Feld `AuthTokenEnabled` sollte auch „false“ lauten.

```

aws elasticache describe-replication-groups --replication-group-id <replication-
group-value>
"AuthTokenEnabled": false

```

Richtlinie für den Datenverkehr zwischen Netzwerken

Amazon ElastiCache verwendet die folgenden Techniken, um Ihre Cache-Daten zu sichern und vor unbefugtem Zugriff zu schützen:

- [Amazon VPCs und ElastiCache Sicherheit](#) erklärt den Typ der Sicherheitsgruppe, die Sie für Ihre Installation benötigen.
- [Identity and Access Management für Amazon ElastiCache](#) Verwenden Sie zum Erteilen und Beschränken von Aktionen von Benutzern, Gruppen und Rollen.

Themen

- [Amazon VPCs und ElastiCache Sicherheit](#)
- [ElastiCache API- und Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#)
- [Subnetze und Subnetzgruppen](#)

Amazon VPCs und ElastiCache Sicherheit

Da die Sicherheit der Daten wichtig ist, stellt ElastiCache Methoden zur Verfügung, mit denen Sie kontrollieren können, wer Zugriff auf Ihre Daten hat. Wie Sie den Zugriff auf Ihre Daten kontrollieren, hängt davon ab, ob Sie Ihre Cluster in einer Amazon Virtual Private Cloud (Amazon VPC) oder Amazon EC2 -Classic gestartet haben oder nicht.

Important

Wir haben die Verwendung von Amazon EC2 -Classic zum Starten von Clustern eingestellt. ElastiCache Alle aktuellen Generierungsknoten werden ausschließlich in Amazon Virtual Private Cloud gestartet.

Der Amazon Virtual Private Cloud (Amazon VPC)-Service definiert ein virtuelles Netzwerk, das einem herkömmlichen Rechenzentrum sehr ähnlich ist. Wenn Sie Ihre Amazon VPC konfigurieren, können Sie den IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen, Netzwerk-Gateways und Sicherheitseinstellungen konfigurieren. Sie können dem virtuellen Netzwerk außerdem einen Cluster hinzufügen und den Zugriff auf den Cluster über Amazon-VPC-Sicherheitsgruppen steuern.

In diesem Abschnitt wird erklärt, wie Sie einen ElastiCache Cluster in einer Amazon VPC manuell konfigurieren. Diese Informationen richten sich an Benutzer, die ein tieferes Verständnis dafür wünschen, wie Amazon VPC ElastiCache und Amazon VPC zusammenarbeiten.

Themen

- [Understanding ElastiCache und Amazon VPCs](#)
- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#)
- [Erstellen einer Virtual Private Cloud \(VPC\)](#)
- [Herstellen einer Verbindung mit einem Cache, der in einer Amazon VPC ausgeführt wird](#)

Understanding ElastiCache und Amazon VPCs

ElastiCache ist vollständig in die Amazon Virtual Private Cloud (Amazon VPC) integriert. Für ElastiCache Benutzer bedeutet das Folgendes:

- Wenn Ihr AWS Konto nur die EC2 -VPC-Plattform unterstützt, wird Ihr Cluster ElastiCache immer in einer Amazon VPC gestartet.
- Wenn Sie noch nicht damit vertraut sind AWS, werden Ihre Cluster in einer Amazon VPC bereitgestellt. Eine Standard-VPC wird automatisch für Sie erstellt.
- Wenn Sie eine Standard-VPC haben und beim Starten eines Clusters kein Subnetz angeben, wird der Cluster in Ihrer Standard-Amazon-VPC gestartet.

Weitere Informationen finden Sie unter [Detecting Your Supported Platforms and Whether You Have a Default VPC](#).

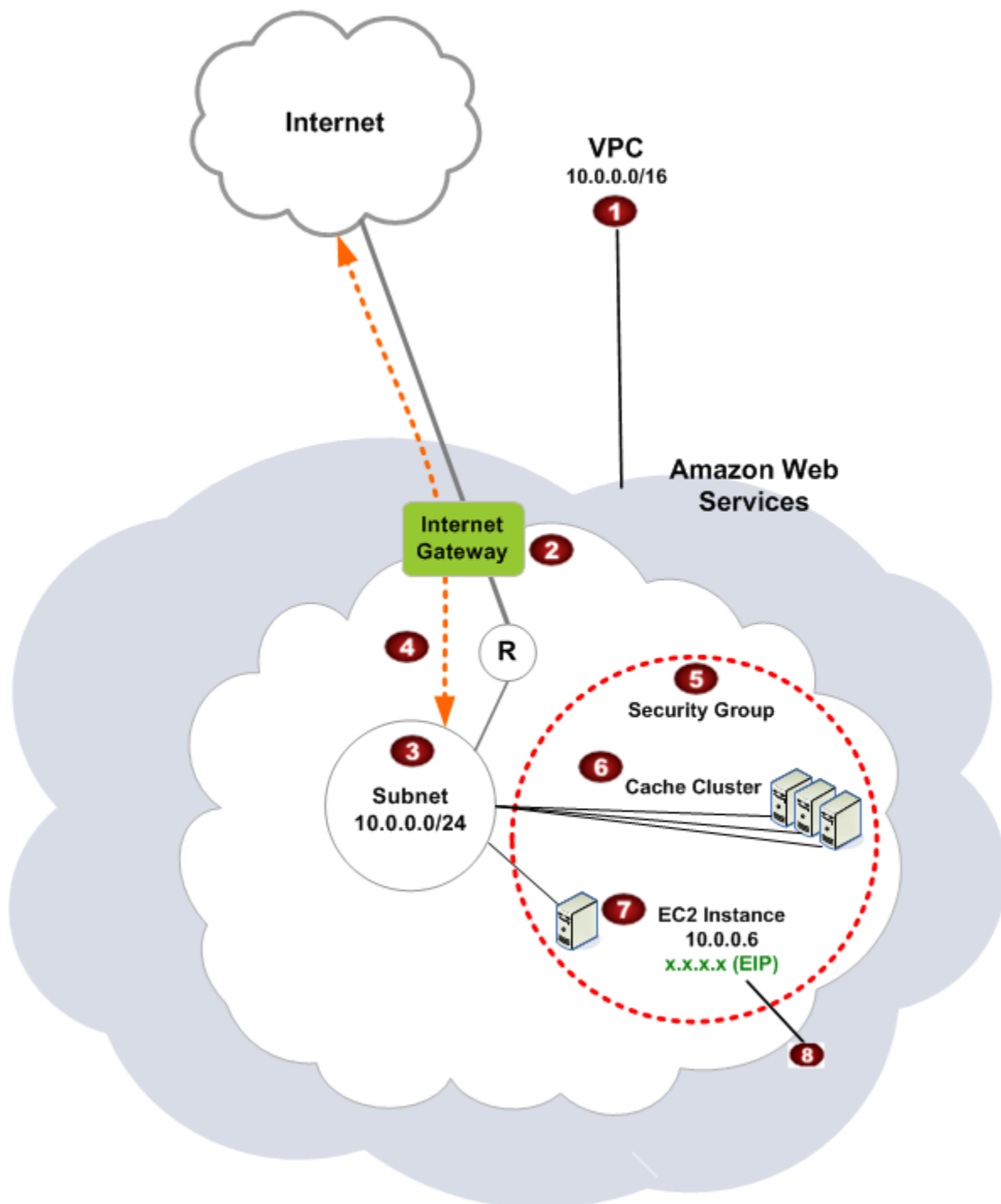
Mit Amazon Virtual Private Cloud können Sie ein virtuelles Netzwerk in der AWS Cloud erstellen, das einem herkömmlichen Rechenzentrum sehr ähnlich ist. Sie können Ihre Amazon VPC konfigurieren und den IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen, Netzwerk-Gateways und Sicherheitseinstellungen konfigurieren.

Die grundlegende Funktionalität von ElastiCache ist dieselbe wie in einer Virtual Private Cloud; ElastiCache verwaltet Software-Upgrades, Patches, Fehlererkennung und Wiederherstellung, unabhängig davon, ob Ihre Cluster innerhalb oder außerhalb einer Amazon VPC bereitgestellt werden.

ElastiCache Cache-Knoten, die außerhalb einer Amazon VPC bereitgestellt werden, erhalten eine IP-Adresse, in die der endpoint/DNS Name aufgelöst wird. Dies bietet Konnektivität von Amazon Elastic Compute Cloud (Amazon EC2) -Instances. Wenn Sie einen ElastiCache Cluster in einem privaten Amazon VPC-Subnetz starten, wird jedem Cache-Knoten eine private IP-Adresse innerhalb dieses Subnetzes zugewiesen.

Überblick über ElastiCache in einer Amazon VPC

Das folgende Diagramm und die folgende Tabelle beschreiben die Amazon VPC-Umgebung zusammen mit ElastiCache Clustern und EC2 Amazon-Instances, die in der Amazon VPC gestartet werden.

**1**

Die Amazon VPC ist ein isolierter Teil der AWS Cloud, dem ein eigener Block von IP-Adressen zugewiesen wurde.

2

Ein Internet-Gateway verbindet Ihre Amazon VPC direkt mit dem Internet und bietet Zugriff auf andere AWS Ressourcen wie Amazon Simple Storage Service (Amazon S3), die außerhalb Ihrer Amazon VPC laufen.

3

Ein Amazon VPC-Subnetz ist ein Segment des IP-Adressbereichs einer Amazon VPC, in dem Sie AWS Ressourcen entsprechend Ihren Sicherheits- und Betriebsanforderungen isolieren können.

4

Eine Routing-Tabelle in der Amazon VPC steuert den Netzwerkdatenverkehr zwischen dem Subnetz und dem Internet. Die Amazon VPC hat einen impliziten Router, der in diesem Diagramm durch den Kreis mit dem R dargestellt wird.

5

Eine Amazon VPC-Sicherheitsgruppe kontrolliert den ein- und ausgehenden Datenverkehr für Ihre ElastiCache Cluster und Amazon-Instances. EC2

6

Sie können einen ElastiCache Cluster im Subnetz starten. Die Cache-Knoten haben private IP-Adressen aus dem Adressbereich des Subnetzes.

7

Sie können EC2 Amazon-Instances auch im Subnetz starten. Jede EC2 Amazon-Instance hat eine private IP-Adresse aus dem Adressbereich des Subnetzes. Die EC2 Amazon-Instance kann eine Verbindung zu jedem Cache-Knoten im selben Subnetz herstellen.

8

Damit eine EC2 Amazon-Instance in Ihrer Amazon VPC über das Internet erreichbar ist, müssen Sie der Instance eine statische, öffentliche Adresse zuweisen, die als Elastic IP-Adresse bezeichnet wird.

Voraussetzungen

Um einen ElastiCache Cluster innerhalb einer Amazon VPC zu erstellen, muss Ihre Amazon VPC die folgenden Anforderungen erfüllen:

- Die Amazon VPC muss nicht dedizierte EC2 Amazon-Instances zulassen. Sie können es nicht ElastiCache in einer Amazon VPC verwenden, die für Dedicated Instance Tenancy konfiguriert ist.

- Für Ihre Amazon VPC muss eine Cache-Subnetzgruppe definiert werden. ElastiCache verwendet diese Cache-Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren VPC-Endpunkten oder Cache-Knoten zugeordnet werden sollen.
- Die CIDR-Blöcke für jedes Subnetz müssen groß genug sein, um Reserve-IP-Adressen für Wartungsarbeiten bereitzustellen. ElastiCache

Weiterleitung und Sicherheit

Sie können Weiterleitung in Ihrer Amazon VPC so konfigurieren, dass gesteuert wird, wo der Datenverkehr fließt (z. B. zum Internet-Gateway oder Virtual Private Gateway). Mit einem Internet-Gateway hat Ihre Amazon VPC direkten Zugriff auf andere AWS Ressourcen, die nicht in Ihrer Amazon VPC laufen. Wenn Sie sich nur für ein virtuelles privates Gateway mit einer Verbindung zum lokalen Netzwerk Ihrer Organisation entscheiden, können Sie Ihren Internet-Datenverkehr über das VPN leiten und lokale Sicherheitsrichtlinien und eine Firewall verwenden, um ausgehenden Datenverkehr zu steuern. In diesem Fall fallen zusätzliche Bandbreitengebühren an, wenn Sie über das Internet auf AWS Ressourcen zugreifen.

Sie können Amazon VPC-Sicherheitsgruppen verwenden, um die ElastiCache Cluster und EC2 Amazon-Instances in Ihrer Amazon VPC zu sichern. Sicherheitsgruppen wirken wie eine Firewall auf der Instance-Ebene, nicht auf der Subnetzebene.

Note

Es wird jedoch dringend empfohlen, DNS-Namen für die Verbindung mit Ihren Cache-Knoten zu verwenden, da sich die zugrunde liegende IP-Adresse ändern kann.

Amazon VPC-Dokumentation

Amazon VPC hat eine eigene Dokumentation, in der das Erstellen und Nutzen Ihrer erklärt wird. In der folgenden Tabelle sind Links zu den Amazon-VPC-Handbüchern aufgeführt.

Beschreibung	Dokumentation
Erste Schritte bei der Verwendung von Amazon VPC	Erste Schritte mit Amazon VPC

Beschreibung	Dokumentation
So verwenden Sie Amazon VPC über AWS-Managementkonsole	Amazon VPC User Guide
Vollständige Beschreibungen aller Amazon-VP C-Befehle	EC2 Amazon-Befehlszeilenreferenz (die Amazon VPC-Befehle finden Sie in der EC2 Amazon-Referenz)
Vollständige Beschreibungen der Amazon-VP C-API-Aktionen, -Datentypen und -Fehler	EC2 Amazon-Befehlszeilenreferenz (die Amazon VPC-API-Operationen finden Sie in der EC2 Amazon-Referenz)
Informationen für den Netzwerkadministrator, der das Gateway an Ihrem Ende einer optionalen IPsec VPN-Verbindung konfigurieren muss	Was ist AWS Site-to-Site VPN?

Weitere Informationen zur Amazon Virtual Private Cloud finden Sie unter [Amazon Virtual Private Cloud](#).

Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC

Amazon ElastiCache unterstützt die folgenden Szenarien für den Zugriff auf einen Cache in einer Amazon VPC:

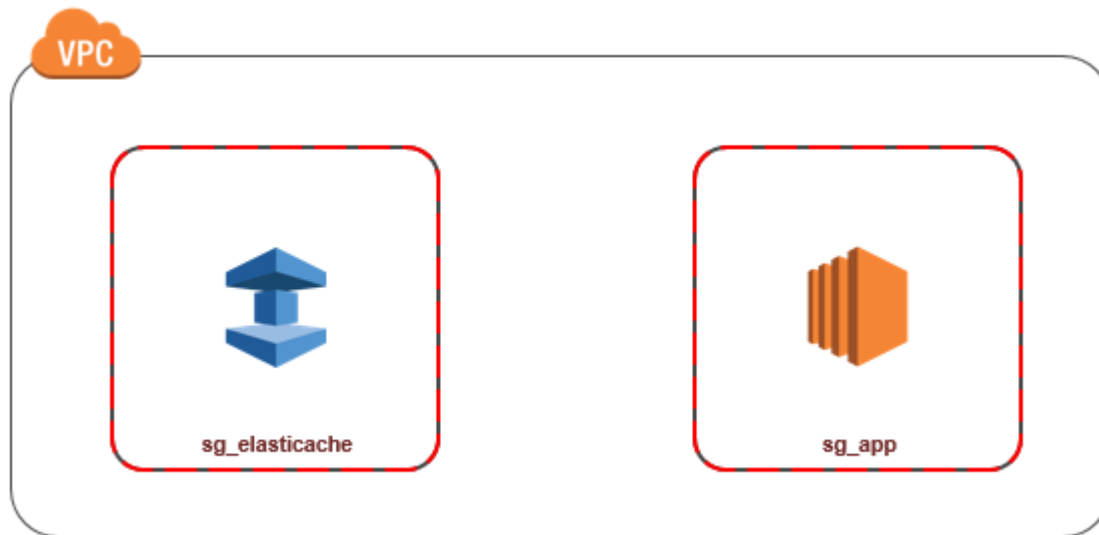
Inhalt

- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in derselben Amazon-VPC befinden](#)
- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon befinden VPCs](#)
 - [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon VPCs in derselben Region befinden](#)
 - [Verwenden von Transit Gateway](#)
 - [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in unterschiedlichen VPCs Amazon-Regionen befinden](#)
 - [Verwenden von Transit VPC](#)
- [Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden ausgeführt wird](#)
 - [Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mithilfe von VPN-Konnektivität ausgeführt wird](#)
 - [Zugreifen auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mit Direct Connect ausgeführt wird](#)

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in derselben Amazon-VPC befinden

Der häufigste Anwendungsfall ist, wenn eine auf einer EC2 Instance bereitgestellte Anwendung eine Verbindung zu einem Cache in derselben VPC herstellen muss.

Das folgende Diagramm veranschaulicht dieses Szenario.



Der einfachste Weg, den Zugriff zwischen EC2 Instances und Caches in derselben VPC zu verwalten, ist wie folgt:

1. Erstellen Sie eine Sicherheitsgruppe für Ihren Cache. Diese Sicherheitsgruppe kann verwendet werden, um den Zugriff auf den Cache zu beschränken. Sie können für diese Sicherheitsgruppe beispielsweise eine benutzerdefinierte Regel erstellen, die TCP-Zugriff über den Port, den Sie dem Cache bei seiner Erstellung zugeordnet haben, und eine IP-Adresse gewährt, mit der Sie auf den Cache zugreifen.

Der Standard-Port für Memcached-Caches ist 11211.

Der Standardport für Valkey- und Redis OSS-Caches ist 6379

2. Erstellen Sie eine VPC-Sicherheitsgruppe für Ihre EC2 Instances (Web- und Anwendungsserver). Diese Sicherheitsgruppe kann bei Bedarf den Zugriff auf die EC2 Instance aus dem Internet über die Routingtabelle der VPC ermöglichen. Sie können beispielsweise Regeln für diese Sicherheitsgruppe festlegen, um den TCP-Zugriff auf die EC2 Instance über Port 22 zu ermöglichen.
3. Erstellen Sie in der Sicherheitsgruppe für Ihren Cache benutzerdefinierte Regeln, die Verbindungen von der Sicherheitsgruppe aus zulassen, die Sie für Ihre EC2 Instances erstellt haben. Damit wird jedem Mitglied der Sicherheitsgruppe der Zugriff auf die Caches gestattet.

Note

Wenn Sie planen, [Local Zones](#) zu nutzen, stellen Sie sicher, dass Sie sie aktiviert haben. Wenn Sie eine Subnetzgruppe in dieser Local Zone erstellen, wird Ihre VPC auf diese Local Zone erweitert, und Ihre VPC behandelt das Subnetz wie jedes Subnetz in einer anderen Availability Zone. Alle relevanten Gateways und Routing-Tabellen werden automatisch angepasst.

So erstellen Sie eine Regel in einer VPC-Sicherheitsgruppe, die Verbindungen über eine andere Sicherheitsgruppe zulässt

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc>.
2. Wählen Sie im Navigationsbereich Security Groups (Sicherheitsgruppen) aus.
3. Wählen Sie eine Sicherheitsgruppe aus oder erstellen Sie eine, die Sie für Ihren Cache verwenden werden. Wählen Sie unter Inbound Rules (Eingangsregeln) die Option Edit Inbound Rules (Eingangsregeln bearbeiten) und dann Add Rule (Regeln hinzufügen). Diese Sicherheitsgruppe gewährt Mitgliedern einer anderen Sicherheitsgruppe Zugriff.
4. Wählen Sie für Type die Option Custom TCP Rule aus.
 - a. Geben Sie für Port-Bereich den Port an, den Sie beim Erstellen des Caches verwendet haben.

Der Standard-Port für Memcached-Caches ist 11211.

Der Standardport für Valkey- und Redis OSS-Caches und Replikationsgruppen ist 6379

- b. Geben Sie in das Feld Source die ersten Zeichen der ID der Sicherheitsgruppe ein. Wählen Sie aus der Liste die Sicherheitsgruppe aus, die Sie für Ihre EC2 Amazon-Instances verwenden möchten.
5. Wählen Sie Save, wenn Sie fertig sind.

Inbound rules (3)									
	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	
☐	-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-	
☐	-	sg-...	-	Custom TCP	TCP	6379	sg-... / default	-	

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon befinden VPCs

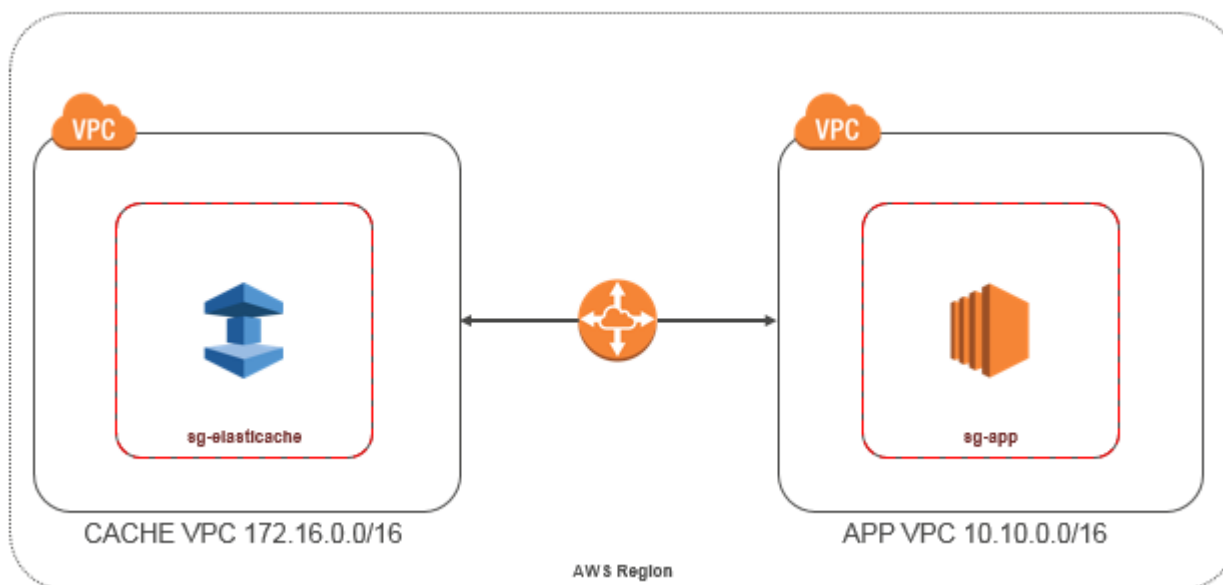
Wenn sich Ihr Cache in einer anderen VPC befindet als die EC2 Instance, die Sie für den Zugriff verwenden, gibt es mehrere Möglichkeiten, auf den Cache zuzugreifen. Wenn sich der Cache und die EC2 Instance in verschiedenen, VPCs aber in derselben Region befinden, können Sie VPC-Peering verwenden. Wenn sich der Cache und die EC2 Instance in unterschiedlichen Regionen befinden, können Sie VPN-Konnektivität zwischen Regionen herstellen.

Themen

- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon VPCs in derselben Region befinden](#)
- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in unterschiedlichen VPCs Amazon-Regionen befinden](#)

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon VPCs in derselben Region befinden

Das folgende Diagramm zeigt den Zugriff auf einen Cache durch eine EC2 Amazon-Instance in einer anderen Amazon VPC in derselben Region mithilfe einer Amazon VPC-Peering-Verbindung.



Cache, auf den eine EC2 Amazon-Instance in einer anderen Amazon VPC innerhalb derselben Region zugreift — VPC Peering Connection

Eine VPC-Peering-Verbindung ist eine Netzwerkverbindung zwischen zwei VPCs , mit der Sie den Verkehr zwischen ihnen mithilfe privater IP-Adressen weiterleiten können. Instances in jeder der VPCs können so miteinander kommunizieren, als befänden sie sich im selben Netzwerk. Sie können eine VPC-Peering-Verbindung zwischen Ihrem eigenen Amazon VPCs oder mit einer Amazon-VPC in einem anderen AWS Konto innerhalb einer einzelnen Region herstellen. Weitere Informationen zum Amazon-VPC-Peering finden Sie in der [VPC-Dokumentation](#)

Note

Je nach den auf die ElastiCache VPC angewendeten Konfigurationen kann die DNS-Namensauflösung für Peered VPCs fehlschlagen. Um dieses Problem zu lösen, VPCs müssen beide Optionen für DNS-Hostnamen und DNS-Auflösung aktiviert sein. Weitere Informationen finden Sie unter [Aktivieren einer DNS-Auflösung für eine VPC-Peering-Verbindung](#).

So greifen Sie auf einen Cache in einer anderen Amazon VPC über Peering zu

1. Stellen Sie sicher, dass die beiden VPCs keinen überlappenden IP-Bereich haben, da Sie sie sonst nicht miteinander verbinden können.
2. Schauen Sie sich die beiden VPCs an. Weitere Informationen finden Sie unter [Erstellen und Akzeptieren einer Amazon-VPC-Peering-Verbindung](#).
3. Aktualisieren Sie Ihre Routing-Tabelle. Weitere Informationen finden Sie unter [Aktualisieren der Routing-Tabellen für eine VPC-Peering-Verbindung](#).

Es folgt eine Darstellung der Routing-Tabellen für das Beispiel im voranstehenden Diagramm. Es ist zu beachten, dass pcx-a894f1c1 eine Peering-Verbindung ist.

Destination	Target	Destination	Target
172.16.0.0/16	local	10.10.0.0/16	local
10.10.0.0/16	pcx-a894f1c1	0.0.0.0/0	igw-bfdcccd8
		172.16.0.0/16	pcx-a894f1c1

VPC-Routing-Tabelle

4. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches, um eingehende Verbindungen von der Anwendungssicherheitsgruppe in der Peering-VPC zuzulassen. Weitere Informationen finden Sie unter [Verweisen auf Peer-VPC-Sicherheitsgruppen](#).

Beim Zugriff auf einen Cache über eine Peering-Verbindung fallen zusätzliche Datenübertragungskosten an.

Verwenden von Transit Gateway

Ein Transit-Gateway ermöglicht es Ihnen, Verbindungen VPCs und VPN-Verbindungen in derselben AWS Region herzustellen und den Verkehr zwischen ihnen weiterzuleiten. Ein Transit-Gateway funktioniert AWS kontenübergreifend, und Sie können AWS Resource Access Manager verwenden, um Ihr Transit-Gateway mit anderen Konten zu teilen. Nachdem Sie ein Transit-Gateway mit einem anderen AWS Konto gemeinsam genutzt haben VPCs , kann der Kontoinhaber dieses Konto mit Ihrem Transit-Gateway verknüpfen. Benutzer in einem der Konten können die Anhang jederzeit löschen.

Sie können Multicast auf einem Transit Gateway aktivieren und dann eine Transit Gateway-Multicast-Domain erstellen, mit der Multicast-Datenverkehr von der Multicast-Quelle über VPC-Anhängen, die Sie der Domain zuordnen, an Multicast-Gruppenmitglieder gesendet werden kann.

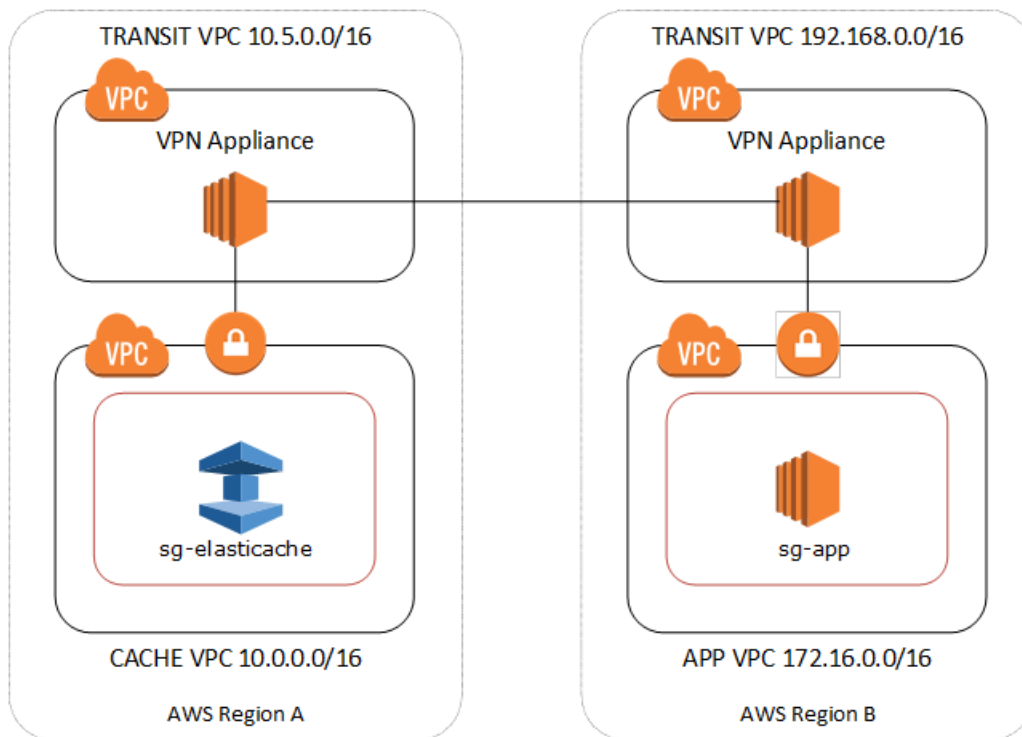
Sie können auch einen Peering-Verbindungsanhang zwischen Transit-Gateways in verschiedenen AWS Regionen erstellen. Auf diese Weise können Sie den Datenverkehr zwischen den Anhängen der Transit Gateways über verschiedene Regionen hinweg leiten.

Weitere Informationen finden Sie unter [Transit Gateways](#).

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in unterschiedlichen VPCs Amazon-Regionen befinden

Verwenden von Transit VPC

Eine Alternative zur Verwendung von VPC-Peering, eine weitere gängige Strategie für die Verbindung mehrerer, geografisch verteilter VPCs und entfernter Netzwerke, ist die Einrichtung einer Transit-VPC, die als globales Netzwerk-Transitzentrum dient. Eine Transit-VPC vereinfacht die Netzwerkverwaltung und minimiert die Anzahl der Verbindungen, die für die Verbindung mehrerer VPCs und entfernter Netzwerke erforderlich sind. Dieses Design kann Zeit und Aufwand verringern und auch Kosten reduzieren, da es virtuell ohne die herkömmlichen Ausgaben implementiert wird, die beim Einrichten einer physischen Präsenz in einem Co-Location-Transit-Hub oder beim Bereitstellen physischer Netzwerkausstattung anfallen.



Verbindungen zwischen VPCs verschiedenen Regionen herstellen

Sobald die Transit Amazon VPC eingerichtet ist, kann eine Anwendung, die in einer „Spoke“ VPC in einer Region bereitgestellt wird, eine Verbindung zu einem ElastiCache Cache in einer „Spoke“ VPC in einer anderen Region herstellen.

So greifen Sie auf einen Cache in einer anderen VPC in einer anderen AWS Region zu

1. Stellen Sie eine Transit-VPC-Lösung bereit. Weitere Informationen finden Sie unter [AWS - Transit-Gateway](#).
2. Aktualisieren Sie die VPC-Routing-Tabellen in der App und im Cache VPCs , um den Datenverkehr über das VGW (Virtual Private Gateway) und die VPN-Appliance weiterzuleiten. Im Falle des dynamischen Routing mit Border Gateway Protocol (BGP) werden Ihre Routen möglicherweise automatisch gefüllt.
3. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches, um eingehende Verbindungen aus dem IP-Bereich der Anwendungsinstanzen zuzulassen. Beachten Sie, dass Sie in diesem Szenario nicht auf die Sicherheitsgruppe des Anwendungsservers verwiesen können.

Beim regionsübergreifenden Zugriff auf einen Cache entstehen Netzwerklatenzen und fallen zusätzliche, regionsübergreifende Datenübertragungskosten an.

Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden ausgeführt wird

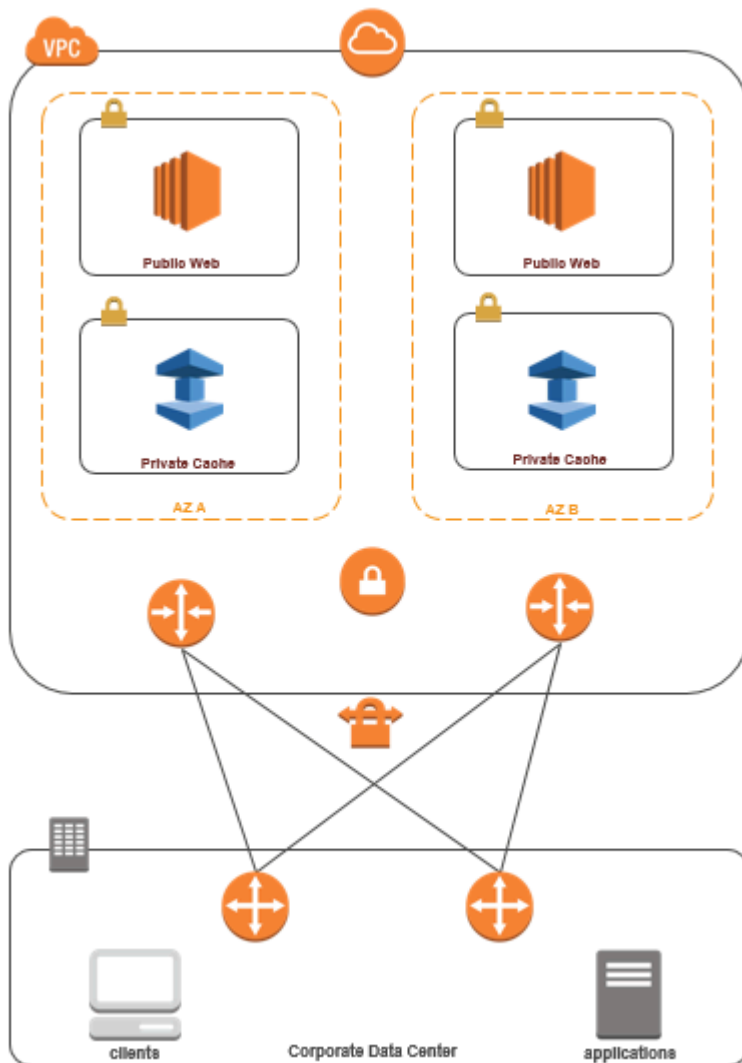
Ein anderes mögliches Szenario ist eine Hybridarchitektur, bei der Clients oder Anwendungen im Rechenzentrum des Kunden möglicherweise auf einen ElastiCache Cache in der VPC zugreifen müssen. Dieses Szenario wird auch unterstützt, vorausgesetzt, dass entweder über VPN oder Direct Connect Konnektivität zwischen der VPC des Kunden und dem Rechenzentrum besteht.

Themen

- [Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mithilfe von VPN-Konnektivität ausgeführt wird](#)
- [Zugreifen auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mit Direct Connect ausgeführt wird](#)

Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mithilfe von VPN-Konnektivität ausgeführt wird

Das folgende Diagramm zeigt den Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die in Ihrem Unternehmensnetzwerk ausgeführt wird, mithilfe von VPN-Verbindungen.



Über ein VPN wird eine Verbindung zu ElastiCache Ihrem Rechenzentrum hergestellt

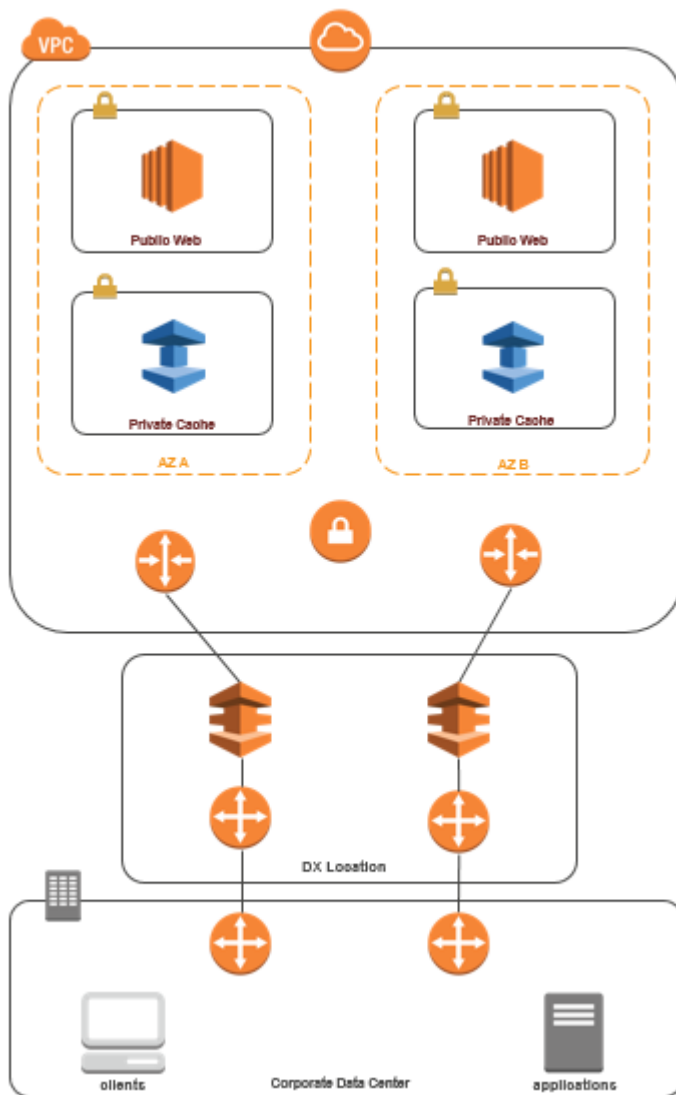
So greifen Sie auf einen Cache in einer VPC von einer Anwendung vor Ort über eine VPN-Verbindung zu

1. Richten Sie VPN-Konnektivität ein, indem Sie ein Hardware Virtual Private Gateway zu Ihrer VPC hinzufügen. Weitere Informationen finden Sie unter [Hinzufügen eines Hardware Virtual Private Gateway zu Ihrer VPC](#).
2. Aktualisieren Sie die VPC-Routentabelle für das Subnetz, in dem Ihr ElastiCache Cache bereitgestellt wird, um Datenverkehr von Ihrem lokalen Anwendungsserver zuzulassen. Im Falle des dynamischen Routing mit BGP werden Ihre Routen möglicherweise automatisch gefüllt.
3. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches, um eingehende Verbindungen von den lokalen Anwendungsservern zuzulassen.

Beim Zugriff auf einen Cache über eine VPN-Verbindung entstehen Netzwerklatenzen und fallen zusätzliche Datenübertragungskosten an.

Zugreifen auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mit Direct Connect ausgeführt wird

Das folgende Diagramm veranschaulicht den Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die in Ihrem Unternehmensnetzwerk mit Direct Connect ausgeführt wird.



Über Direct Connect eine Verbindung zu ElastiCache Ihrem Rechenzentrum herstellen

So greifen Sie mit Direct Connect von einer in Ihrem Netzwerk ausgeführten Anwendung auf einen ElastiCache Cache zu

1. Richten Sie Direct Connect-Konnektivität ein. Weitere Informationen finden Sie unter [Erste Schritte mit AWS Direct Connect](#).
2. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches, um eingehende Verbindungen von den lokalen Anwendungsservern zuzulassen.

Beim Zugriff auf einen Cache über eine DX-Verbindung können Netzwerklatenzen entstehen und zusätzliche Datenübertragungskosten anfallen.

Erstellen einer Virtual Private Cloud (VPC)

In diesem Beispiel erstellen Sie eine Amazon VPC mit einem privaten Subnetz für jede Availability Zone.

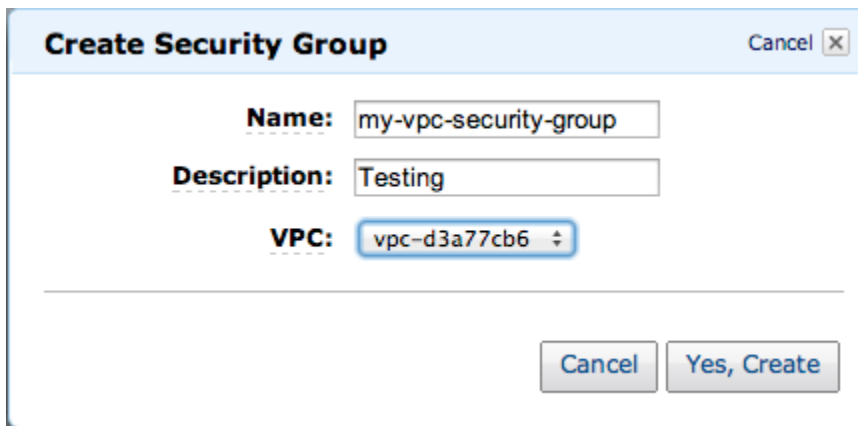
Erstellen einer Amazon VPC (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie auf dem VPC-Dashboard Create VPC (VPC erstellen) aus.
3. Wählen Sie unter Resources to create (Zu erstellende Ressourcen) die Option VPC and more (VPC und mehr) aus.
4. Wählen Sie unter Anzahl der Availability Zones (AZs) die Anzahl der Availability Zones aus, in denen Sie Ihre Subnetze starten möchten.
5. Wählen Sie unter Number of public subnets (Anzahl der öffentlichen Subnetze) die Anzahl der öffentlichen Subnetze aus, die Sie zu Ihrer VPC hinzufügen möchten.
6. Wählen Sie unter Number of private subnets (Anzahl der privaten Subnetze) die Anzahl der privaten Subnetze aus, die Sie zu Ihrer VPC hinzufügen möchten.

Tip

Notieren Sie sich Ihre Subnetz-IDs und welches öffentlich und welches privat ist. Sie benötigen diese Informationen später, wenn Sie Ihre Cluster starten und Ihrer Amazon VPC eine EC2 Amazon-Instance hinzufügen.

7. Erstellen Sie eine Amazon-VPC-Sicherheitsgruppe. Sie werden diese Gruppe für Ihren Cluster und Ihre EC2 Amazon-Instance verwenden.
 - a. Wählen Sie im Navigationsbereich der Amazon-VPC-Managementkonsole Sicherheitsgruppen.
 - b. Wählen Sie Sicherheitsgruppen erstellen aus.
 - c. Geben Sie in den entsprechenden Feldern einen Namen und eine Beschreibung für Ihre Sicherheitsgruppe ein. Wählen Sie im Feld VPC die ID für Ihre Amazon VPC aus.



- d. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Ja, erstellen aus.
8. Definieren Sie eine Netzwerkeingangsregel für Ihre Sicherheitsgruppe. Diese Regel ermöglicht es Ihnen, über Secure Shell (SSH) eine Verbindung zu Ihrer EC2 Amazon-Instance herzustellen.
- a. Wählen Sie in der Navigationsliste Security Groups aus.
 - b. Suchen Sie Ihre Sicherheitsgruppe in der Liste und wählen Sie sie aus.
 - c. Wählen Sie unter Security Group die Registerkarte Inbound aus. Wählen Sie im Feld Create a new rule die Option SSH und anschließend Add Rule aus.
 - d. Stellen Sie die folgenden Werte für Ihre neue eingehende Regel ein, um HTTP-Zugriff zuzulassen:
 - Typ: HTTP
 - Quelle: 0.0.0.0/0

Wählen Sie Apply Rule Changes aus.

Jetzt sind Sie bereit, eine Cache-Subnetzgruppe zu erstellen und einen Cluster in Ihrer Amazon VPC zu starten.

- [Erstellen einer Subnetzgruppe](#)
- [Erstellen eines Memcached-Clusters \(Konsole\).](#)
- [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\).](#)

Herstellen einer Verbindung mit einem Cache, der in einer Amazon VPC ausgeführt wird

Dieses Beispiel zeigt, wie Sie eine EC2 Amazon-Instance in Ihrer Amazon VPC starten. Sie können sich dann bei dieser Instance anmelden und auf den ElastiCache Cache zugreifen, der in der Amazon VPC ausgeführt wird.

Herstellen einer Verbindung mit einem Cache, der in einer Amazon VPC ausgeführt wird (Konsole)

In diesem Beispiel erstellen Sie eine EC2 Amazon-Instance in Ihrer Amazon VPC. Sie können diese EC2 Amazon-Instance verwenden, um eine Verbindung zu Cache-Knoten herzustellen, die in der Amazon VPC ausgeführt werden.

Note

Informationen zur Nutzung von Amazon EC2 finden Sie im [Amazon-Leitfaden „EC2 Erste Schritte“](#) in der [EC2 Amazon-Dokumentation](#).

So erstellen Sie mithilfe der Amazon-Konsole eine EC2 Amazon-Instance in Ihrer Amazon EC2 VPC

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie in der Konsole Launch Instance aus und folgen Sie diesen Schritten:
3. Wählen Sie auf der Seite Choose an Amazon Machine Image (AMI) (Amazon Machine Image (AMI) auswählen) das 64-Bit-Amazon Linux-AMI aus und klicken Sie auf Select (Auswählen).
4. Wählen Sie auf der Seite Instance-Typ auswählen die Option 3 aus. Instance konfigurieren.
5. Treffen Sie auf der Seite Configure Instance Details (Instance-Details konfigurieren) folgende Auswahl:
 - a. Wählen Sie der Liste Netzwerk Ihre Amazon VPC aus.
 - b. Wählen Sie in der Liste Subnet (Subnetz) Ihr öffentliches Subnetz aus.

Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie 4. Speicher hinzufügen.

6. Wählen Sie auf der Seite Add Storage (Speicher hinzufügen) die Option 5. Taggen der Instance.
7. Geben Sie auf der Seite Tag Instance einen Namen für Ihre EC2 Amazon-Instance ein und wählen Sie dann 6. Konfigurieren der Sicherheitsgruppe.
8. Wählen Sie auf der Seite Configure Security Group (Sicherheitsgruppe konfigurieren) die Option Select an existing security group (Bestehende Sicherheitsgruppe auswählen) aus. Weitere Informationen zu Sicherheitsgruppen finden Sie unter [EC2 Amazon-Sicherheitsgruppen für Linux-Instances](#).

Security Group ID	Name	Description
☐ sg-1a3d2178	default	default VPC security group
☒ sg-f13d2193	my-vpc-security-group	Testing

Wählen Sie den Namen Ihrer Amazon-VPC-Sicherheitsgruppe aus und wählen Sie anschließend Prüfen und starten.

9. Wählen Sie auf der Seite Review Instance and Launch (Instance prüfen und starten) die Option Launch (Starten) aus.

10. Geben Sie im Fenster Select an existing key pair or create a new key pair (Ein vorhandenes Schlüsselpaar auswählen oder ein neues Schlüsselpaar erstellen) ein Schlüsselpaar an, das Sie mit dieser Instance verwenden möchten.

 Note

Informationen zur Verwaltung von Schlüsselpaaren finden Sie im [Amazon EC2 Getting Started Guide](#).

11. Wenn Sie bereit sind, Ihre EC2 Amazon-Instance zu starten, wählen Sie Launch.

Sie können der EC2 Amazon-Instance, die Sie gerade erstellt haben, jetzt eine Elastic IP-Adresse zuweisen. Sie müssen diese IP-Adresse verwenden, um eine Verbindung zur EC2 Amazon-Instance herzustellen.

So weisen Sie eine Elastic IP-Adresse zu (Konsole)

1. Öffnen Sie die Amazon-VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie in der Navigationsliste Elastic aus IPs.
3. Wählen Sie Elastic-IP-Adresse zuweisen aus.
4. Übernehmen Sie im Dialogfeld Allocate Elastic IP address (Elastic IP-Adresse zuweisen) die standardmäßige Network Border Group (Netzwerkrenzgruppe) und klicken Sie auf Allocate (Zuweisen).
5. Wählen Sie die Elastic IP-Adresse aus, die Sie gerade aus der Liste zugewiesen haben, und klicken Sie dann auf Associate Address (Adresse verknüpfen).
6. Wählen Sie im Dialogfeld Associate Address im Feld Instance die ID der EC2 Amazon-Instance aus, die Sie gestartet haben.

Markieren Sie im Feld Private IP address (Private IP-Adresse) das Kästchen, um die private IP-Adresse abzurufen, und wählen Sie dann Associate (Verknüpfen).

Sie können jetzt SSH verwenden, um mit der Elastic IP-Adresse, die Sie erstellt haben, eine Verbindung zur EC2 Amazon-Instance herzustellen.

So stellen Sie eine Verbindung zu Ihrer EC2 Amazon-Instance her

- Öffnen Sie ein Befehlsfenster. Geben Sie an der Eingabeaufforderung den folgenden Befehl aus, wobei Sie `mykeypair.pem` durch den Namen Ihrer Schlüsselpaardatei und `54.207.55.251` durch Ihre Elastic IP-Adresse ersetzen.

```
ssh -i mykeypair.pem ec2-user@54.207.55.251
```

 Important

Melden Sie sich noch nicht von Ihrer EC2 Amazon-Instance ab.

Sie sind jetzt bereit, mit Ihrem ElastiCache Cluster zu interagieren. Vorher aber müssen Sie, wenn Sie es noch nicht getan haben, das Dienstprogramm telnet installieren.

Um Telnet zu installieren und mit Ihrem Cluster zu interagieren ()AWS CLI

- Öffnen Sie ein Befehlsfenster. Führen Sie an der Eingabeaufforderung den folgenden Befehl aus. Geben Sie bei der Eingabeaufforderung zur Bestätigung `y` ein.

```
sudo yum install telnet
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                | 63 kB      00:00

...(output omitted)...

Complete!
```

Sie können jetzt entweder mit Memcached oder Redis eine Verbindung zu einer VPC herstellen.

Mit Memcached eine Verbindung zu einer VPC herstellen

1. Rufen Sie die ElastiCache Konsole unter auf <https://console.aws.amazon.com/elasticache/> und rufen Sie den Endpunkt für einen der Knoten in Ihrem Cluster ab. Weitere Informationen finden Sie unter [Finden von Verbindungsendpunkten](#).
2. Verwenden Sie telnet, um über Port 11211 eine Verbindung zu Ihrem Cache-Knotenendpunkt herzustellen. Ersetzen Sie den unten gezeigten Hostnamen durch den Hostnamen Ihres Cache-Knotens.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 11211
```

Sie sind nun mit der Cache-Engine verbunden und können Befehle ausgeben. In diesem Beispiel fügen Sie dem Cache ein Datenelement hinzu und rufen es dann sofort danach ab. Schließlich trennen Sie die Verbindung zum Cache-Knoten.

Um einen Schlüssel und einen Wert zu speichern, geben Sie die folgenden zwei Zeilen ein:

```
add mykey 0 3600 28  
This is the value for mykey
```

Die Cache-Engine antwortet wie folgt:

```
OK
```

So rufen Sie den Wert für mykey. Geben Sie für den Wert Folgendes ein:

```
get mykey
```

Die Cache-Engine antwortet wie folgt:

```
VALUE mykey 0 28  
This is the value for my key  
END
```

Geben Sie Folgendes ein, um die Verbindung zum Cache-Modul zu trennen:

```
quit
```

Mit Redis eine Verbindung zu einer VPC herstellen

1. Rufen Sie die ElastiCache Konsole unter auf <https://console.aws.amazon.com/elasticache/> und rufen Sie den Endpunkt für einen der Knoten in Ihrem Cluster ab. Weitere Informationen [finden Sie unter Verbindungsendpunkte](#) für Redis suchen.
2. Verwenden Sie telnet, um über Port 6379 eine Verbindung zu Ihrem Cache-Knotenendpunkt herzustellen. Ersetzen Sie den unten gezeigten Hostnamen durch den Hostnamen Ihres Cache-Knotens.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Sie sind nun mit der Cache-Engine verbunden und können Befehle ausgeben. In diesem Beispiel fügen Sie dem Cache ein Datenelement hinzu und rufen es dann sofort danach ab. Schließlich trennen Sie die Verbindung zum Cache-Knoten.

Um einen Schlüssel und einen Wert zu speichern, geben Sie die folgenden zwei Zeilen ein:

```
set mykey myvalue
```

Die Cache-Engine antwortet wie folgt:

```
OK
```

So rufen Sie den Wert für mykey. Geben Sie für den Wert Folgendes ein:

```
get mykey
```

Geben Sie Folgendes ein, um die Verbindung zum Cache-Modul zu trennen:

```
quit
```

3. Rufen Sie die ElastiCache Konsole unter auf <https://console.aws.amazon.com/elasticache/> und rufen Sie den Endpunkt für einen der Knoten in Ihrem Cluster ab. Weitere Informationen [finden Sie unter Verbindungsendpunkte](#) für Redis OSS suchen.

4. Verwenden Sie telnet, um über Port 6379 eine Verbindung zu Ihrem Cache-Knotenendpunkt herzustellen. Ersetzen Sie den unten gezeigten Hostnamen durch den Hostnamen Ihres Cache-Knotens.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Sie sind nun mit der Cache-Engine verbunden und können Befehle ausgeben. In diesem Beispiel fügen Sie dem Cache ein Datenelement hinzu und rufen es dann sofort danach ab. Schließlich trennen Sie die Verbindung zum Cache-Knoten.

Geben Sie zum Speichern eines Schlüssels und eines Werts Folgendes ein:

```
set mykey myvalue
```

Die Cache-Engine antwortet wie folgt:

```
OK
```

So rufen Sie den Wert für mykey. Geben Sie für den Wert Folgendes ein:

```
get mykey
```

Die Cache-Engine antwortet wie folgt:

```
get mykey  
myvalue
```

Geben Sie Folgendes ein, um die Verbindung zum Cache-Modul zu trennen:

```
quit
```

Important

Um zusätzliche Gebühren für Ihr AWS Konto zu vermeiden, sollten Sie alle AWS Ressourcen löschen, die Sie nicht mehr benötigen, nachdem Sie diese Beispiele ausprobiert haben.

ElastiCache API- und Schnittstellen-VPC-Endpunkte (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihrer VPC und den Amazon ElastiCache API-Endpunkten herstellen, indem Sie einen VPC-Schnittstellen-Endpunkt erstellen. Schnittstellen-Endpunkte werden betrieben von [AWS PrivateLink](#). AWS PrivateLink ermöglicht Ihnen den privaten Zugriff auf Amazon ElastiCache API-Operationen ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder AWS Direct Connect-Verbindung.

Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit ElastiCache Amazon-API-Endpunkten zu kommunizieren. Ihre Instances benötigen auch keine öffentlichen IP-Adressen, um die verfügbaren ElastiCache API-Operationen nutzen zu können. Der Verkehr zwischen Ihrer VPC und Amazon verlässt das Amazon-Netzwerk ElastiCache nicht. Jeder Schnittstellenendpunkt wird durch eine oder mehrere Elastic Network-Schnittstellen in Ihren Subnetzen dargestellt. Weitere Informationen zu Elastic Network Interfaces finden Sie unter [Elastic Network Interfaces](#) im EC2 Amazon-Benutzerhandbuch.

- Weitere Informationen zu VPC-Endpunkten finden Sie unter [Interface VPC Endpoints \(AWS PrivateLink\)](#) im Amazon VPC-Benutzerhandbuch.
- [Weitere Informationen zu API-Vorgängen finden Sie unter ElastiCache API-Operationen.](#)

Nachdem Sie einen VPC-Schnittstellen-Endpunkt erstellt haben und [private DNS-Hostnamen](#) für den Endpunkt aktivieren, ist dies der ElastiCache Standardendpunkt (<https://elasticache.Region.amazonaws.com>) und DualStack-Endpunkt (<https://elasticache.Region.api.aws>) auf Ihren VPC-Endpunkt auflösen. Wenn Sie keine privaten DNS-Hostnamen aktiviert haben, stellt Amazon VPC einen DNS-Endpunktnamen bereit, den Sie im folgenden Format verwenden können:

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

Weitere Informationen finden Sie unter [Interface VPC Endpoints \(AWS PrivateLink\)](#) im Amazon VPC-Benutzerhandbuch. ElastiCache unterstützt Aufrufe aller [API-Aktionen](#) in Ihrer VPC.

Note

Private DNS-Hostnamen können nur für einen VPC-Endpunkt in der VPC aktiviert werden. Wenn Sie einen zusätzlichen VPC-Endpunkt erstellen möchten, sollte der private DNS-Hostname für diesen Endpunkt deaktiviert werden.

Verfügbare Privatelink-Regionen

Code	Speicherort	Region
CPT	Afrika (Kapstadt)	AF-SOUTH-1
HKG	Asien-Pazifik (Hongkong)	AP-OST-1
TYP	Asien-Pazifik (Taipeh)	AP-OST-2
NRT	Asien-Pazifik (Tokio)	AP-NORDOST-1
ICON	Asien-Pazifik (Seoul)	AP-NORDOST-2
MISCHEN	Asia Pacific (Osaka)	AP-NORDOST-3
BOM	Asien-Pazifik (Mumbai)	AP-SÜD-1
HYD	Asien-Pazifik (Hyderabad)	AP-SÜD-2
SIN	Asien-Pazifik (Singapur)	AP-SÜDOST-1
SYD	Asien-Pazifik (Sydney)	AP-SÜDOST-2
CGK	Asien-Pazifik (Jakarta)	AP-SÜDOST-3
MEL	Asien-Pazifik (Melbourne)	AP-SÜDOST-4
KUL	Asien-Pazifik (Malaysia)	AP-SÜDOST-5
BKK	Asien-Pazifik (Thailand)	AP-SÜDOST-7
YUL	Kanada (Zentral)	CA-ZENTRAL-1
YYC	Kanada West (Calgary)	CA-WEST-1
BUS	China (Beijing)	CN-NORTH-1
SCHÜCHTERN	China (Ningxia)	CN-NORDWEST-1
FRA	Europa (Frankfurt)	EU-CENTRAL-1
ZRH	Europa (Zürich)	EU-CENTRAL-2

Code	Speicherort	Region
ARN	Europa (Stockholm)	EU-NORD-1
MXP	Europa (Milan)	EU-SÜD-1
ZAZ	Europa (Spain)	EU-SÜD-2
DUB	Europa (Irland)	EU-WEST-1
LHR	Europa (London)	EU-WEST-2
CDG	Europa (Paris)	EU-WEST-3
TLV	Tel Aviv (Israel)	IL CENTRAL-1
DXB	Naher Osten (VAE)	ME-CENTRAL-1
EIN BAD	Naher Osten (Bahrain)	ICH-SÜD-1
QRO	Mexiko (Zentral)	MX-CENTRAL-1
GRU	Südamerika (São Paulo)	SA-OST-1
IAD	USA Ost (Nord-Virginia)	US-OST-1
CMH	USA Ost (Ohio)	US-OST-2
OSU	AWS GovCloud (US-Ost)	US-GOV-OST-1
SFO	USA West (Nordkalifornien)	US-WEST-1
PDX	USA West (Oregon)	US-WEST-2
PDT	AWS GovCloud (US-West)	US-WEST-1

Überlegungen zu VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für Amazon ElastiCache API-Endpunkte einrichten, sollten Sie die [Eigenschaften und Einschränkungen der Schnittstellen-Endpunkte](#) im Amazon VPC-

Benutzerhandbuch lesen. Alle ElastiCache API-Operationen, die für die Verwaltung von ElastiCache Amazon-Ressourcen relevant sind, sind in Ihrer VPC über AWS PrivateLink verfügbar.

VPC-Endpunktrichtlinien werden für ElastiCache API-Endpunkte unterstützt. Standardmäßig ist der vollständige Zugriff auf ElastiCache API-Operationen über den Endpunkt zulässig. Weitere Informationen finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon-VPC-Benutzerhandbuch.

Erstellen eines VPC-Schnittstellen-Endpunkts für die API ElastiCache

Sie können einen VPC-Endpunkt für die ElastiCache Amazon-API entweder mit der Amazon VPC-Konsole oder mit dem erstellen.AWS CLI Weitere Informationen finden Sie unter [Erstellung eines Schnittstellenendpunkts](#) im Benutzerhandbuch für Amazon VPC.

Nachdem Sie einen Schnittstellen-VPC-Endpunkt erstellt haben, können Sie private DNS-Hostnamen für den Endpunkt aktivieren. Wenn Sie dies tun, der ElastiCache Standard-Amazon-Endpunkt (<https://elasticache.Region.amazonaws.com>) und Dual-Stack-Endpunkt (<https://elasticache.Region.api.aws>) auf Ihren VPC-Endpunkt auflösen. Für die AWS Regionen China (Peking) und China (Ningxia) können Sie API-Anfragen mit dem VPC-Endpunkt stellen, indem Sie `elasticache.cn-north-1.amazonaws.com.cn` und `elasticache.cn-north-1.api.amazonwebservices.com.cn` für Peking und und für Ningxia verwenden. `elasticache.cn-northwest-1.amazonaws.com.cn` `elasticache.cn-northwest-1.api.amazonwebservices.com.cn` Weitere Informationen finden Sie unter [Zugriff auf einen Service über einen Schnittstellenendpunkt](#) im Benutzerhandbuch für Amazon VPC.

Erstellen einer VPC-Endpunktrichtlinie für die Amazon-API ElastiCache

Sie können Ihrem VPC-Endpunkt eine Endpunktrichtlinie hinzufügen, die den Zugriff auf die ElastiCache API steuert. Die Richtlinie legt Folgendes fest:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon-VPC-Benutzerhandbuch.

Example VPC-Endpunktrichtlinie für ElastiCache API-Aktionen mit Valkey oder Redis OSS

Das Folgende ist ein Beispiel für eine Endpunktrichtlinie für die API. ElastiCache Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die aufgelisteten ElastiCache API-Aktionen.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster",
      "elasticache:CreateSnapshot"
    ],
    "Resource": "*"
  }]
}
```

Example VPC-Endpunktrichtlinie ElastiCache für Memcached-API-Aktionen

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für die API. ElastiCache Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die aufgelisteten ElastiCache API-Aktionen.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster"
    ],
    "Resource": "*"
  }]
}
```

Example VPC-Endpunktrichtlinie, die jeglichen Zugriff von einem bestimmten Konto aus verweigert AWS

Die folgende VPC-Endpunktrichtlinie verweigert dem AWS Konto **123456789012** jeglichen Zugriff auf Ressourcen, die den Endpunkt verwenden. Die Richtlinie erlaubt alle Aktionen von anderen Konten.

```
{
  "Statement": [{
    "Action": "*",
    "Effect": "Allow",
    "Resource": "*",
    "Principal": "*"
  },
  {
    "Action": "*",
    "Effect": "Deny",
    "Resource": "*",
    "Principal": {
      "AWS": [
        "123456789012"
      ]
    }
  }
]
```

Subnetze und Subnetzgruppen

Eine Subnetzgruppe ist eine Sammlung von Subnetzen (normalerweise privat), die Sie für Ihre knotenbasierten Cluster festlegen können, die in einer Amazon Virtual Private Cloud (VPC) - Umgebung ausgeführt werden.

Wenn Sie einen knotenbasierten Cluster in einer Amazon VPC erstellen, müssen Sie eine Subnetzgruppe verwenden. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten zugeordnet werden sollen.

ElastiCache stellt eine IPv4 Standard-Subnetzgruppe bereit, oder Sie können wählen, ob Sie eine neue erstellen möchten. Denn IPv6 Sie müssen eine Subnetzgruppe mit einem IPv6 CIDR-Block erstellen. Wenn Sie sich für Dual-Stack entscheiden, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder oder IPv6 . IPv4

ElastiCache Serverless verwendet keine Subnetzgruppenressource, sondern verwendet stattdessen direkt bei der Erstellung eine Liste von Subnetzen.

In diesem Abschnitt wird beschrieben, wie Sie Subnetze und Subnetzgruppen erstellen und nutzen, um den Zugriff auf Ihre Ressourcen zu verwalten. ElastiCache

Weitere Informationen zur Verwendung von Subnetzgruppen in einer Amazon-VPC-Umgebung finden Sie unter [Zugreifen auf Ihren ElastiCache Cluster oder Ihre Replikationsgruppe](#).

Themen

- [Erstellen einer Subnetzgruppe](#)
- [Zuweisen einer Subnetzgruppe zu einem Cache](#)
- [Ändern einer Subnetzgruppe](#)
- [Löschen einer Subnetzgruppe](#)

Erstellen einer Subnetzgruppe

Eine Cache-Subnetzgruppe ist eine Zusammenstellung von Subnetzen, die Sie für Ihre Cache-Cluster in einer VPC festlegen können. Wenn Sie einen Cache in einer VPC starten, müssen Sie eine Cache-Subnetzgruppe auswählen. ElastiCache verwendet dann diese Cache-Subnetzgruppe, um jedem Cache-Knoten im Cache IP-Adressen innerhalb dieses Subnetzes zuzuweisen.

Wenn Sie eine neue Subnetzgruppe erstellen, notieren Sie sich die Anzahl der verfügbaren IP-Adressen. Wenn das Subnetz nur über wenige freie IP-Adressen verfügt, beschränkt dies auch die Anzahl der neuen Knoten, die Sie einem Cluster hinzufügen können. Um dieses Problem zu lösen, können Sie einer Subnetzgruppe weitere Subnetze zuweisen, um ausreichend IP-Adressen in der Availability Zone Ihres Clusters bereitzustellen. Danach können Sie dem Cluster weitere Knoten hinzufügen.

Wenn Sie IPV4 als Netzwerktyp wählen, ist eine Standard-Subnetzgruppe verfügbar, oder Sie können wählen, ob Sie eine neue erstellen möchten. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten zugeordnet werden sollen. Wenn Sie Dual-Stack oder wählen, werden Sie angewiesen IPV6, Dual-Stack oder Subnetze zu erstellen. IPV6 Weitere Informationen zu Netzwerktypen finden Sie unter [Auswahl eines Netzwerktyps in ElastiCache](#). Weitere Informationen finden Sie unter [Erstellen eines Subnetzes in Ihrer VPC](#).

Die folgenden Verfahren zeigen Ihnen, wie Sie eine Subnetzgruppe mit den Namen mysubnetgroup (Konsole) AWS CLI, die und die API erstellen. ElastiCache

Erstellen einer Subnetzgruppe (Konsole)

Im folgenden Verfahren wird das Erstellen einer Subnetzgruppe (Konsole) erläutert.

Erstellen einer DB-Sicherheitsgruppe (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie in der Navigationsliste auf Subnetzgruppen.
3. Wählen Sie Create subnet group (Subnetz-Grupp erstellen) aus.
4. Gehen Sie im Assistenten Subnetzgruppe erstellen wie folgt vor. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Erstellen aus.
 - a. Geben Sie im Feld Name einen Namen für Ihre Subnetzgruppe ein.

- b. Geben Sie im Feld Description eine Beschreibung für Ihre Subnetzgruppe ein.
 - c. Wählen Sie im Feld VPC-ID Ihre Amazon VPC aus.
 - d. Standardmäßig werden alle Subnetze ausgewählt. Klicken Sie im Bereich Ausgewählte Subnetze auf Verwalten und wählen Sie die Availability Zones oder [Local Zones](#) sowie Ihre privaten Subnetze IDs aus, und wählen Sie dann Auswählen aus.
5. Klicken Sie in der angezeigten Bestätigungsmeldung auf Close.

Ihre neue Subnetzgruppe wird in der Liste der Subnetzgruppen der Konsole angezeigt. ElastiCache Unten im Fenster können Sie die Subnetzgruppe auswählen, um Details wie die der Gruppe zugeordneten Subnetze anzuzeigen.

Erstellen einer Subnetzgruppe (AWS CLI)

Geben Sie in einem Befehlszeilenfenster den Befehl `create-cache-subnet-group` ein, um eine Subnetzgruppe zu erstellen.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-subnet-group \
  --cache-subnet-group-name mysubnetgroup \
  --cache-subnet-group-description "Testing" \
  --subnet-ids subnet-53df9c3a
```

Für Windows:

```
aws elasticache create-cache-subnet-group ^
  --cache-subnet-group-name mysubnetgroup ^
  --cache-subnet-group-description "Testing" ^
  --subnet-ids subnet-53df9c3a
```

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

```
{
  "CacheSubnetGroup": {
    "VpcId": "vpc-37c3cd17",
    "CacheSubnetGroupDescription": "Testing",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-53df9c3a",
```

```
        "SubnetAvailabilityZone": {
            "Name": "us-west-2a"
        }
    ],
    "CacheSubnetGroupName": "mysubnetgroup"
}
```

Weitere Informationen finden Sie im AWS CLI Thema. [create-cache-subnet-group](#)

Zuweisen einer Subnetzgruppe zu einem Cache

Nach dem Erstellen einer Subnetzgruppe können Sie einen Cache in einer Amazon-VPC starten. Weitere Informationen finden Sie unter den folgenden Topics.

- Memcached-Cluster – Informationen zum Starten eines Memcached-Clusters finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#). Wählen Sie in Schritt 7.a (Advanced Memcached Settings) eine VPC-Subnetzgruppe aus.
- Eigenständiger Valkey- oder Redis OSS-Cluster — Informationen zum Starten eines Valkey- oder Redis OSS-Clusters mit einem Knoten finden Sie unter. [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#) Wählen Sie in Schritt 7.a (Erweiterte Redis OSS-Einstellungen) eine VPC-Subnetzgruppe aus.
- Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus deaktiviert) — Informationen zum Starten einer Valkey- oder Redis OSS-Replikationsgruppe (Cluster-Modus deaktiviert) in einer VPC finden Sie unter. [Eine Valkey- oder Redis OSS-Replikationsgruppe \(Cluster Mode Disabled\) von Grund auf neu erstellen](#) Wählen Sie in Schritt 7.b (Erweiterte Redis OSS-Einstellungen) eine VPC-Subnetzgruppe aus.
- Valkey- oder Redis OSS-Replikationsgruppe (Clustermodus aktiviert) —. [Erstellen eines Valkey- oder Redis OSS-Clusters \(Clustermodus aktiviert\) \(Konsole\)](#) Wählen Sie in Schritt 6.i (Erweiterte Redis OSS-Einstellungen) eine VPC-Subnetzgruppe aus.

Ändern einer Subnetzgruppe

Sie können die Beschreibung einer Subnetzgruppe oder die Liste der Subnetze ändern, die der Subnetzgruppe IDs zugeordnet sind. Es ist nicht möglich, Subnetz-IDs aus einer Subnetzgruppe zu löschen, wenn das Subnetz derzeit von einem Cache verwendet wird.

Das folgende Verfahren zeigt, wie Sie eine Subnetzgruppe ändern.

Subnetzgruppen ändern (Konsole)

So ändern Sie eine Subnetzgruppe

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.
3. Wählen Sie in der Liste der Subnetzgruppen das Optionsfeld für die gewünschte Subnetzgruppe aus und klicken Sie auf Ändern.
4. Wählen Sie im Bereich Ausgewählte Subnetze die Option Verwalten aus.
5. Nehmen Sie die gewünschten Änderungen an den ausgewählten Subnetzen vor und klicken Sie auf Auswählen.
6. Klicken Sie auf Änderungen speichern, um Ihre Änderungen zu speichern.

Ändern von Subnetzgruppen (AWS CLI)

Geben Sie in einem Befehlszeilenfenster den Befehl `modify-cache-subnet-group` ein, um eine Subnetzgruppe zu ändern.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-subnet-group \
  --cache-subnet-group-name mysubnetgroup \
  --cache-subnet-group-description "New description" \
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Für Windows:

```
aws elasticache modify-cache-subnet-group ^
  --cache-subnet-group-name mysubnetgroup ^
  --cache-subnet-group-description "New description" ^
```

```
--subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

```
{
  "CacheSubnetGroup": {
    "VpcId": "vpc-73cd3c17",
    "CacheSubnetGroupDescription": "New description",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-42dcf93a",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      },
      {
        "SubnetIdentifier": "subnet-48fc12a9",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      }
    ],
    "CacheSubnetGroupName": "mysubnetgroup"
  }
}
```

Weitere Informationen finden Sie im AWS CLI Thema [modify-cache-subnet-group](#).

Löschen einer Subnetzgruppe

Wenn Sie eine Subnetzgruppe nicht mehr benötigen, können Sie sie löschen. Sie können eine Subnetzgruppe, die derzeit von einem Cache verwendet wird, nicht löschen.

Das folgende Verfahren zeigt, wie Sie eine Subnetzgruppe löschen.

Löschen einer Subnetzgruppe (Konsole)

So löschen Sie eine Subnetzgruppe

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.
3. Wählen Sie in der Liste der Subnetzgruppen die zu löschende Subnetzgruppe aus und klicken Sie auf Delete.
4. Wenn Sie aufgefordert werden, diesen Vorgang zu bestätigen, geben Sie den Namen der Subnetzgruppe in das Texteingabefeld ein und wählen Sie Löschen aus.

Löschen einer Subnetzgruppe (AWS CLI)

Rufen Sie mit dem AWS CLI den Befehl `delete-cache-subnet-group` mit dem folgenden Parameter auf:

- `--cache-subnet-group-name` *mysubnetgroup*

Für Linux, macOS oder Unix:

```
aws elasticache delete-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup
```

Für Windows:

```
aws elasticache delete-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CLI Thema [delete-cache-subnet-group](#).

Identity and Access Management für Amazon ElastiCache

AWS Identity and Access Management(IAM) hilft einem Administrator AWS-Service, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. ElastiCache IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So ElastiCache arbeitet Amazon mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)
- [Fehlerbehebung Amazon ElastiCache Amazon-Identität und -Zugriff](#)
- [Zugriffskontrolle](#)
- [Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre ElastiCache Ressourcen](#)

Zielgruppe

Wie Sie AWS Identity and Access Management(IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Funktionen zugreifen können (siehe [Fehlerbehebung Amazon ElastiCache Amazon-Identität und -Zugriff](#))
- Dienstadministrator – bestimmen Sie den Benutzerzugriff und reichen Sie Berechtigungsanfragen ein (siehe [So ElastiCache arbeitet Amazon mit IAM](#))
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#))

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter [So melden Sie sich bei Ihrem AWS-Konto an](#) im Benutzerhandbuch für AWS-Anmeldung.

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter [AWS Signature Version 4 for API requests](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Tasks that require root user credentials](#) im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle

von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu können](#).

Eine [IAM-Gruppe](#) spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie [von einer Benutzer- zu einer IAM-Rolle \(Konsole\) wechseln](#) AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter [Methoden, um eine Rolle zu übernehmen](#) im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für Verbundbenutzerzugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, dienstübergreifenden Zugriff und Anwendungen, die auf Amazon ausgeführt werden. EC2 Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an Identitäten oder Ressourcen anhängen. AWS Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcen und unter welchen Bedingungen durchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter [Choose between managed policies and inline policies](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#).

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinienarten

AWS unterstützt zusätzliche Richtlinienarten, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinienarten gewährt werden:

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Richtlinien zur Dienstkontrolle (SCPs)** — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter [Service-Kontrollrichtlinien](#) im AWS Organizations-Benutzerhandbuch.
- **Richtlinien zur Ressourcenkontrolle (RCPs)** — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter [Richtlinien zur Ressourcenkontrolle \(RCPs\)](#).

- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter [Bewertungslogik für Richtlinien](#) im IAM-Benutzerhandbuch.

So ElastiCache arbeitet Amazon mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf verwenden, sollten Sie sich darüber informieren ElastiCache, mit welchen IAM-Funktionen Sie arbeiten können. ElastiCache

IAM-Funktionen, die Sie mit Amazon verwenden können ElastiCache

IAM-Feature	ElastiCache Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Ja
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja

IAM-Feature	ElastiCache Unterstützung
Service-verknüpfte Rollen	Ja

Einen allgemeinen Überblick darüber, wie ElastiCache und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im [IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren](#).

Identitätsbasierte Richtlinien für ElastiCache

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für ElastiCache

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

Ressourcenbasierte Richtlinien in ElastiCache

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein

bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören.AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Richtlinienaktionen für ElastiCache

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der ElastiCache Aktionen finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#) in der Service Authorization Reference.

Bei den in der Richtlinie ElastiCache verwendeten Aktionen wird vor der Aktion das folgende Präfix verwendet:

```
elasticache
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
    "elasticache:action1",  
    "elasticache:action2"  
]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "elasticache:Describe"
```

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

Politische Ressourcen für ElastiCache

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der ElastiCache Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Von Amazon definierte Ressourcen ElastiCache](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

Richtlinien-Bedingungsschlüssel für ElastiCache

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#)

verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der ElastiCache Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon ElastiCache](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

Zugriffskontrolllisten () ACLs in ElastiCache

Unterstützt ACLs: Ja

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle (ABAC) mit ElastiCache

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-Entitäten und AWS-Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Definieren von Berechtigungen mit ABAC-Autorisierung](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Temporäre Anmeldeinformationen verwenden mit ElastiCache

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie einen Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporary security credentials in IAM](#) und [AWS-Services that work with IAM](#) im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipal-Berechtigungen für ElastiCache

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen AWS-Service an nachgeschaltete Dienste zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für ElastiCache

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Warning

Durch das Ändern der Berechtigungen für eine Servicerolle kann die ElastiCache Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, wenn ElastiCache eine Anleitung dazu gibt.

Mit Diensten verknüpfte Rollen für ElastiCache

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene

Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS-Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, ElastiCache-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien \(Konsole\)](#) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden ElastiCache, einschließlich des Formats von ARNs für jeden der Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon ElastiCache](#) in der Service Authorization Reference.

Themen

- [Best Practices für Richtlinien](#)
- [Verwenden der Konsole ElastiCache](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand ElastiCache Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar.AWS-Konto Wir empfehlen Ihnen,

die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) oder [Von AWS verwaltete Richtlinien für Auftragsfunktionen](#) im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z.CloudFormation B. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung mit IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Sicherer API-Zugriff mit MFA](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Best Practices für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der Konsole ElastiCache

Um auf die ElastiCache Amazon-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Informationen zu den ElastiCache Ressourcen in Ihrem Verzeichnis aufzulisten und einzusehen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die ElastiCache Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die ElastiCache ConsoleAccess oder die ReadOnlyAWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI/AWS OR-API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
```

```
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
  }
]
```

Fehlerbehebung Amazon ElastiCache Amazon-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit ElastiCache IAM auftreten können.

Themen

- [Ich bin nicht autorisiert, eine Aktion in ElastiCache auszuführen.](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine ElastiCache Ressourcen ermöglichen](#)

Ich bin nicht autorisiert, eine Aktion in ElastiCache auszuführen.

Wenn Ihnen AWS-Managementkonsole mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion durchzuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson`-Benutzer versucht, die Konsole zum Anzeigen von Details zu einer fiktiven *my-example-widget*-Ressource zu verwenden, jedoch nicht über `elasticache:GetWidget`-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
elasticache:GetWidget on resource: my-example-widget
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion *my-example-widget* auf die Ressource *elasticache:GetWidget* zugreifen zu können.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an ElastiCache übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in ElastiCache auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine ElastiCache Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen ElastiCache unterstützt werden, finden Sie unter [So ElastiCache arbeitet Amazon mit IAM](#)
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto, den Sie besitzen](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Zugriffskontrolle

Sie können über gültige Anmeldeinformationen verfügen, um Ihre Anfragen zu authentifizieren, aber ohne die entsprechenden Berechtigungen können Sie keine Ressourcen erstellen oder darauf zugreifen ElastiCache. Sie benötigen beispielsweise die erforderlichen Berechtigungen, um einen ElastiCache Cluster zu erstellen.

In den folgenden Abschnitten wird beschrieben, wie Sie Berechtigungen für verwalten ElastiCache. Wir empfehlen Ihnen, zunächst die Übersicht zu lesen.

- [Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre ElastiCache Ressourcen](#)
- [Verwendung identitätsbasierter Richtlinien \(IAM-Richtlinien\) für Amazon ElastiCache](#)

Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre ElastiCache Ressourcen

Jede AWS Ressource gehört einem AWS Konto, und die Berechtigungen zum Erstellen oder Zugreifen auf eine Ressource werden durch Berechtigungsrichtlinien geregelt. Ein Kontoadministrator kann Berechtigungsrichtlinien an IAM-Identitäten (Benutzer, Gruppen und Rollen) anfügen. Darüber hinaus unterstützt Amazon ElastiCache auch das Anhängen von Berechtigungsrichtlinien an Ressourcen.

Note

Ein Kontoadministrator (oder Administratorbenutzer) ist ein Benutzer mit Administratorrechten. Weitere Informationen finden Sie unter [Bewährte Methoden für IAM](#) im IAM-Benutzerhandbuch.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter [Erstellen eines Berechtigungssatzes](#) im AWS IAM Identity Center-Benutzerhandbuch.

- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter [Eine Rolle für einen externen Identitätsanbieter \(Verbund\) erstellen](#) im IAM-Benutzerhandbuch.

- IAM-Benutzer:

- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter [Eine Rolle für einen IAM-Benutzer erstellen](#) im IAM-Benutzerhandbuch.
- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter [Hinzufügen von Berechtigungen zu einem Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Themen

- [ElastiCache Ressourcen und Betriebsabläufe von Amazon](#)
- [Grundlegendes zum Eigentum an Ressourcen](#)

- [Verwaltung des Zugriffs auf -Ressourcen](#)
- [AWS verwaltete Richtlinien für Amazon ElastiCache](#)
- [Verwendung identitätsbasierter Richtlinien \(IAM-Richtlinien\) für Amazon ElastiCache](#)
- [Berechtigungen auf Ressourcenebene](#)
- [Verwenden von Bedingungsschlüssel](#)
- [Verwenden von serviceverknüpften Rollen für Amazon ElastiCache](#)
- [ElastiCache API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen](#)

ElastiCache Ressourcen und Betriebsabläufe von Amazon

Eine Liste der ElastiCache Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Von Amazon definierte Ressourcen ElastiCache](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

Grundlegendes zum Eigentum an Ressourcen

Ein Ressourcenbesitzer ist das AWS Konto, das die Ressource erstellt hat. Das heißt, der Ressourcenbesitzer ist das AWS Konto der Prinzipalentität, die die Anforderung authentifiziert, mit der die Ressource erstellt wird. Eine Haupt-Entität kann das Stammkonto, ein IAM-Benutzer oder eine IAM-Rolle sein. Die Funktionsweise wird anhand der folgenden Beispiele deutlich:

- Angenommen, Sie verwenden die Root-Kontoanmeldeinformationen Ihres AWS Kontos, um einen Cache-Cluster zu erstellen. In diesem Fall ist Ihr AWS Konto der Eigentümer der Ressource. ElastiCache ist die Ressource der Cluster.
- Angenommen, Sie erstellen in Ihrem AWS Konto einen IAM-Benutzer und gewähren diesem Benutzer die Berechtigung, einen Cluster zu erstellen. In diesem Fall kann der Benutzer einen Cluster erstellen. Ihr AWS Konto, zu dem der Benutzer gehört, besitzt jedoch die Clusterressource.
- Angenommen, Sie erstellen in Ihrem AWS Konto eine IAM-Rolle mit Berechtigungen zum Erstellen eines Clusters. In diesem Fall kann jeder, der die Rolle übernehmen kann, einen Cluster erstellen. Ihr AWS Konto, zu dem die Rolle gehört, besitzt die Cache-Cluster-Ressource.

Verwaltung des Zugriffs auf -Ressourcen

Eine Berechtigungsrichtlinie beschreibt, wer Zugriff auf welche Objekte hat. Im folgenden Abschnitt werden die verfügbaren Optionen zum Erstellen von Berechtigungsrichtlinien erläutert.

 Note

In diesem Abschnitt wird die Verwendung von IAM im Kontext von Amazon ElastiCache beschrieben. Er enthält keine detaillierten Informationen über den IAM-Service. Eine umfassende IAM-Dokumentation finden Sie unter [Was ist IAM?](#) im IAM-Benutzerhandbuch. Für Informationen über die Syntax und Beschreibungen von [AWS-IAM-Richtlinien](#) lesen Sie die IAM-Richtlinienreferenz im IAM-Benutzerhandbuch.

An eine IAM-Identität angefügte Richtlinien werden als identitätsbasierte Richtlinien (oder IAM-Richtlinien) bezeichnet. An Ressourcen angehängte Richtlinien werden als ressourcenbasierte Richtlinien bezeichnet.

Themen

- [Identitätsbasierte Richtlinien \(IAM-Richtlinien\)](#)
- [Angaben der Richtlinienelemente: Aktionen, Effekte, Ressourcen und Prinzipale](#)
- [Angaben von Bedingungen in einer Richtlinie](#)

Identitätsbasierte Richtlinien (IAM-Richtlinien)

Richtlinien können IAM-Identitäten angefügt werden. Sie können z. B. Folgendes tun:

- Anfügen einer Berechtigungsrichtlinie zu einem Benutzer oder einer Gruppe in Ihrem Konto – Ein Kontoadministrator kann eine Berechtigungsrichtlinie verwenden, die einem bestimmten Benutzer zugeordnet ist, um Berechtigungen zu erteilen. In diesem Fall verfügt der Benutzer über die erforderlichen Berechtigungen zum Erstellen einer ElastiCache Ressource, z. B. eines Clusters, einer Parametergruppe oder einer Sicherheitsgruppe.
- Einer Rolle eine Berechtigungsrichtlinie zuweisen (kontoübergreifende Berechtigungen gewähren) – Sie können einer IAM-Rolle eine identitätsbasierte Berechtigungsrichtlinie zuweisen, um kontoübergreifende Berechtigungen zu erteilen. Der Administrator in Konto A kann beispielsweise wie folgt eine Rolle erstellen, um einem anderen Konto (z. B. AWS Konto B) oder einem AWS Dienst kontoübergreifende Berechtigungen zu gewähren:
 1. Der Administrator von Konto A erstellt eine IAM-Rolle und fügt ihr eine Berechtigungsrichtlinie an, die Berechtigungen für Ressourcen in Konto A erteilt.
 2. Der Administrator von Konto A weist der Rolle eine Vertrauensrichtlinie zu, die Konto B als den Prinzipal identifiziert, der die Rolle übernehmen kann.

3. Der Administrator von Konto B kann dann die Berechtigungen zur Übernahme der Rolle an alle Benutzer in Konto B delegieren. Auf diese Weise können Benutzer in Konto B Ressourcen in Konto A erstellen oder darauf zugreifen. In einigen Fällen möchten Sie einem AWS Dienst möglicherweise Berechtigungen zur Übernahme der Rolle erteilen. Zum Support dieses Ansatzes kann es sich beim Prinzipal in der Vertrauensrichtlinie auch um einen AWS-Service-Prinzipal handeln.

Weitere Informationen zum Delegieren von Berechtigungen mithilfe von IAM finden Sie unter [Zugriffsverwaltung](#) im IAM-Benutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die es einem Benutzer ermöglicht, die `DescribeCacheClusters` Aktion für Ihr AWS Konto durchzuführen. ElastiCache unterstützt auch die Identifizierung bestimmter Ressourcen, die die Ressource ARNs für API-Aktionen verwenden. (Dieser Ansatz heißt auch Ressourcenebenen-Berechtigungen.)

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DescribeCacheClusters",
      "Effect": "Allow",
      "Action": [
        "elasticache:DescribeCacheClusters"
      ],
      "Resource": "arn:aws:iam::*:role/*"
    }
  ]
}
```

Weitere Informationen zur Verwendung identitätsbasierter Richtlinien mit finden Sie ElastiCache unter [Verwendung identitätsbasierter Richtlinien \(IAM-Richtlinien\) für Amazon ElastiCache](#) Weitere Informationen zu Benutzern, Gruppen, Rollen und Berechtigungen finden Sie unter [Identitäten \(Benutzer, Gruppen und Rollen\)](#) im IAM-Benutzerhandbuch.

Angeben der Richtlinienelemente: Aktionen, Effekte, Ressourcen und Prinzipale

Für jede ElastiCache Amazon-Ressource (siehe [ElastiCache Ressourcen und Betriebsabläufe von Amazon](#)) definiert der Service eine Reihe von API-Vorgängen (siehe [Aktionen](#)). ElastiCache Definiert eine Reihe von Aktionen, die Sie in einer Richtlinie angeben können, um Berechtigungen für diese API-Operationen zu erteilen. Für die ElastiCache Clusterressource sind beispielsweise die folgenden Aktionen definiert: `CreateCacheCluster`, `DeleteCacheCluster`, und `DescribeCacheCluster`. Für das Durchführen einer API-Operation können Berechtigungen für mehrere Aktionen erforderlich sein.

Grundlegende Richtlinienelemente:

- **Ressource** – In einer Richtlinie wird der Amazon-Ressourcenname (ARN) zur Identifizierung der Ressource verwendet, für die die Richtlinie gilt. Weitere Informationen finden Sie unter [ElastiCache Ressourcen und Betriebsabläufe von Amazon](#).
- **Aktion** – Mit Aktionsschlüsselwörtern geben Sie die Ressourcenoperationen an, die Sie zulassen oder verweigern möchten. Je nach Angabe `Effect` gewährt oder verweigert die `elasticache:CreateCacheCluster` Berechtigung dem Benutzer beispielsweise die Berechtigungen zur Durchführung des ElastiCache `CreateCacheCluster` Amazon-Vorgangs.
- **Auswirkung** – Die von Ihnen festgelegte Auswirkung, wenn der Benutzer die jeweilige Aktion anfordert – entweder „allow“ (Zugriffserlaubnis) oder „deny“ (Zugriffsverweigerung). Wenn Sie den Zugriff auf eine Ressource nicht ausdrücklich gestatten ("Allow"), wird er automatisch verweigert. Sie können den Zugriff auf eine Ressource auch explizit verweigern. So können Sie zum Beispiel sicherstellen, dass ein Benutzer nicht auf die Ressource zugreifen kann, auch wenn der Zugriff durch eine andere Richtlinie gestattet wird.
- **Prinzipal** – In identitätsbasierten Richtlinien (IAM-Richtlinien) ist der Benutzer, dem die Richtlinie zugewiesen ist, automatisch der Prinzipal. In ressourcenbasierten Richtlinien müssen Sie den Benutzer, das Konto, den Service oder die sonstige Entität angeben, die die Berechtigungen erhalten soll (gilt nur für ressourcenbasierte Richtlinien).

Weitere Informationen zur Syntax und zu Beschreibungen von IAM-Richtlinien finden Sie in der [AWS-IAM-Richtlinienreferenz](#) im IAM-Benutzerhandbuch.

Eine Tabelle mit allen ElastiCache Amazon-API-Aktionen finden Sie unter [ElastiCache API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen](#).

Angeben von Bedingungen in einer Richtlinie

Beim Erteilen von Berechtigungen können Sie mithilfe der IAM-Richtliniensyntax die Bedingungen angeben, unter denen die Richtlinie wirksam werden soll. Beispielsweise kann festgelegt werden, dass eine Richtlinie erst ab einem bestimmten Datum gilt. Weitere Informationen zum Angeben von Bedingungen in einer Richtliniensyntax finden Sie im Thema [Bedingung](#) im IAM Benutzerhandbuch.

Bedingungen werden mithilfe vordefinierter Bedingungsschlüssel formuliert. Informationen zur Verwendung von ElastiCache -spezifischen Bedingungsschlüsseln finden Sie unter [Verwenden von Bedingungsschlüssel](#). Es gibt Bedingungsschlüssel AWS für alle Bereiche, die Sie je nach Bedarf verwenden können. Eine vollständige Liste der AWS-weiten Schlüssel finden Sie unter [Verfügbare Schlüssel für Bedingungen](#) im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinien für Amazon ElastiCache

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird. 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 für alle AWS Kunden verfügbar sind. 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 aktualisiert werden, wirkt sich das Update auf alle Prinzipalidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: ElastiCacheServiceRolePolicy

Sie können keine Verbindungen ElastiCacheServiceRolePolicy zu Ihren IAM-Entitäten herstellen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, mit der Sie Aktionen ElastiCache in Ihrem Namen ausführen können.

Diese Richtlinie ermöglicht es ElastiCache, AWS Ressourcen in Ihrem Namen zu verwalten, sofern dies für die Verwaltung Ihres Caches erforderlich ist:

- `ec2`— Verwaltung von EC2 Netzwerkressourcen, die an Cache-Knoten angehängt werden sollen, darunter VPC-Endpunkte (für serverlose Caches), Elastic Network Interfaces (ENIs) (für knotenbasierte Cluster) und Sicherheitsgruppen.
- `cloudwatch`— Sendet Metrikdaten aus dem Service in CloudWatch
- `outposts`— Erlaubt die Erstellung von Cache-Knoten auf AWS Outposts.

Sie finden die [ElastiCacheServiceRolePolicy](#) Richtlinie auf der IAM-Konsole und [ElastiCacheServiceRolePolicy](#) im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AmazonElastiCacheFullAccess

Sie können die AmazonElastiCacheFullAccess-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Principals vollen Zugriff auf ElastiCache die AWS Management Console:

- `elasticache`— Zugriff auf alle APIs.
- `iam` – Erstellen einer serviceverknüpften Rolle, die für den Servicebetrieb erforderlich ist.
- `ec2`— Beschreiben Sie abhängige EC2 Ressourcen, die für die Cache-Erstellung erforderlich sind (VPC, Subnetz, Sicherheitsgruppe), und ermöglichen Sie die Erstellung von VPC-Endpunkten (für serverlose Caches).
- `kms`— Erlaubt die Nutzung von kundenverwaltetem für. CMKs encryption-at-rest
- `cloudwatch`— Erlaubt den Zugriff auf Metriken, um ElastiCache Metriken in der Konsole anzuzeigen.
- `application-autoscaling` – Erlauben des Zugriffs zur Beschreibung von AutoScaling-Richtlinien für Caches.
- `logs` – Wird verwendet, um Protokollstreams für die Protokollbereitstellungsfunktion in der Konsole aufzufüllen.
- `firehose` – Wird verwendet, um Bereitstellungsdatenströme für die Protokollbereitstellungsfunktion in der Konsole aufzufüllen.
- `s3` – Wird verwendet, um S3-Buckets für die Funktion zur Wiederherstellung von Snapshots in der Konsole aufzufüllen.
- `outposts`— Wird verwendet, um AWS Outposts für die Cache-Erstellung in der Konsole zu füllen.
- `sns` – Wird verwendet, um SNS-Themen für die Benachrichtigungsfunktion in der Konsole auszufüllen.

Sie finden die [AmazonElastiCacheFullAccess](#) Richtlinie auf der IAM-Konsole und [AmazonElastiCacheFullAccess](#) im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AmazonElastiCacheReadOnlyAccess

Sie können die AmazonElastiCacheReadOnlyAccess-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Prinzipalen nur Lesezugriff auf die ElastiCache AWS Managementkonsole:

- `elasticache`— Nur-Lese-Zugriff. Describe APIs

Sie finden die [AmazonElastiCacheReadOnlyAccess](#) Richtlinie auf der IAM-Konsole und [AmazonElastiCacheReadOnlyAccess](#) im AWS Managed Policy Reference Guide.

ElastiCache Aktualisierungen der AWS verwalteten Richtlinien

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien, die ElastiCache seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden. Abonnieren Sie den RSS-Feed auf der Seite ElastiCache Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderungen	Beschreibung	Datum
AmazonElastiCacheFullAccess – Aktualisierung auf eine bestehende Richtlinie	ElastiCache neue Berechtigungen wurden hinzugefügt, um eine vertikale Skalierung für MemCached, für die Aktion <code>elasticache:ModifyCacheCluster</code> .	27. März 2025
AmazonElastiCacheFullAccess – Aktualisierung auf eine bestehende Richtlinie	ElastiCache neue Berechtigungen wurden hinzugefügt, um die Verwaltung von serverlosen Caches und die Nutzung aller Servicefunktionen über die Konsole zu ermöglichen.	8. November 2023
ElastiCacheServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	ElastiCache neue Berechtigungen hinzugefügt, um die Verwaltung von VPC-Endpunkten für serverlose Cache-Ressourcen zu ermöglichen.	8. November 2023
ElastiCache hat begonnen, Änderungen zu verfolgen	ElastiCache hat begonnen, Änderungen für die AWS	7. Februar 2020

Änderungen	Beschreibung	Datum
	verwalteten Richtlinien zu verfolgen.	

Verwendung identitätsbasierter Richtlinien (IAM-Richtlinien) für Amazon ElastiCache

In diesem Thema finden Sie Beispiele für identitätsbasierte Richtlinien, in denen ein Kontoadministrator den IAM-Identitäten (Benutzer, Gruppen und Rollen) Berechtigungsrichtlinien anfügen kann.

Important

Wir empfehlen Ihnen, zunächst die Themen zu lesen, in denen die grundlegenden Konzepte und Optionen zur Verwaltung des Zugriffs auf ElastiCache Amazon-Ressourcen erläutert werden. Weitere Informationen finden Sie unter [Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre ElastiCache Ressourcen](#).

Dieses Thema besteht aus folgenden Abschnitten:

- [AWS verwaltete Richtlinien für Amazon ElastiCache](#)
- [Beispiele für vom Kunden verwaltete Richtlinien](#)

Im Folgenden finden Sie ein Beispiel für eine Berechtigungsrichtlinie bei der Verwendung von Redis OSS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowClusterPermissions",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache",
        "elasticache:CreateCacheCluster",

```

```

        "elasticache:DescribeServerlessCaches",
        "elasticache:DescribeReplicationGroups",
        "elasticache:DescribeCacheClusters",
        "elasticache:ModifyServerlessCache",
        "elasticache:ModifyReplicationGroup",
        "elasticache:ModifyCacheCluster"
    ],
    "Resource": "*"
},
{
    "Sid": "AllowUserToPassRole",
    "Effect": "Allow",
    "Action": [ "iam:PassRole" ],
    "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
}
]
}

```

Im Folgenden wird ein Beispiel für eine Berechtigungsrichtlinie bei der Verwendung von Memcached gezeigt.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowClusterPermissions",
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache",
            "elasticache:CreateCacheCluster",
            "elasticache:DescribeServerlessCaches",
            "elasticache:DescribeCacheClusters",
            "elasticache:ModifyServerlessCache",
            "elasticache:ModifyCacheCluster"
        ],
        "Resource": "*"
    },
    {
        "Sid": "AllowUserToPassRole",
        "Effect": "Allow",

```

```
"Action": [ "iam:PassRole" ],
"Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
}
]
}
```

Die Richtlinie enthält zwei Anweisungen:

- Die erste Anweisung erteilt Berechtigungen für die ElastiCache Amazon-Aktionen (`elasticache:Create*`, `elasticache:Describe*`, `elasticache:Modify*`)
- Die zweite Anweisung erteilt Berechtigungen für die IAM-Aktion (`iam:PassRole`) für den IAM-Rollenamen am Ende des Resource-Wertes.

Das Element `Principal` ist in der Richtlinie nicht angegeben, da in identitätsbasierten Richtlinien die Angabe des Prinzipals als Empfänger der Berechtigung nicht erforderlich ist. Wenn Sie einem Benutzer eine Richtlinie zuweisen, ist der Benutzer automatisch der Prinzipal. Wird die Berechtigungsrichtlinie einer IAM-Rolle angefügt, erhält der in der Vertrauensrichtlinie der Rolle angegebene Prinzipal die Berechtigungen.

Eine Tabelle mit allen ElastiCache Amazon-API-Aktionen und den Ressourcen, für die sie gelten, finden Sie unter [ElastiCache API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen](#).

Beispiele für vom Kunden verwaltete Richtlinien

Wenn Sie keine Standardrichtlinie verwenden und eine benutzerdefinierte verwaltete Richtlinie verwenden möchten, stellen Sie eines sicher. Sie sollten entweder die Genehmigung zum Aufrufen von `iam:createServiceLinkedRole` haben (weitere Informationen finden Sie unter [Beispiel 4: Erlauben Sie einem Benutzer, die IAM-API `CreateServiceLinkedRole` aufzurufen](#)). Oder Sie hätten eine ElastiCache dienstbezogene Rolle erstellen sollen.

In Kombination mit den Mindestberechtigungen, die für die Nutzung der ElastiCache Amazon-Konsole erforderlich sind, gewähren die Beispielrichtlinien in diesem Abschnitt zusätzliche Berechtigungen. Die Beispiele sind auch relevant für die AWS SDKs und die AWS CLI.

Anweisungen zum Einrichten von IAM-Benutzern und -Gruppen finden Sie unter [Erstellen Ihrer ersten IAM-Benutzer- und -Administratorengruppe](#) im IAM-Benutzerhandbuch.

⚠ Important

Testen Sie Ihre IAM-Richtlinien immer gründlich, bevor Sie sie in der Produktion verwenden. Einige ElastiCache Aktionen, die einfach erscheinen, können andere Aktionen erfordern, um sie zu unterstützen, wenn Sie die ElastiCache Konsole verwenden. `elasticache:CreateCacheCluster` Gewährt beispielsweise Berechtigungen zum Erstellen von ElastiCache Clustern. Um diesen Vorgang auszuführen, verwendet die ElastiCache Konsole jedoch eine Reihe von `Describe` `List` AND-Aktionen, um Konsolenlisten aufzufüllen.

Beispiele

- [Beispiel 1: Erlauben Sie einem Benutzer nur Lesezugriff auf Ressourcen ElastiCache](#)
- [Beispiel 2: Erlauben Sie einem Benutzer, allgemeine ElastiCache Systemadministrationsaufgaben auszuführen](#)
- [Beispiel 3: Erlauben Sie einem Benutzer den Zugriff auf alle ElastiCache API-Aktionen](#)
- [Beispiel 4: Erlauben Sie einem Benutzer, die IAM-API `CreateServiceLinkedRole` aufzurufen](#)
- [Beispiel 5: einem Benutzer erlauben, mithilfe der IAM-Authentifizierung eine Verbindung mit dem Serverless-Cache herzustellen](#)

Beispiel 1: Erlauben Sie einem Benutzer nur Lesezugriff auf Ressourcen ElastiCache

Die folgende Richtlinie gewährt ElastiCache Zugriffsaktionen, die es einem Benutzer ermöglichen, Ressourcen aufzulisten. Gewöhnlich ordnen Sie diese Art von Berechtigungsrichtlinie einer Gruppe von Managern zu.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECReadOnly",
    "Effect": "Allow",
    "Action": [
      "elasticache:Describe*",
      "elasticache:List*"
    ],
    "Resource": "*"
  }
]
```

```
}  
]  
}
```

Beispiel 2: Erlauben Sie einem Benutzer, allgemeine ElastiCache Systemadministrationsaufgaben auszuführen

Zu den allgemeinen Aufgaben des Systemadministrators gehört das Ändern von Ressourcen. Ein Systemadministrator möchte vielleicht auch Informationen zu den ElastiCache -Ereignissen anfordern. Die folgende Richtlinie gewährt einem Benutzer die Berechtigung, ElastiCache Aktionen für diese allgemeinen Systemadministrationsaufgaben auszuführen. Normalerweise ordnen Sie diese Art Berechtigungsrichtlinie der Systemadministratorengruppe zu.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Sid": "ECAAllowMutations",  
    "Effect": "Allow",  
    "Action": [  
      "elasticache:Modify*",  
      "elasticache:Describe*",  
      "elasticache:ResetCacheParameterGroup"  
    ],  
    "Resource": "*"   
  ]  
}
```

Beispiel 3: Erlauben Sie einem Benutzer den Zugriff auf alle ElastiCache API-Aktionen

Die folgende Richtlinie ermöglicht einem Benutzer den Zugriff auf alle ElastiCache Aktionen. Es wird empfohlen, diese Art von Berechtigungsrichtlinie nur einem Administratorbenutzer zu gewähren.

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [{
  "Sid": "ECAAllowAll",
  "Effect": "Allow",
  "Action": [
    "elasticache:*"
  ],
  "Resource": "*"
}]
}
```

Beispiel 4: Erlauben Sie einem Benutzer, die IAM-API `CreateServiceLinkedRole` aufzurufen

Die folgende Richtlinie ermöglicht dem Benutzer den Aufruf der IAM-`CreateServiceLinkedRole`-API. Wir empfehlen, dass Sie dem Benutzer, der ElastiCache mutative Operationen aufruft, diese Art von Berechtigungsrichtlinie gewähren.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateSLRAllows",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "elasticache.amazonaws.com"
        }
      }
    }
  ]
}
```

Beispiel 5: einem Benutzer erlauben, mithilfe der IAM-Authentifizierung eine Verbindung mit dem Serverless-Cache herzustellen

Die folgende Richtlinie ermöglicht es jedem Benutzer, mithilfe der IAM-Authentifizierung zwischen 01-04-2023 und 30-06-2023 eine Verbindung mit jedem beliebigen Serverless-Cache herzustellen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:*"
      ],
      "Condition": {
        "DateGreaterThan": {"aws:CurrentTime": "2023-04-01T00:00:00Z"},
        "DateLessThan": {"aws:CurrentTime": "2023-06-30T23:59:59Z"}
      }
    },
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:user:*"
      ]
    }
  ]
}
```

Berechtigungen auf Ressourcenebene

Sie können den Umfang der Berechtigungen einschränken, indem Sie Ressourcen in einer IAM-Richtlinie festlegen. Viele ElastiCache API-Aktionen unterstützen einen Ressourcentyp, der je nach Verhalten der Aktion variiert. Jede IAM-Richtlinienanweisung erteilt die Berechtigung für eine Aktion, die auf eine Ressource ausgeführt wird. Wenn die Aktion nicht auf eine benannte Ressource wirkt oder wenn Sie die Erlaubnis erteilen, die Aktion auf allen Ressourcen durchzuführen, ist der Wert

der Ressource in der Richtlinie ein Platzhalter (*). Für viele API-Aktionen können Sie die Ressourcen einschränken, die ein Benutzer ändern kann. Hierzu geben Sie den Amazon-Ressourcennamen (ARN) einer Ressource oder ein ARN-Muster an, das mehreren Ressourcen entspricht. Zum Einschränken von Berechtigungen nach Ressource bestimmen Sie die Ressource unter Angabe des ARN.

Eine Liste der ElastiCache Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Von Amazon definierte Ressourcen ElastiCache](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

Beispiele

- [Beispiel 1: Erlauben Sie einem Benutzer vollen Zugriff auf bestimmte ElastiCache Ressourcentypen](#)
- [Beispiel 2: Verweigern Sie einem Benutzer den Zugriff auf einen Serverless-Cache.](#)

Beispiel 1: Erlauben Sie einem Benutzer vollen Zugriff auf bestimmte ElastiCache Ressourcentypen

Die folgende Richtlinie erlaubt explizit alle Ressourcen vom Typ Serverless-Cache.

```
{
  "Sid": "Example1",
  "Effect": "Allow",
  "Action": "elasticache:*",
  "Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:*"
  ]
}
```

Beispiel 2: Verweigern Sie einem Benutzer den Zugriff auf einen Serverless-Cache.

Im folgenden Beispiel wird der Zugriff auf einen bestimmten Serverless-Cache explizit verweigert.

```
{
  "Sid": "Example2",
  "Effect": "Deny",
  "Action": "elasticache:*",
  "Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:name"
  ]
}
```



```
}
```

Verwenden von Bedingungsschlüssel

Sie können Bedingungen angeben, die bestimmen, wie eine IAM-Richtlinie wirksam wird. In können Sie das `Condition` Element einer JSON-Richtlinie verwenden ElastiCache, um Schlüssel im Anforderungskontext mit Schlüsselwerten zu vergleichen, die Sie in Ihrer Richtlinie angeben. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#).

Eine Liste der ElastiCache Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon ElastiCache](#) in der Service Authorization Reference.

Eine Liste der globalen Bedingungsschlüssel finden Sie unter [AWS Globale Bedingungskontextschlüssel](#).

Verwendung ElastiCache mit AWS globalen Bedingungsschlüsseln

Wenn Sie [AWS globale Bedingungsschlüssel](#) verwenden, für die ein [Principal](#) erforderlich ElastiCache ist, verwenden Sie eine OR Bedingung mit beiden Prinzipalen: `elasticache.amazonaws.com` und `undec.amazonaws.com`.

Note

Wenn Sie nicht beide Prinzipale für hinzufügen ElastiCache, wird die beabsichtigte Aktion „Zulassen“ oder „Verweigern“ für keine der in Ihrer Richtlinie aufgeführten Ressourcen korrekt durchgesetzt.

Beispiel für eine Richtlinie mit `aws:CalledVia` globalem Bedingungsschlüssel:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",

```

```
"Condition": {
  "ForAnyValue:StringLike": {
    "aws:CalledVia": [
      "ec.amazonaws.com",
      "elasticache.amazonaws.com"
    ]
  }
}
```

Festlegung von Bedingungen: Verwenden von Bedingungsschlüsseln

Um eine differenzierte Steuerung zu implementieren, schreiben Sie eine IAM-Berechtigungs-Richtlinie, die Bedingungen zum Steuern eines Satzes einzelner Parameter für bestimmte Anforderungen festlegt. Anschließend wenden Sie die Richtlinie für die IAM-Benutzer, -Gruppen oder -Rollen an, die Sie mithilfe der IAM-Konsole erstellen.

Um eine Bedingung anzuwenden, fügen Sie die Bedingungsinformationen der IAM-Richtlinienanweisung hinzu. Im folgenden Beispiel geben Sie die Bedingung an, dass jeder erstellte knotenbasierte Cluster vom Knotentyp ist. `cache.r5.large`

Note

- Um Condition Elemente mithilfe von Bedingungsschlüsseln des String Typs zu erstellen, verwenden Sie Bedingungsoperatoren, bei denen Groß- und Kleinschreibung nicht berücksichtigt wird `StringNotEqualsIgnoreCase`, `StringEqualsIgnoreCase` oder um einen Schlüssel mit einem Zeichenkettenwert zu vergleichen.
- ElastiCache verarbeitet die Eingabeargumente für `CacheNodeType` und ohne Berücksichtigung `CacheParameterGroupName` der Groß- und Kleinschreibung. Aus diesem Grund `StringNotEqualsIgnoreCase` sollten die String-Bedingungsoperatoren `StringEqualsIgnoreCase` und in Berechtigungsrichtlinien verwendet werden, die auf sie verweisen.

Im Folgenden wird ein Beispiel für diese Berechtigungsrichtlinie bei der Verwendung von Valkey oder Redis OSS gezeigt.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheNodeType": [
            "cache.r5.large"
          ]
        }
      }
    }
  ]
}
```

Im Folgenden wird ein Beispiel für diese Berechtigungsrichtlinie bei der Verwendung von Memcached gezeigt.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheNodeType": [
            "cache.r5.large"
          ]
        }
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).

Weitere Informationen zur Verwendung von Richtlinienbedingungsoperatoren finden Sie unter [ElastiCache API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen](#).

Beispielrichtlinien: Verwenden von Bedingungen für die differenzierte Parameterkontrolle

In diesem Abschnitt werden Beispielrichtlinien für die Implementierung einer differenzierten Zugriffskontrolle für die zuvor aufgeführten Parameter beschrieben. ElastiCache

1. `elasticache:MaximumDataStorage`: Geben Sie den maximalen Datenspeicher eines serverlosen Caches an. Unter den angegebenen Bedingungen kann der Kunde keine Caches erstellen, die mehr als eine bestimmte Datenmenge speichern können.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
      ],
      "Condition": {
        "NumericLessThanEquals": {
          "elasticache:MaximumDataStorage": "30"
        },
        "StringEquals": {
          "elasticache:DataStorageUnit": "GB"
        }
      }
    }
  ]
}
```

```
    ]
  }
}
```

2. `elasticache:MaximumECPUPerSecond`: Geben Sie den maximalen Wert für ECPU pro Sekunde für einen serverlosen Cache an. Unter den angegebenen Bedingungen kann der Kunde keine Caches erstellen, die mehr als eine bestimmte Anzahl von Caches pro Sekunde ausführen können. ECPUs

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
      ],
      "Condition": {
        "NumericLessThanEquals": {
          "elasticache:MaximumECPUPerSecond": "100000"
        }
      }
    }
  ]
}
```

3. `elasticache:CacheNodeType`: Geben Sie an, welche `NodeType` (s) ein Benutzer erstellen kann. Unter Verwendung der bereitgestellten Bedingungen kann der Kunde einen einzelnen Wert oder einen Bereichswert für einen Knotentyp angeben.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheNodeType": [
            "cache.t2.micro",
            "cache.t2.medium"
          ]
        }
      }
    }
  ]
}
```

4. `elasticache:CacheNodeType`: Geben Sie mit Memcached an, welche `NodeType` (s) ein Benutzer erstellen kann. Unter Verwendung der bereitgestellten Bedingungen kann der Kunde einen einzelnen Wert oder einen Bereichswert für einen Knotentyp angeben.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheNodeType": [
            "cache.t2.micro",
            "cache.t2.medium"
          ]
        }
      }
    }
  ]
}
```

5. `elasticache:NumNodeGroups`: Erstellt eine Replikationsgruppe mit weniger als 20 Knotengruppen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "NumericLessThanEquals": {
          "elasticache:NumNodeGroups": "20"
        }
      }
    }
  ]
}
```

6. elasticache:ReplicasPerNodeGroup: Geben Sie die Replikate pro Knoten zwischen 5 und 10 an.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```

        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "NumericGreaterThanEquals": {
            "elasticache:ReplicasPerNodeGroup": "5"
        },
        "NumericLessThanEquals": {
            "elasticache:ReplicasPerNodeGroup": "10"
        }
    }
}
]
}

```

7. elasticache:EngineVersion: Geben Sie die Verwendung der Engine-Version 5.0.6 an.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        }
    ]
}

```

```

    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*",
      "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
      "StringEquals": {
        "elasticache:EngineVersion": "5.0.6"
      }
    }
  }
]
}

```

8. elasticache:EngineVersion: Gibt die Verwendung der Memcached-Engine-Version 1.6.6 an
JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ]
    }
  ]
}

```

```

    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticache:EngineVersion": "1.6.6"
        }
    }
}
]
}

```

9. `elasticache:EngineType`: Geben Sie an, dass nur eine Valkey- oder Redis-OSS-Engine verwendet wird.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {

```

```

        "StringEquals": {
            "elasticache:EngineType": "redis"
        }
    }
}

```

10. `elasticache:AtRestEncryptionEnabled`: Geben Sie an, dass Replikationsgruppen nur bei aktivierter Verschlüsselung erstellt werden.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:AtRestEncryptionEnabled": "true"
                }
            }
        }
    ]
}

```

```
}
```

11Elasticache: TransitEncryptionEnabled

- a. Setzen Sie den `elasticache:TransitEncryptionEnabled` Bedingungsschlüssel `false` für die [CreateReplicationGroup](#)Aktion auf, um anzugeben, dass Replikationsgruppen nur erstellt werden können, wenn TLS nicht verwendet wird:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {
          "elasticache:TransitEncryptionEnabled": "false"
        }
      }
    }
  ]
}
```

Wenn der `elasticache:TransitEncryptionEnabled` Bedingungsschlüssel `false` in einer Richtlinie für die [CreateReplicationGroup](#)Aktion auf gesetzt ist, ist eine

CreateReplicationGroup Anfrage nur zulässig, wenn TLS nicht verwendet wird (d. h. wenn die Anfrage keinen Parameter enthält, der auf gesetzt ist true oder einen TransitEncryptionEnabled TransitEncryptionMode Parameter, der auf gesetzt ist)required.

- b. Setzen Sie den elasticache:TransitEncryptionEnabled Bedingungsschlüssel auf true für die [CreateReplicationGroup](#)Aktion, um anzugeben, dass Replikationsgruppen nur erstellt werden können, wenn TLS verwendet wird:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {
          "elasticache:TransitEncryptionEnabled": "true"
        }
      }
    }
  ]
}
```

Wenn der `elasticache:TransitEncryptionEnabled` Bedingungsschlüssel `true` in einer Richtlinie für die [CreateReplicationGroup](#)-Aktion auf gesetzt ist, ist eine `CreateReplicationGroup` Anfrage nur zulässig, wenn die Anforderung einen Parameter enthält, der auf gesetzt ist, `true` und einen `TransitEncryptionEnabled` Parameter, der auf gesetzt ist. `required`

- c. Legen Sie `elasticache:TransitEncryptionEnabled` für die `ModifyReplicationGroup`-Aktion auf `true` fest, um anzugeben, dass Replikationsgruppen nur geändert werden können, wenn TLS verwendet wird:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:ModifyReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "BoolIfExists": {
          "elasticache:TransitEncryptionEnabled": "true"
        }
      }
    }
  ]
}
```

Wenn der `elasticache:TransitEncryptionEnabled` Bedingungsschlüssel `true` in einer Richtlinie für die [ModifyReplicationGroup](#)-Aktion auf gesetzt ist, ist eine `ModifyReplicationGroup` Anfrage nur zulässig, wenn die Anforderung einen `TransitEncryptionMode` Parameter enthält, der auf gesetzt ist `required`. Der `TransitEncryptionEnabled`-Parameter, der auf `true` festgelegt ist, kann optional ebenfalls enthalten sein, ist in diesem Fall jedoch nicht erforderlich, um TLS zu aktivieren.

12. `elasticache:AutomaticFailoverEnabled`: Geben Sie an, dass Replikationsgruppen nur bei aktiviertem automatischem Failover erstellt werden.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {
          "elasticache:AutomaticFailoverEnabled": "true"
        }
      }
    }
  ]
}
```

13elasticache:MultiAZEnabled: Geben Sie an, dass Replikationsgruppen nicht erstellt werden können, wenn Multi-AZ deaktiviert ist.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

        "Action": [
            "elasticache:CreateCacheCluster",
            "elasticache:CreateReplicationGroup"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:parametergroup:*",
            "arn:aws:elasticache:*:*:subnetgroup:*"
        ]
    },
    {
        "Effect": "Deny",
        "Action": [
            "elasticache:CreateCacheCluster",
            "elasticache:CreateReplicationGroup"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:cluster:*",
            "arn:aws:elasticache:*:*:replicationgroup:*"
        ],
        "Condition": {
            "Bool": {
                "elasticache:MultiAZEnabled": "false"
            }
        }
    }
]
}

```

14. `elasticache:ClusterModeEnabled`: Geben Sie an, dass Replikationsgruppen nur bei aktiviertem Clustermodus erstellt werden können.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        }
    ]
}

```

```

    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
      "Bool": {
        "elasticache:ClusterModeEnabled": "true"
      }
    }
  }
]
}

```

15. `elasticache:AuthTokenEnabled`: Geben Sie an, dass Replikationsgruppen nur mit aktiviertem AUTH-Token erstellt werden können.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [

```

```

        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "Bool": {
            "elasticache:AuthTokenEnabled": "true"
        }
    }
}
]
}

```

16. `elasticache:SnapshotRetentionLimit`: Geben Sie die Anzahl der Tage (oder min/max.) an, für die der Snapshot aufbewahrt werden soll. Die folgende Richtlinie erzwingt die Speicherung von Backups für mindestens 30 Tage.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup",
                "elasticache:CreateServerlessCache"
            ]
        }
    ]
}

```

```

    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*",
        "arn:aws:elasticache:*:*:serverlesscache:*"
    ],
    "Condition": {
        "NumericGreaterThanEquals": {
            "elasticache:SnapshotRetentionLimit": "30"
        }
    }
}

```

17 elasticache:KmsKeyId: Geben Sie die Verwendung von vom Kunden verwalteten KMS-Schlüsseln an.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
      ],
      "Condition": {

```

```

        "StringEquals": {
            "elasticache:KmsKeyId": "my-key"
        }
    }
}
]
}

```

18. `elasticache:CacheParameterGroupName`: Geben Sie eine nicht standardmäßige Parametergruppe mit spezifischen Parametern von einer Organisation in Ihren Clustern an. Sie können auch ein Benennungsmuster für Ihre Parametergruppen angeben oder das Löschen eines bestimmten Parametergruppennamens blockieren. Im Folgenden finden Sie ein Beispiel, das die Verwendung von nur "my-org-param-group" einschränkt.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {

```

```

        "StringEquals": {
            "elasticache:CacheParameterGroupName": "my-org-param-group"
        }
    }
}
]
}

```

19. **elasticache:CacheParameterGroupName:** Geben Sie mit Memcached eine nicht standardmäßige Parametergruppe mit spezifischen Parametern von einer Organisation in Ihren Clustern an. Sie können auch ein Benennungsmuster für Ihre Parametergruppen angeben oder das Löschen eines bestimmten Parametergruppennamens blockieren. Im Folgenden finden Sie ein Beispiel, das die Verwendung von nur "" einschränkt. my-org-param-group

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheParameterGroupName": "my-org-param-group"
        }
      }
    }
  ]
}

```

```

    }
  }
]
}

```

20elasticacheCreateCacheCluster: CreateCacheCluster Aktion wird verweigert, wenn das Anforderungs-Tag Project fehlt oder nicht gleich oder ist. Dev QA Prod

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*",
        "arn:aws:elasticache:*:*:securitygroup:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/Project": "true"
        }
      }
    }
  ],
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:AddTagsToResource"
    ]
  }
}

```



```

    ],
    "Resource": "arn:aws:elasticache:*:*:cluster:*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Project": [
                "Dev",
                "Prod",
                "QA"
            ]
        }
    }
}

```

21 elasticache:CacheNodeType: Zulassen CreateCacheCluster mit cacheNodeType cache.r5.large oder cache.r6g.4xlarge und Tag. Project=XYZ

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEqualsIfExists": {

```

```

        "elasticache:CacheNodeType": [
            "cache.r5.large",
            "cache.r6g.4xlarge"
        ]
    },
    "StringEquals": {
        "aws:RequestTag/Project": "XYZ"
    }
}
}
]
}

```

22elasticache:: Zulassen mit cache.r5.large oder cache.r6g.4xlarge und Tag. CacheNodeType
CreateCacheCluster cacheNodeType Project=XYZ

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEqualsIfExists": {
          "elasticache:CacheNodeType": [
            "cache.r5.large",
            "cache.r6g.4xlarge"
          ]
        }
      }
    }
  ]
}

```

```
    ]
  },
  "StringEquals": {
    "aws:RequestTag/Project": "XYZ"
  }
}
```

Note

Beim Erstellen von Richtlinien zum gemeinsamen Erzwingen von Tags und anderen Bedingungsschlüsseln kann die Bedingung `IfExists` aufgrund der zusätzlichen `elasticache:AddTagsToResource`-Richtlinienanforderungen für Erstellungsanforderungen mit dem `--tags`-Parameter für Bedingungsschlüsselemente erforderlich sein.

Verwenden von serviceverknüpften Rollen für Amazon ElastiCache

Amazon ElastiCache verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte Rollen](#). Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit einem AWS Service wie Amazon verknüpft ist. ElastiCache Rollen, die mit dem ElastiCache Service von Amazon verknüpft sind, werden von Amazon ElastiCache vordefiniert. Sie enthalten alle Berechtigungen, die der Dienst benötigt, um AWS -Dienste im Namen Ihrer Cluster aufzurufen.

Eine serviceverknüpfte Rolle ElastiCache erleichtert die Einrichtung von Amazon, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Die Rollen sind bereits in Ihrem AWS Konto vorhanden, sind jedoch mit ElastiCache Amazon-Anwendungsfällen verknüpft und verfügen über vordefinierte Berechtigungen. Nur Amazon ElastiCache kann diese Rollen übernehmen, und nur diese Rollen können die vordefinierten Berechtigungsrichtlinien verwenden. Sie können die Rollen nur nach dem Löschen der zugehörigen Ressourcen löschen. Dies schützt Ihre ElastiCache Amazon-Ressourcen, da Sie die für den Zugriff auf die Ressourcen erforderlichen Berechtigungen nicht versehentlich entfernen können.

Informationen zu anderen Services, die serviceverknüpften Rollen unterstützen, finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Ja in der Spalte

Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Inhalt

- [Servicebezogene Rollenberechtigungen für Amazon ElastiCache](#)
 - [Berechtigungen zum Erstellen einer serviceverknüpften Rolle](#)
- [Erstellen einer serviceverknüpften Rolle \(IAM\)](#)
 - [Erstellen einer serviceverknüpften Rolle \(IAM-Konsole\)](#)
 - [Erstellen einer serviceverknüpften Rolle \(IAM-CLI\)](#)
 - [Erstellen einer serviceverknüpften Rolle \(IAM-API\)](#)
- [Die Beschreibung einer serviceverknüpften Rolle für Amazon bearbeiten ElastiCache](#)
 - [Bearbeiten der Beschreibung einer serviceverknüpften Rolle \(IAM-Konsole\)](#)
 - [Bearbeiten der Beschreibung einer serviceverknüpften Rolle \(IAM-CLI\)](#)
 - [Bearbeiten der Beschreibung einer serviceverknüpften Rolle \(IAM-API\)](#)
- [Löschen einer serviceverknüpften Rolle für Amazon ElastiCache](#)
 - [Bereinigen einer serviceverknüpften Rolle](#)
 - [Löschen einer serviceverknüpften Rolle \(IAM-Konsole\)](#)
 - [Löschen einer serviceverknüpften Rolle \(IAM-CLI\)](#)
 - [Löschen einer serviceverknüpften Rolle \(IAM-API\)](#)

Servicebezogene Rollenberechtigungen für Amazon ElastiCache

Berechtigungen zum Erstellen einer serviceverknüpften Rolle

Um einer IAM-Entität zu ermöglichen, serviceverknüpfte Rollen zu erstellen AWS ServiceRoleForElastiCache

Die folgende Berechtigungsanweisung zu den Berechtigungen für diese IAM-Entität hinzufügen:

```
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole",
    "iam:PutRolePolicy"
  ]
}
```

```
"Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWS  
ServiceRoleForElastiCache*",  
"Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}  
}
```

Um einer IAM-Entität das Löschen einer dienstverknüpften Rolle zu ermöglichen AWS ServiceRoleForElastiCache

Die folgende Berechtigungsanweisung zu den Berechtigungen für diese IAM-Entität hinzufügen:

```
{  
  "Effect": "Allow",  
  "Action": [  
    "iam:DeleteServiceLinkedRole",  
    "iam:GetServiceLinkedRoleDeletionStatus"  
  ],  
  "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWS  
ServiceRoleForElastiCache*",  
  "Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}  
}
```

Alternativ können Sie eine AWS verwaltete Richtlinie verwenden, um vollen Zugriff auf Amazon zu gewähren ElastiCache.

Erstellen einer serviceverknüpften Rolle (IAM)

Sie können eine serviceverknüpfte Rolle mithilfe der IAM-Konsole, der CLI oder API erstellen.

Erstellen einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Erstellen einer serviceverknüpften Rolle verwenden.

So erstellen Sie eine serviceverknüpfte Rolle (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus. Klicken Sie auf Create New Role (Neue Rolle erstellen).
3. Wählen Sie unter Select type of trusted entity (Typ der vertrauenswürdigen Entität auswählen) die Option AWS -Service aus.

4. Wählen Sie unter Oder wählen Sie einen Dienst aus, um dessen Anwendungsfälle anzuzeigen die Option ElastiCache.
5. Wählen Sie Weiter: Berechtigungen aus.
6. Beachten Sie unter Richtlinienname, dass `ElastiCacheServiceRolePolicy` für diese Rolle erforderlich ist. Wählen Sie Weiter: Tags aus.
7. Beachten Sie, dass Tags für serviceverknüpfte Rollen nicht unterstützt werden. Klicken Sie auf Next:Review (Weiter: Prüfen).
8. (Optional:) Bearbeiten Sie in Role description die Beschreibung für die neue serviceverknüpfte Rolle.
9. Prüfen Sie die Rolle und klicken Sie dann auf Create Role (Rolle erstellen).

Erstellen einer serviceverknüpften Rolle (IAM-CLI)

Sie können IAM-Operationen von verwenden, AWS Command Line Interface um eine serviceverknüpfte Rolle zu erstellen. Diese Rolle kann die Vertrauensrichtlinie, sowie die enthaltenen Richtlinien enthalten, die der Service für die Zuweisung der Rolle benötigt.

So erstellen Sie eine serviceverknüpfte Rolle (CLI)

Führen Sie die folgenden Operationen aus:

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

Erstellen einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API für das Erstellen einer serviceverknüpften Rolle verwenden. Diese Rolle kann die Vertrauensrichtlinie, sowie die enthaltenen Richtlinien enthalten, die der Service für die Zuweisung der Rolle benötigt.

So erstellen Sie eine serviceverknüpfte Rolle (API)

Verwenden Sie den [CreateServiceLinkedRole](#)-API-Aufruf. Geben Sie in der Anforderung einen Servicenamen im Format `elasticache.amazonaws.com` an.

Die Beschreibung einer serviceverknüpften Rolle für Amazon bearbeiten ElastiCache

Amazon ElastiCache erlaubt Ihnen nicht, die AWS ServiceRoleForElastiCache 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.

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So bearbeiten Sie die Beschreibung einer serviceverknüpften Rolle (Konsole)

1. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus.
2. Wählen Sie den Namen der zu ändernden Rolle.
3. Wählen Sie neben Role description ganz rechts Edit.
4. Geben Sie eine neue Beschreibung im Dialogfeld ein und klicken Sie auf Save (Speichern).

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-CLI)

Sie können IAM-Operationen vom verwenden, um eine Beschreibung einer serviceverknüpften Rolle AWS Command Line Interface zu bearbeiten.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (CLI)

1. (Optional) Verwenden Sie den Vorgang for IAM, um die aktuelle Beschreibung AWS CLI für eine Rolle anzuzeigen. [get-role](#)

Example

```
$ aws iam get-role --role-name AWS ServiceRoleForElastiCache
```

Verwenden Sie den Rollennamen, nicht den ARN, um sich auf Rollen mit den CLI-Operationen zu beziehen. Wenn eine Rolle zum Beispiel folgenden ARN hat: `arn:aws:iam::123456789012:role/myrole`, verweisen Sie auf die Rolle als **myrole**.

2. Verwenden Sie den Vorgang AWS CLI for IAM, um die Beschreibung einer serviceverknüpften Rolle zu aktualisieren. [update-role-description](#)

Für Linux, macOS oder Unix:

```
$ aws iam update-role-description \  
    --role-name AWS ServiceRoleForElastiCache \  
    --description "New description"
```

```
--description "new description"
```

Für Windows:

```
$ aws iam update-role-description ^  
    --role-name AWS ServiceRoleForElastiCache ^  
    --description "new description"
```

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (API)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie die IAM-API-Operation [GetRole](#).

Example

```
https://iam.amazonaws.com/  
?Action=GetRole  
&RoleName=AWS ServiceRoleForElastiCache  
&Version=2010-05-08  
&AUTHPARAMS
```

2. Um die Beschreibung einer Rolle zu aktualisieren, verwenden Sie die IAM-API-Operation [UpdateRoleDescription](#).

Example

```
https://iam.amazonaws.com/  
?Action=UpdateRoleDescription  
&RoleName=AWS ServiceRoleForElastiCache  
&Version=2010-05-08  
&Description="New description"
```


Löschen einer serviceverknüpften Rolle für Amazon ElastiCache

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 Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie löschen können.

Amazon ElastiCache löscht die serviceverknüpfte Rolle nicht für Sie.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie eine serviceverknüpfte Rolle mit IAM löschen können, bestätigen Sie zunächst, dass der Rolle keine Ressourcen (Cluster oder Replikationsgruppen) zugeordnet sind.

So überprüfen Sie in der IAM-Konsole, ob die serviceverknüpfte Rolle über eine aktive Sitzung verfügt

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter.
<https://console.aws.amazon.com/iam/>
2. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus. Wählen Sie dann den Namen (nicht das Kontrollkästchen) der AWS ServiceRoleForElastiCache Rolle aus.
3. Wählen Sie auf der Seite Summary für die ausgewählte Rolle die Registerkarte Access Advisor.
4. Überprüfen Sie auf der Registerkarte Access Advisor die jüngsten Aktivitäten für die serviceverknüpfte Rolle.

Um ElastiCache Amazon-Ressourcen zu löschen, die AWS ServiceRoleForElastiCache

- Informationen zum Löschen eines Clusters finden Sie unter:
 - [Mit dem AWS-Managementkonsole](#)
 - [Verwenden Sie den AWS CLI, um einen ElastiCache Cluster zu löschen](#)
 - [Verwenden der ElastiCache API](#)
- Informationen zum Löschen einer Replikationsgruppe finden Sie unter:
 - [Löschen einer Replikationsgruppe \(Konsole\)](#)
 - [Löschen einer Replikationsgruppe \(AWS CLI\)](#)
 - [Löschen einer Replikationsgruppe \(ElastiCache API\)](#)

Löschen einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich der IAM Console Roles aus. Aktivieren Sie dann das Kontrollkästchen neben dem Rollennamen, den Sie löschen möchten, nicht den Namen oder die Zeile selbst.
3. Wählen Sie für Role actions oben auf der Seite Delete role aus.
4. Überprüfen Sie im Bestätigungsdialogfeld die Daten, auf die der Dienst zuletzt zugegriffen hat. Aus diesen Daten geht hervor, wann jede der ausgewählten Rollen zuletzt auf einen AWS Dienst zugegriffen hat. Auf diese Weise können Sie leichter bestätigen, ob die Rolle derzeit aktiv ist. Wenn Sie fortfahren möchten, wählen Sie Yes, Delete aus, um die serviceverknüpfte Rolle zur Löschung zu übermitteln.
5. Sehen Sie sich die Benachrichtigungen in der IAM-Konsole an, um den Fortschritt der Löschung der serviceverknüpften Rolle zu überwachen. Da die Löschung der serviceverknüpften IAM-Rolle asynchron erfolgt, kann die Löschung nach dem Übermitteln der Rolle für die Löschung erfolgreich sein oder fehlschlagen. Wenn der Vorgang fehlschlägt, können Sie in den Benachrichtigungen View details oder View Resources auswählen, um zu erfahren, warum die Löschung fehlgeschlagen ist.

Löschen einer serviceverknüpften Rolle (IAM-CLI)

Sie können IAM-Operationen von verwenden, AWS Command Line Interface um eine dienstverknüpfte Rolle zu löschen.

So löschen Sie eine serviceverknüpfte Rolle (CLI)

1. Wenn Sie den Namen der serviceverknüpften Rolle, die Sie löschen möchten, nicht kennen, geben Sie den folgenden Befehl ein. Dieser Befehl listet die Rollen und ihre Amazon-Ressourcennamen (ARNs) in Ihrem Konto auf.

```
$ aws iam get-role --role-name role-name
```

Verwenden Sie den Rollennamen, nicht den ARN, um sich auf Rollen mit den CLI-Operationen zu beziehen. Wenn eine Rolle zum Beispiel den ARN `arn:aws:iam::123456789012:role/myrole` hat, verweisen Sie auf die Rolle als **myrole**.

2. Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschungsanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die `deletion-task-id` aus der Antwort, um den Status der Löschaufgabe zu überprüfen. Geben Sie Folgendes ein, um eine Anforderung zum Löschen einer serviceverknüpften Rolle abzusenden.

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. Geben Sie Folgendes ein, um den Status der Löschaufgabe zu überprüfen.

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Der Status der Löschaufgabe kann `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` oder `FAILED` lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

Löschen einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API zum Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (API)

1. Um eine Löschanfrage für eine serviceverknüpfte Rolle zu übermitteln, rufen Sie [DeleteServiceLinkedRole](#) auf. Geben Sie in der Anforderung einen Rollennamen an.

Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschungsanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die `DeletionTaskId` aus der Antwort, um den Status der Löschaufgabe zu überprüfen.

2. Um den Status der Löschung zu überprüfen, rufen Sie [GetServiceLinkedRoleDeletionStatus](#) auf. Geben Sie in der Anforderung die `DeletionTaskId` an.

Der Status der Löschaufgabe kann NOT_STARTED, IN_PROGRESS, SUCCEEDED oder FAILED lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

ElastiCache API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen

Wenn Sie Richtlinien für [Zugriffskontrolle](#) und Schreibberechtigungen einrichten, die an eine IAM-Richtlinie angehängt werden sollen (entweder identitäts- oder ressourcenbasiert), verwenden Sie die folgende Tabelle als Referenz. In der Tabelle sind alle ElastiCache Amazon-API-Operationen und die entsprechenden Aktionen aufgeführt, für die Sie Berechtigungen zur Ausführung der Aktion erteilen können. Sie geben die Aktionen im Feld `Action` der Richtlinie und einen Ressourcenwert im Feld `Resource` der Richtlinie an. Sofern nicht anders angegeben, ist die Ressource erforderlich. Einige Felder enthalten sowohl eine erforderliche Ressource als auch optionale Ressourcen. Wenn kein Ressourcen-ARN vorhanden ist, ist die Ressource in der Richtlinie als Platzhalter (*) dargestellt.

Sie können Bedingungsschlüssel in Ihren ElastiCache Richtlinien verwenden, um Bedingungen auszudrücken. Eine Liste mit ElastiCache spezifischen Bedingungsschlüsseln sowie den Aktionen und Ressourcentypen, für die sie gelten, finden Sie unter [Verwenden von Bedingungsschlüssel](#). Eine vollständige Liste der AWS-weiten Schlüssel finden Sie im IAM-Benutzerhandbuch unter [Kontext-Schlüssel für AWS globale Bedingungen](#).

Note

Um eine Aktion anzugeben, verwenden Sie das Präfix `elasticache:` gefolgt vom Namen der API-Operation (z. B. `elasticache:DescribeCacheClusters`).

Eine Liste der ElastiCache Aktionen finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#) in der Service Authorization Reference.

Konformitätsvalidierung für Amazon ElastiCache

Externe Prüfer bewerten die Sicherheit und Konformität von AWS Services im Rahmen mehrerer AWS Compliance-Programme wie SOC, PCI, FedRAMP und HIPAA.

Informationen darüber, ob ein in den Geltungsbereich bestimmter Compliance-Programme AWS-Service fällt, finden Sie unter [AWS-Services Umfang nach Compliance-Programm](#) unter [Umfang nach Compliance-Programm](#) AWS-Services. Wählen Sie aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#).

Sie können Prüfberichte von Drittanbietern unter [herunterladen AWS Artifact](#). Weitere Informationen finden Sie unter [Berichte herunterladen unter](#).

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der [AWS Sicherheitsdokumentation](#).

- Wenn Amazon als Teil eines Compliance-Programms aufgeführt ElastiCache ist, gilt dieser Geltungsbereich für alle Cache-Engines, die von unterstützt werden ElastiCache. Dies gilt auch ElastiCache für Valkey, ElastiCache für Memcached und ElastiCache für Redis OSS.

Weitere Informationen

Allgemeine Informationen zur AWS Cloud-Compliance finden Sie im Folgenden:

- [FIPS-Endpunkte nach Service](#)
- [Service-Updates in ElastiCache](#)
- [AWS Cloud-Konformität](#)
- [Modell der übergreifenden Verantwortlichkeit](#)
- [AWS PCI DSS-Compliance-Programm](#)

Resilienz bei Amazon ElastiCache

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](#).

Zusätzlich zur AWS globalen Infrastruktur ElastiCache bietet Amazon mehrere Funktionen, um Ihre Datenstabilität und Backup-Anforderungen zu erfüllen.

Themen

- [Minimieren von Ausfällen](#)

Minimieren von Ausfällen

Bei der Planung Ihrer ElastiCache Amazon-Implementierung sollten Sie so planen, dass Ausfälle nur minimale Auswirkungen auf Ihre Anwendung und Daten haben. In diesem Abschnitt werden verschiedene Ansätze vorgestellt, mit denen Sie Ihre Anwendung und Ihre Daten vor Ausfällen schützen können.

Themen

- [Minimieren von Ausfällen mit Memcached](#)
- [Minimierung von Fehlern beim Ausführen von Valkey oder Redis OSS](#)
- [Empfehlungen](#)

Minimieren von Ausfällen mit Memcached

Wenn Sie die Memcached-Engine verwenden, haben Sie folgende Möglichkeiten, um die Auswirkungen von Ausfällen möglichst gering zu halten. Es gibt zwei Arten von Ausfällen, die es zu berücksichtigen gilt: Knotenausfälle und Ausfälle einzelner Availability Zones.

Minimieren von Knotenausfällen

Serverless-Caches minimieren Knotenausfälle automatisch mit einer replizierten Multi-AZ-Architektur, sodass Knotenausfälle für Ihre Anwendung transparent sind. Um die Auswirkungen eines Knotenausfalls in einem knotenbasierten Cluster zu minimieren, verteilen Sie Ihre zwischengespeicherten Daten auf mehrere Knoten. Da knotenbasierte Cluster die Replikation nicht unterstützen, führt ein Knotenausfall immer zu einem gewissen Datenverlust in Ihrem Cluster.

Wenn Sie Ihren Memcached-Cluster erstellen, können Sie ihn mit 1 bis 60 Knoten oder auf Anfrage auch mit mehr Knoten erstellen. Wenn Sie Ihre Daten über mehrere Knoten partitionieren, gehen bei einem Knotenausfall weniger Daten verloren. Angenommen, Sie partitionieren Ihre Daten über 10 Knoten und ein einzelner Knoten speichert daher etwa 10 % der Cache-Daten. Bei einem Knotenausfall gehen in diesem Fall etwa 10 % Ihres Caches verloren und muss ersetzt werden, wenn ein neuer Knoten erstellt und bereitgestellt wird. Wenn dieselben Daten auf drei größere Knoten verteilt werden, gehen dagegen bei einem Knotenausfall etwa 33 % Ihrer Cache-Daten verloren.

Weitere Informationen zum Festlegen der Knotenanzahl in einem Memcached-Cluster erhalten Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

Minimieren von Ausfällen einer Availability Zone

Serverless-Caches minimieren Ausfälle von Availability Zones automatisch mit einer replizierten Multi-AZ-Architektur, sodass AZ-Ausfälle für Ihre Anwendung transparent sind.

Um die Auswirkungen eines Ausfalls der Availability Zone in einem knotenbasierten Cluster zu minimieren, platzieren Sie Ihre Knoten in so vielen Availability Zones wie möglich. Im unwahrscheinlichen Fall eines AZ-Fehlers verlieren Sie die in dieser AZ zwischengespeicherten Daten, nicht die in der anderen. AZs

Warum so viele Knoten?

Wenn meine Region nur 3 Availability Zones bereitstellt, warum sollte ich dann mehr als 3 Knoten erstellen, da im Fall eines AZ-Ausfalls sowieso etwa ein Drittel meiner Daten verloren geht?

Das ist eine sehr gute Frage. Dabei gilt es zu bedenken, dass wir zwei unterschiedliche Ausfalltypen minimieren wollen, Ausfälle einzelner Knoten und Availability Zones. Wenn Daten auf mehrere Availability Zones verteilt werden und eine der Zonen ausfällt, gehen nur die Cache-Daten dieser AZ verloren, unabhängig von der Anzahl der Knoten. Wenn jedoch ein Knoten ausfällt, gehen weniger Daten verloren, je mehr Knoten Sie verwenden.

Es gibt keine "Zauberformel", anhand derer sich die beste Anzahl an Knoten pro Cluster bestimmen lässt. Wägen Sie die Auswirkungen von Datenverlusten gegen die Wahrscheinlichkeit eines Ausfalls und die Kosten ab und treffen Sie anhand dieser Faktoren eine Entscheidung.

Weitere Informationen zum Festlegen der Knotenanzahl in einem Memcached-Cluster erhalten Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

Weitere Informationen zu Regionen und Availability Zones finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

Minimierung von Fehlern beim Ausführen von Valkey oder Redis OSS

Wenn Sie eine Valkey- oder Redis OSS-Engine ausführen, haben Sie die folgenden Optionen, um die Auswirkungen eines Knoten- oder Availability Zone-Ausfalls zu minimieren.

Minimieren von Knotenausfällen

Serverless-Caches minimieren Knotenausfälle automatisch mit einer Multi-AZ-Architektur, sodass Knotenausfälle für Ihre Anwendung transparent sind. Knotenbasierte Cluster müssen entsprechend konfiguriert werden, um den Ausfall eines einzelnen Knotens zu minimieren.

Um die Auswirkungen von Valkey- oder Redis OSS-Knotenausfällen auf knotenbasierte Cluster zu minimieren, haben Sie die folgenden Optionen:

Themen

- [Behebung von Ausfällen: Valkey- oder Redis OSS-Replikationsgruppen](#)

Behebung von Ausfällen: Valkey- oder Redis OSS-Replikationsgruppen

Eine Valkey- oder Redis OSS-Replikationsgruppe besteht aus einem einzigen primären Knoten, von dem Ihre Anwendung sowohl lesen als auch schreiben kann, sowie aus 1 bis 5 schreibgeschützten Replikatknoten. Wenn Daten in den Primärknoten geschrieben werden, werden diese asynchron auf die Lesereplikat-Knoten aktualisiert.

Wenn ein Lesereplikat ausfällt,

1. ElastiCache erkennt das fehlgeschlagene Lesereplikat.
2. ElastiCache schaltet den ausgefallenen Knoten aus.
3. ElastiCache startet und stellt einen Ersatzknoten in derselben AZ bereit.
4. Der neue Knoten wird mit dem primären Knoten synchronisiert.

Währenddessen kann die Anwendung weiterhin Lese- und Schreibvorgänge auf den anderen Knoten ausführen.

Valkey oder Redis OSS Multi-AZ

Sie können Multi-AZ in Ihren Valkey- oder Redis OSS-Replikationsgruppen aktivieren. Unabhängig davon, ob Sie Multi-AZ aktivieren oder nicht, werden Ausfälle des primären Knotens erkannt und dieser wird automatisch ersetzt. Der genaue Vorgang ist dabei abhängig davon, ob Multi-AZ aktiviert ist.

Wenn Multi-AZ aktiviert ist

1. ElastiCache erkennt den Ausfall des Primärknotens.
2. ElastiCache stuft den Read-Replica-Knoten mit der geringsten Replikationsverzögerung zum Primärknoten hoch.
3. Die anderen Replikate synchronisieren sich mit dem neuen primären Knoten.
4. ElastiCache erstellt eine Read Replica in der AZ des ausgefallenen Primärservers.

5. Der neue Knoten synchronisiert sich mit dem neu ernannten primären Knoten.

Das Failover zu einem Replikationsknoten erfolgt in der Regel schneller als das Erstellen und Bereitstellen eines neuen primären Knotens. Dadurch kann Ihre Anwendung schneller wieder auf den primären Knoten schreiben, als wenn Multi-AZ nicht aktiviert ist.

Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS](#).

Wenn Multi-AZ deaktiviert ist

1. ElastiCache erkennt einen primären Fehler.
2. ElastiCache schaltet den Primärserver offline.
3. ElastiCache erstellt einen neuen Primärknoten und stellt ihn bereit, um den ausgefallenen Primärknoten zu ersetzen.
4. ElastiCache synchronisiert den neuen Primärserver mit einem der vorhandenen Replikate.
5. Nach der Synchronisierung dient der neue Knoten als primärer Knoten des Clusters.

Während der Ausführung der Schritte 1 bis 4 dieses Vorgangs können keine Daten in den primären Knoten geschrieben werden. Die Anwendung kann jedoch weiterhin Lesezugriffe auf den Replikationsknoten ausführen.

Für zusätzlichen Schutz empfehlen wir, die Knoten in Ihrer Replikationsgruppe in verschiedenen Availability Zones () AZs zu starten. Dadurch wirken sich Ausfälle einer AZ nur auf die Knoten in dieser AZ aus.

Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

Minimieren von Ausfällen einer Availability Zone

Serverless-Caches minimieren Ausfälle von Availability Zones automatisch mit einer replizierten Multi-AZ-Architektur, sodass AZ-Ausfälle für Ihre Anwendung transparent sind.

Um die Auswirkungen eines Ausfalls der Availability Zone in einem knotenbasierten Cluster zu minimieren, suchen Sie Ihre Knoten für jeden Shard in so vielen Availability Zones wie möglich.

Unabhängig von der Anzahl der Knoten in einem Shard führt ein katastrophaler Ausfall einer Availability Zone zu einem vollständigen Datenverlust Ihres Shards, wenn Sie Ihre Daten in nur einer

Availability Zone speichern. Wenn Sie Ihre Knoten jedoch in mehreren Zonen lokalisieren AZs, führt ein Ausfall einer beliebigen AZ dazu, dass Sie nur die Knoten in dieser AZ verlieren.

Bei einem Knotenausfall kann es zu einem Leistungsabfall kommen, da sich die Lesevorgänge nun auf weniger Knoten verteilen. Dieser Leistungsabfall bleibt bestehen, bis der ausgefallene Knoten ersetzt wurde.

Informationen zur Angabe der Availability Zones für Valkey- oder Redis OSS-Knoten finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)

Weitere Informationen zu Regionen und Availability Zones finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

Empfehlungen

Wir empfehlen, serverlose Caches über knotenbasierten Clustern zu erstellen, da Sie ohne zusätzliche Konfiguration automatisch eine bessere Fehlertoleranz erzielen. Bei der Erstellung eines knotenbasierten Clusters gibt es jedoch zwei Arten von Ausfällen, mit denen Sie rechnen müssen: Ausfälle einzelner Knoten und Ausfälle in der breiten Availability Zone. Um Datenverluste durch Ausfälle möglichst gering zu halten, sollten Sie beiden Arten von Ausfällen vorbeugen.

Minimieren der Auswirkungen von Knotenausfällen

Um die Auswirkungen eines Knotenausfalls bei der Verwendung von Valkey oder Redis OSS zu minimieren, empfehlen wir, dass Ihre Implementierung mehrere Knoten in jedem Shard verwendet und die Knoten auf mehrere Availability Zones verteilt. Dies erfolgt automatisch für Serverless-Caches.

Für knotenbasierte Cluster auf Valkey oder Redis OSS empfehlen wir, Multi-AZ in Ihrer Replikationsgruppe zu aktivieren, sodass bei einem Ausfall des primären Knotens automatisch ein Failover auf ein Replikat durchgeführt ElastiCache wird.

Wenn Sie Memcached ausführen und die Daten über mehrere Knoten partitionieren, sinkt die Größe des Datenverlustes beim Ausfall eines Knotens mit steigender Anzahl verwendeter Knoten.

Minimieren der Auswirkungen von Ausfällen einer Availability Zone

Knoten sollten auf möglichst viele Availability Zones verteilt werden, um die Auswirkungen von Ausfällen einer Availability Zone gering zu halten. Durch die gleichmäßige Verteilung Ihrer Knoten

AZs werden die Auswirkungen im unwahrscheinlichen Fall eines AZ-Ausfalls minimiert. Dies erfolgt automatisch für Serverless-Caches.

Weitere Vorsichtsmaßnahmen

Wenn Sie Valkey oder Redis OSS verwenden, empfehlen wir Ihnen zusätzlich zu den oben genannten, regelmäßige Backups Ihres Clusters zu planen. Bei Backups (Snapshots) wird eine RDB-Datei erstellt, die Sie im Fall eines Ausfalls oder von Datenbeschädigungen zur Wiederherstellung Ihres Caches verwenden können. Weitere Informationen finden Sie unter [Snapshot und Wiederherstellung](#).

Infrastruktursicherheit in AWS ElastiCache

Als verwalteter Dienst AWS ElastiCache wird er durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im [AWS Architecture Center](#) im Abschnitt Sicherheit und Compliance beschrieben sind.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff ElastiCache über das Netzwerk. Clients müssen Transport Layer Security (TLS) 1.2 oder höher unterstützen. Wir empfehlen TLS 1.3 oder höher. Clients müssen außerdem Verschlüsselungssammlungen mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen wie Java 7 und höher 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.

Service-Updates in ElastiCache

ElastiCache überwacht automatisch Ihre Flotte von Caches, Clustern und Knoten, um Service-Updates anzuwenden, sobald sie verfügbar sind. Service-Updates für Serverless-Caches werden automatisch und transparent angewendet. Für knotenbasierte Cluster richten Sie ein vordefiniertes Wartungsfenster ein, damit diese Updates angewendet ElastiCache werden können. In einigen Fällen könnte dieser Ansatz jedoch zu starr sein und möglicherweise Ihre Geschäftsabläufe einschränken.

Mit Service-Updates steuern Sie, wann und welche Updates auf Ihre knotenbasierten Cluster angewendet werden. Sie können den Fortschritt dieser Updates für Ihren ausgewählten ElastiCache Cluster auch in Echtzeit überwachen.

Themen

- [Verwaltung von Dienstupdates für knotenbasierte Cluster](#)

Verwaltung von Dienstupdates für knotenbasierte Cluster

ElastiCache Dienstupdates für knotenbasierte Cluster werden regelmäßig veröffentlicht. Wenn Sie über einen oder mehrere qualifizierte knotenbasierte Cluster für diese Service-Updates verfügen, erhalten Sie Benachrichtigungen per E-Mail, SNS, über das Personal Health Dashboard (PHD) und über CloudWatch Amazon-Events, wenn die Updates veröffentlicht werden. Die Updates werden auch auf der Seite Service Updates in der Konsole angezeigt. ElastiCache Mithilfe dieses Dashboards können Sie alle Service-Updates und deren Status für Ihre ElastiCache Flotte einsehen. Service-Updates für Serverless-Caches werden transparent angewendet und können nicht über Service-Updates verwaltet werden.

Sie legen fest, wann ein Update angewendet wird, bevor ein automatisches Update gestartet wird. Wir empfehlen dringend, alle Updates des Typs Sicherheitsupdate so schnell wie möglich zu installieren, um sicherzustellen, dass Ihre ElastiCache Cluster immer über aktuelle up-to-date Sicherheitspatches verfügen.

In den folgenden Abschnitten werden diese Optionen ausführlich erörtert.

Anwenden der Service-Updates

Sie können die Service-Updates auf Ihre Flotte anwenden, sobald diese Updates den Status available (Verfügbar) haben. Service-Updates sind kumulativ. Das bedeutet, dass bisher nicht angewendete Updates im neuesten Update enthalten sind.

Wenn für ein Service-Update die automatische Aktualisierung aktiviert ist, können Sie festlegen, ob Sie jede Aktion notieren möchten, sobald sie verfügbar ist. ElastiCache plant, dass das Update in einem der nächsten Wartungsfenster Ihrer Cluster nach dem Startdatum der automatischen Aktualisierung installiert wird. Sie erhalten entsprechende Benachrichtigungen für jede Stufe des Updates.

Note

Sie können nur die Service-Updates anwenden, die über den Status verfügbar oder geplant verfügen.

Weitere Informationen zur Überprüfung und Installation von dienstspezifischen Updates auf die entsprechenden ElastiCache Cluster finden Sie unter. [Anwenden der Service-Updates mithilfe der Konsole](#)

Wenn ein neues Service-Update für einen oder mehrere Ihrer ElastiCache Cluster verfügbar ist, können Sie die ElastiCache Konsole, die API oder verwenden, AWS CLI um das Update anzuwenden. Die folgenden Abschnitte beschreiben die Optionen, die zum Anwenden von Updates genutzt werden können.

Anwenden der Service-Updates mithilfe der Konsole

Um die Liste der Service-Updates und weitere Informationen anzuzeigen, gehen Sie zur Seite Service-Updates in der Konsole.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie im Navigationsbereich auf Service-Updates.
3. Unter Service updates (Service-Updates) können Sie Folgendes anzeigen:
 - Name des Service-Updates: Der eindeutige Name des Service-Updates
 - Aktualisierungs-Typ: Typ des Service-Updates, also einer der folgenden Werte: security-update oder engine-update
 - Update Severity (Update-Dringlichkeit): gibt die Priorität an, die für die Anwendung des Updates gilt:
 - kritisch: Wir empfehlen Ihnen, dieses Update sofort (innerhalb von 14 Tagen oder weniger) anzuwenden.
 - Wichtig: Wir empfehlen Ihnen, dieses Update zu installieren, sobald Ihr Geschäftsfluss dies zulässt (innerhalb von 30 Tagen oder weniger).
 - medium: Wir empfehlen Ihnen, dieses Update so schnell wie möglich (innerhalb von 60 Tagen oder weniger) anzuwenden.
 - niedrig: Wir empfehlen Ihnen, dieses Update so schnell wie möglich (innerhalb von 90 Tagen oder weniger) anzuwenden.
 - Engine-Version: Wenn der Aktualisierungstyp Engine-Update ist, die Engine-Version, die aktualisiert wird.
 - Veröffentlichungsdatum: Datum der Veröffentlichung des Updates, ab dem es zur Anwendung auf die Cluster verfügbar ist.

- Recommended Apply By Date (Empfohlene Anwendung nach Datum): ElastiCache-Empfehlungsdatum, bis zu dem die Updates angewendet werden sollten.
 - Status: Status des Updates, einer der folgenden Werte:
 - Verfügbar: Das Update ist für die relevanten Cluster verfügbar.
 - Abgeschlossen: Das Update wurde erfolgreich angewendet.
 - cancelled (Storniert): Das Update wurde storniert und ist nicht mehr erforderlich.
 - expired (Abgelaufen): Das Update kann nicht mehr angewendet werden.
4. Wählen Sie ein einzelnes Update (nicht die Schaltfläche links davon), um Details zum Service-Update anzuzeigen.

Im Abschnitt Cluster update status (Cluster-Aktualisierungsstatus) können Sie eine Liste von Clustern anzeigen, auf denen das Service-Update nicht oder erst kürzlich angewendet wurde. Für jeden Cluster können Sie Folgendes anzeigen:

- Cluster-Name – Der Name des Clusters
- Nodes Updated (Aktualisierte Knoten): Verhältnis der individuellen Knoten in einem bestimmten Cluster, die aktualisiert wurden bzw. weiterhin für das betreffende Service-Update verfügbar sind.
- Aktualisierungs-Typ: Typ des Service-Updates, also einer der folgenden Werte: security-update oder engine-update
- Status: Status des Service-Updates auf dem Cluster, also einer der folgenden Werte:
 - verfügbar: Das Update ist für die relevanten Cluster verfügbar.
 - in Bearbeitung: Das Update wird gerade auf diesen Cluster angewendet.
 - scheduled (geplant): Das Aktualisierungsdatum wurde geplant.
 - complete (abgeschlossen): Das Update wurde erfolgreich angewendet. Cluster mit dem Status „Abgeschlossen“ werden nach Abschluss 7 Tage lang angezeigt.

Wenn Sie einen oder alle Cluster mit dem Status verfügbar oder geplant auswählen und dann auf Apply now (Jetzt anwenden) klicken, wird das Update auf diesen Clustern angewendet.

Anwenden der Service-Updates mit dem AWS CLI

Nachdem Sie benachrichtigt wurden, dass Service-Updates verfügbar sind, können Sie sie mit der AWS CLI inspizieren und anwenden:

- Führen Sie den folgenden Befehl aus, um eine Beschreibung der verfügbaren Serviceaktualisierungen abzurufen:

```
aws elasticache describe-service-updates --service-update-status available
```

Weitere Informationen finden Sie unter [describe-service-updates](#).

- Führen Sie den folgenden Befehl aus, um ein Service-Update auf eine Liste von Clustern anzuwenden:

```
aws elasticache batch-apply-update-action --service-update ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1 cluster2
```

Weitere Informationen finden Sie unter [batch-apply-update-action](#).

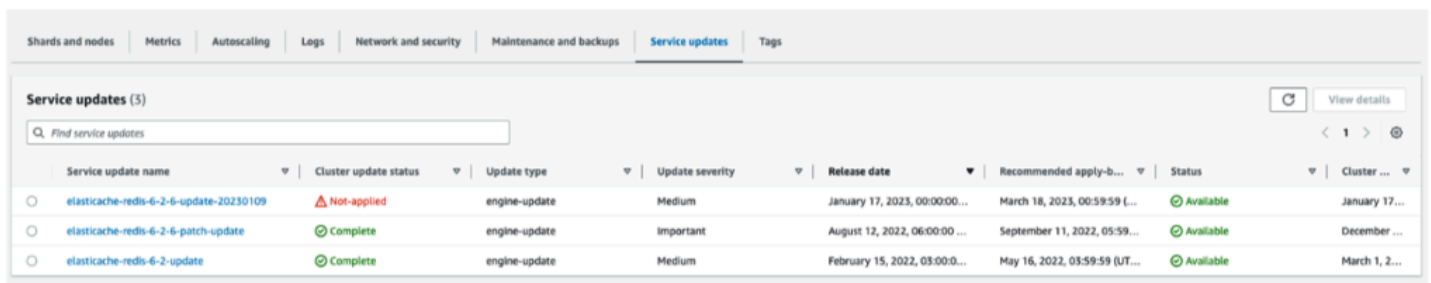
Mithilfe der Konsole wird überprüft, ob das neueste Service-Update angewendet wurde

AWS

Sie können überprüfen, ob auf Ihren ElastiCache Redis OSS-Clustern das neueste Service-Update ausgeführt wird, indem Sie die folgenden Schritte ausführen:

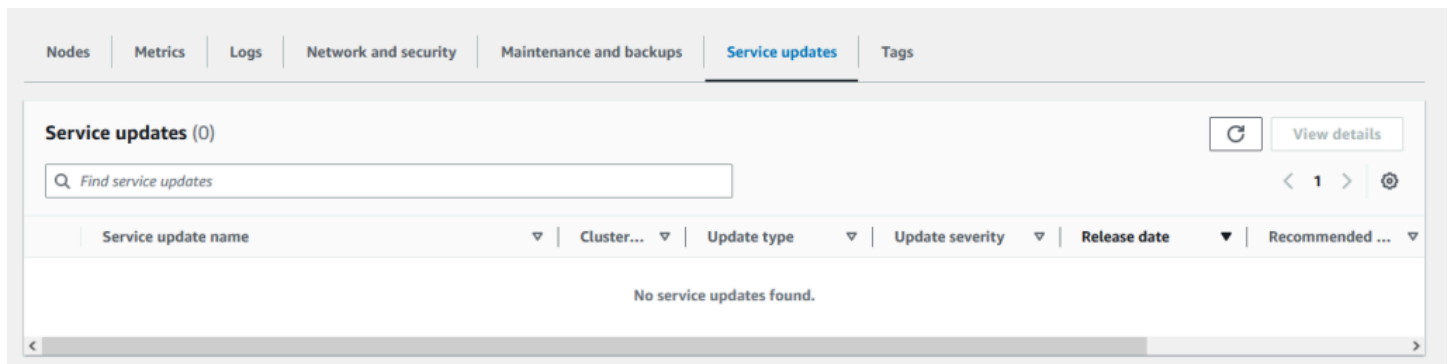
- Wählen Sie auf der Seite Redis OSS-Cluster einen geeigneten Cluster aus
- Wählen Sie im Navigationsbereich Service Updates aus, um die entsprechenden Service-Updates für diesen Cluster, falls vorhanden, zu sehen.

Wenn in der Konsole eine Liste mit Service-Updates angezeigt wird, können Sie das Service-Update auswählen und auf Jetzt anwenden klicken.



Service update name	Cluster update status	Update type	Update severity	Release date	Recommended apply-b...	Status	Cluster ...
elasticache-redis-6-2-6-update-20230109	Not-applied	engine-update	Medium	January 17, 2023, 00:00:00...	March 16, 2023, 00:59:59 (...)	Available	January 17...
elasticache-redis-6-2-6-patch-update	Complete	engine-update	Important	August 12, 2022, 06:00:00 ...	September 11, 2022, 05:59...	Available	December ...
elasticache-redis-6-2-update	Complete	engine-update	Medium	February 15, 2022, 03:00:0...	May 16, 2022, 05:59:59 (UT...	Available	March 1, 2...

Wenn auf der Konsole „Keine Dienstupdates gefunden“ angezeigt wird, bedeutet dies, dass auf dem ElastiCache für Redis OSS-Cluster das neueste Service-Update bereits installiert wurde.



Stoppen der Service-Updates

Sie können Updates für Cluster bei Bedarf beenden. Sie können Updates beispielsweise bei einem unerwarteten Anforderungsanstieg bei den Clustern, die aktualisiert werden, beenden. Oder Sie können Aktualisierungen beenden, wenn sie zu lange dauern und Ihren Geschäftsfluss zu Spitzenzeiten unterbrechen.

Die Operation [Stopping](#) unterbricht sofort alle Updates für die noch zu aktualisierenden Cluster und Knoten. Die Aktualisierung von Knoten mit dem Status in progress (Wird ausgeführt) wird jedoch abgeschlossen. Die Updates für andere Knoten im betreffenden Cluster, die den Status update available (Update verfügbar) aufweisen, werden dagegen nicht angewendet. Die betreffenden Knoten erhalten den Status Stopping (Wird gestoppt).

Sobald der Stopping (Wird gestoppt)-Workflow abgeschlossen ist, erhalten die Knoten mit dem Status Stopping (Wird gestoppt) den Status Stopped (Gestoppt). In Abhängigkeit vom Workflow des Updates werden in einigen Clustern keine Knoten aktualisiert. Andere Cluster enthalten dagegen bereits aktualisierte Knoten sowie Knoten, die weiterhin den Status update available (Update verfügbar) aufweisen.

Sie können den Update-Prozess später und unter Berücksichtigung der betrieblichen Anforderungen abschließen. Wählen Sie in diesem Fall die Cluster, deren Aktualisierung abgeschlossen werden soll, und dann Apply Now (Jetzt anwenden) aus. Weitere Informationen finden Sie unter [Anwenden der Service-Updates](#).

Verwenden der Konsole

Sie können ein Service-Update über die ElastiCache Konsole unterbrechen. Das folgende Beispiel veranschaulicht dies:

- Nachdem ein Service-Update auf einem ausgewählten Cluster abgeschlossen wurde, zeigt die ElastiCache Konsole oben im Dashboard die Registerkarte „Update anzeigen/beenden“ an. ElastiCache
- Wählen Sie zum Unterbrechen des Updates Stop Update (Update stoppen) aus.
- Wählen Sie nach dem Anhalten des Updates den Cluster aus und ermitteln Sie dessen Status. Er wird mit dem Status Stopping (Wird gestoppt) und später mit dem Status Stopped (Gestoppt) angezeigt.

Mit dem AWS CLI

Sie können ein Service-Update mit der AWS CLI unterbrechen. Das folgende Codebeispiel veranschaulicht, wie dazu vorgegangen wird.

Gehen Sie für eine Replikationsgruppe wie folgt vor:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --replication-group-ids my-replication-group-1 my-replication-group-2
```

Gehen Sie für einen Cluster wie folgt vor:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --cache-cluster-ids my-cache-cluster-1 my-cache-cluster-2
```

Weitere Informationen finden Sie unter [BatchStopUpdateAction](#).

Common Vulnerabilities and Exposures (CVE): Sicherheitslücken, die behoben wurden in ElastiCache

Common Vulnerabilities and Exposures (CVE) ist eine Liste von Einträgen für öffentlich bekannte Cybersicherheitsschwachstellen. Jeder Eintrag ist ein Link, der eine Identifikationsnummer, eine Beschreibung und mindestens eine öffentliche Referenz enthält. Auf dieser Seite finden Sie eine Liste von Sicherheitslücken, die unter behoben wurden ElastiCache.

Wir empfehlen, immer auf die neuesten Versionen von ElastiCache Valkey, Redis OSS oder ElastiCache Memcached zu aktualisieren, um vor bekannten Sicherheitslücken geschützt zu sein. Wenn Sie einen ElastiCache serverlosen Cache betreiben, werden CVE-Fixes automatisch auf Ihren Cache angewendet. Beim Betrieb knotenbasierter Cluster mit Valkey oder Redis OSS wird

die PATCH-Komponente verfügbar gemacht. ElastiCache Wenn Sie beispielsweise Version 6.2.6 ElastiCache für Redis OSS verwenden, ist die Hauptversion 6, die Nebenversion ist 2 und die Patch-Version ist 6. PATCH-Versionen sind für abwärtskompatible Bugfixes, Sicherheitskorrekturen und nicht funktionale Änderungen vorgesehen.

Anhand der folgenden Tabelle können Sie überprüfen, ob eine bestimmte Version von ElastiCache Valkey und Redis OSS eine Lösung für eine bestimmte Sicherheitslücke enthält. Wenn auf Ihrem ElastiCache Valkey- oder Redis OSS-Cluster eine Version ohne den Sicherheitspatch ausgeführt wird, schlagen Sie in der folgenden Tabelle nach und ergreifen Sie Maßnahmen. Sie können entweder ein Upgrade auf eine neuere ElastiCache Valkey- oder Redis OSS-Version durchführen, die den Fix enthält, oder wenn Sie eine Version verwenden, die den Fix enthält, stellen Sie anhand von sicher, dass Sie das neueste Service-Update installiert haben, indem Sie auf [Verwaltung von Dienstupdates für knotenbasierte Cluster](#) Weitere Informationen zu den unterstützten ElastiCache Engine-Versionen und zur Durchführung eines Upgrades finden Sie unter [Motorversionen und Aufrüstung in ElastiCache](#)

Note

- Wenn ein CVE in einer ElastiCache Version adressiert wird, bedeutet das, dass es auch in den neueren Versionen adressiert wird. Wenn also beispielsweise eine Sicherheitslücke in ElastiCache Redis OSS Version 6.0.5 behoben wird, gilt dies auch für die Versionen 6.2.6, 7.0.7 und 7.1.
- Ein Sternchen (*) in der folgenden Tabelle gibt an, dass Sie das neueste Service-Update für den for Redis OSS-Cluster installiert haben müssen, auf dem die ElastiCache für Redis OSS angegebene Version ausgeführt wird, um die ElastiCache Sicherheitslücke zu beheben. Weitere Informationen darüber, wie Sie überprüfen können, ob Sie das neueste Service-Update für die Redis OSS-Version installiert haben, auf der ElastiCache Ihr Cluster ausgeführt wird, finden Sie unter [Verwaltung von Dienstupdates für knotenbasierte Cluster](#)

ElastiCache Version	CVEs Adressiert
Valkey 8.1 und alle früheren Versionen von Valkey	CVE-2025-49844 * , CVE-2025-46817 * , CVE-2025-46818 * , CVE-2025-46819 *
Redis OSS 7.1 und alle früheren Versionen von Redis OSS	

ElastiCache Version	CVEs Adressiert
Valkey 7.2 und 7.3	CVE-2025-21607 * , CVE-2025-21605 * , CVE-2024-31449 * , CVE-2024-31227 * , CVE-2024-31228 *
Tal 7.2.7	CVE-2024-51741
Redis OSS 7.1 und 6.2	CVE-2025-21605 * , CVE-2024-31449 * , CVE-2024-31227 * , CVE-2024-31228 * , CVE-2023-41056
Redis OS 7.0.7	CVE-2023-41056 *
Redis OS 6.2.7	CVE-2024-46981
Redis OS 6.2.6	CVE-2023-24834 * , CVE-2022-35977 * , CVE-2023-36021 * , CVE-2023-22458 , CVE-2023-25155 , CVE-2023-28856 CVE-2023-45145 : Beachten Sie, dass dieser CVE in Redis OSS 6.2 und 7.0 behoben wurde, aber nicht in Redis OSS 7.1.
Redis OSS 6.0.5	CVE-2022-24735 * , CVE-2022-24736 *

Protokollierung und Überwachung in Amazon ElastiCache

Um Ihren Cache zu verwalten, ist es wichtig, dass Sie wissen, wie Ihre Caches funktionieren. ElastiCache generiert Metriken, die in Amazon CloudWatch Logs veröffentlicht werden, um Ihre Cache-Leistung zu überwachen. ElastiCache generiert außerdem Ereignisse, wenn signifikante Änderungen an Ihren Cache-Ressourcen vorgenommen werden (z. B. wenn ein neuer Cache erstellt oder ein Cache gelöscht wird).

Themen

- [Metriken und Ereignisse für serverlose OSS-Caches von Valkey und Redis](#)
- [Metriken und Ereignisse für knotenbasierte Valkey- und Redis OSS-Cluster](#)
- [Metriken und Ereignisse für Memcached-Caches und -Cluster](#)
- [Protokollierung Amazon ElastiCache Amazon-API-Aufrufen mit AWS CloudTrail](#)
- [Amazon SNS SNS-Überwachung von Ereignissen ElastiCache](#)
- [Protokollzustellung](#)
- [Überwachung der Nutzung mit CloudWatch Metrics](#)
- [Protokollierung Amazon ElastiCache Amazon-API-Aufrufen mit AWS CloudTrail](#)

Metriken und Ereignisse für serverlose OSS-Caches von Valkey und Redis

ElastiCache bietet eine Vielzahl von Metriken und Ereignissen für die Überwachung bei der Arbeit mit serverlosen Caches. Dazu gehören CloudWatch Metriken, Metriken auf Befehlsebene und Ereignisprotokolle, die über Amazon EventBridge aufgenommen werden können.

Themen

- [Metriken für Serverless-Caches](#)
- [Serverless-Cache-Ereignisse](#)

Metriken für Serverless-Caches

Der AWS/ElastiCache Namespace umfasst die folgenden CloudWatch Metriken für Ihre serverlosen Valkey- oder Redis OSS-Caches.

Metrikcodes für Valkey oder Redis OSS

Metrik	Description	Einheit
BytesUsedForCache	Die Gesamtzahl der von den in Ihrem Cache gespeicherten Daten verwendeten Bytes.	Bytes
ElastiCacheProcessingUnits	Die Gesamtzahl der ElastiCacheProcessingUnits (ECPUs), die von den in Ihrem Cache ausgeführten Anfragen verbraucht wurden	Anzahl
SuccessfulReadRequestLatency	Latenz erfolgreicher Leseanfragen.	Mikrosekunden
SuccessfulWriteRequestLatency	Latenz erfolgreicher Schreibanforderungen	Mikrosekunden
TotalCmdsCount	Gesamtzahl aller Befehle, die in Ihrem Cache ausgeführt wurden	Anzahl
CacheHitRate	Gibt die Trefferrate Ihres Caches an. Dies wird mit <code>cache_hits</code> - und <code>cache_misses</code> -Statistiken auf folgende Weise berechnet : $\text{cache\_hits} / (\text{cache\_hits} + \text{cache\_misses})$.	Prozent

Metrik	Description	Einheit
CacheHits	Die Anzahl der erfolgreichen schreibgeschützten Schlüssel suchereignisse im Cache.	Anzahl
CacheMisses	Die Anzahl der erfolglosen Schlüsselabfragen mit Schreibschutz im Cache.	Anzahl
CurrConnections	Die Anzahl der Client-Verbindungen mit Ihrem Cache.	Anzahl
ThrottledCmds	Die Anzahl der Anfragen, um die gedrosselt wurden, ElastiCache weil die Arbeitslast schneller skaliert wurde, als sie skaliert werden kann. ElastiCache	Anzahl
NewConnections	Gesamtanzahl der Verbindungen, die in diesem Zeitraum vom Server akzeptiert worden sind.	Anzahl
CurrItems	Anzahl der Elemente im Cache.	Anzahl
CurrVolatileItems	Anzahl der Elemente im Cache mit TTL.	Anzahl
NetworkBytesIn	Gesamtzahl der in den Cache übertragenen Bytes	Bytes
NetworkBytesOut	Gesamtzahl der aus dem Cache übertragenen Bytes	Bytes
Evictions	Die Anzahl der Schlüssel, die vom Cache bereinigt wurden	Anzahl

Metrik	Description	Einheit
IamAuthenticationExpirations	Die Gesamtzahl der abgelaufenen IAM-authentifizierten Valkey- oder Redis-OSS -Verbindungen. Weitere Informationen über Authentifizieren mit IAM finden Sie im Benutzerhandbuch.	Anzahl
IamAuthenticationThrottling	Die Gesamtzahl der gedrosselten IAM-authentifizierten Valkey- oder Redis OSS AUTH- oder HELLO-Anfragen. Weitere Informationen über Authentifizieren mit IAM finden Sie im Benutzerhandbuch.	Anzahl
KeyAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, auf Schlüssel zuzugreifen, für die sie keine Zugriffsberechtigung haben. Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl
AuthenticationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche, sich mit dem AUTH-Befehl bei Valkey oder Redis OSS zu authentifizieren. Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl

Metrik	Description	Einheit
CommandAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, Befehle auszuführen, für deren Aufruf sie keine Berechtigung haben. Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl

Metriken auf Befehlsebene

ElastiCache gibt auch die folgenden Metriken auf Befehlsebene aus. Gibt für jeden Befehlstyp ElastiCache die Gesamtzahl der Befehle und die Anzahl der von diesem Befehlstyp ECPUs verwendeten Befehle aus.

Metrik	Description	Einheit
EvalBasedCmds	Die Anzahl der get-Befehle, die der Cache erhalten hat.	Anzahl
EvalBasedCmdsECPUs	ECPUs von Eval-basierten Befehlen verbraucht.	Anzahl
GeoSpatialBasedCmds	Die Gesamtzahl der Befehle für raumbezogene Befehle. Dies wird aus der Valkey- oder Redis-OSS-Commands tats-Statistik abgeleitet. Die Ableitung erfolgt durch Summieren aller Befehle des Geo-Typs: geoadd, geodist, geohash, geopos, georadius und georadiusbymember.	Anzahl

Metrik	Description	Einheit
GeoSpatialBasedCmdsECPUs	ECPUs wird von Befehlen verwendet, die auf Geodaten basieren.	Anzahl
GetTypeCmds	Die Gesamtzahl der schreibgeschützten Befehle. Dies wird aus der Commandstats-Statistik Valkey oder Redis OSS abgeleitet, indem alle Befehle mit Schreibschutz (get, hget, scard, lrange usw.) summiert werden.	Anzahl
GetTypeCmdsECPUs	ECPUs wird von Lesebefehlen verbraucht.	Anzahl
HashBasedCmds	Gesamtanzahl der Hash-basierten Befehle. Dies wird aus der OSS-Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Hashes einwirken (hget, hkeys, hvals, hdel usw.).	Anzahl
HashBasedCmdsECPUs	ECPUs wird von Hash-basierten Befehlen verbraucht.	Anzahl

Metrik	Description	Einheit
HyperLogLogBasedCmds	Die Gesamtzahl der HyperLogLog basierten Befehle. Dies wird aus der Commandstats-Statistik Valkey oder Redis OSS abgeleitet, indem alle Befehle vom Typ pf (pfadd, pfcount, pfmerge usw.) summiert werden.	Anzahl
HyperLogLogBasedCmdsECPUs	ECPUs HyperLogLog wird von basierten Befehlen verbraucht.	Anzahl
JsonBasedCmds	Die Gesamtzahl der JSON-Befehle, einschließlich Lese- und Schreibbefehlen. Dies wird aus der OSS-Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle JSON-Befehle summiert werden, die auf JSON-Schlüssel reagieren.	Anzahl
JsonBasedCmdsECPUs	ECPUs wird von allen JSON-Befehlen, einschließlich Lese- und Schreibbefehlen, verbraucht.	Anzahl

Metrik	Description	Einheit
JsonBasedGetCmds	Gesamtanzahl der JASON-Schreibschutzbefehle. Dies wird aus der OSS-Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle JSON-Lesebefehle summiert werden, die auf JSON-Schlüssel einwirken.	Anzahl
JsonBasedGetCmdsECPUs	ECPUs wird von schreibgeschützten JSON-Befehlen verbraucht.	Anzahl
JsonBasedSetCmds	Gesamtanzahl der JASON-Schreibbefehle. Dies wird aus der OSS-Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle JSON-Schreibbefehle summiert werden, die auf JSON-Schlüssel einwirken.	Anzahl
JsonBasedSetCmdsECPUs	ECPUs wird von JSON-Schreibbefehlen verbraucht.	Anzahl
KeyBasedCmds	Gesamtanzahl der schlüsselbasierten Befehle. Dies wird aus der OSS-Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die sich auf einen oder mehrere Schlüssel in mehreren Datenstrukturen auswirken (del, expire, rename usw.).	Anzahl

Metrik	Description	Einheit
KeyBasedCmdsECPUs	ECPUs wird von schlüsselbasierten Befehlen verbraucht.	Anzahl
ListBasedCmds	Gesamtanzahl der listenbasierten Befehle. Dies wird aus der OSS-Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf eine oder mehrere Listen (lindex, lrange, lpush, ltrim usw.) einwirken.	Anzahl
ListBasedCmdsECPUs	ECPUs wird von listenbasierten Befehlen verbraucht.	Anzahl
NonKeyTypeCmds	Gesamtanzahl der nicht schlüsselbasierten Befehle. Dies wird aus der Commandstats-Statistik Valkey oder Redis OSS abgeleitet, indem alle Befehle summiert werden, die nicht auf einen Schlüssel einwirken, z. B. acl, dbsize oder info.	Anzahl
NonKeyTypeCmdsECPUs	ECPUs non-key-based von Befehlen verbraucht.	Anzahl

Metrik	Description	Einheit
PubSubBasedCmds	Die Gesamtzahl der Befehle für die pub/sub Funktionalität. Dies wird aus den OSS-Commandstatsstatistics von Valkey oder Redis abgeleitet, indem alle für die pub/sub Funktionalität verwendeten Befehle zusammengefasst werden: psubscribe, publish, pubsub, punsubscribe, ssubscribe, sunsubscribe, spublish, subscribe und unsubscribe.	Anzahl
PubSubBasedCmdsECPUs	ECPUs wird von Pub/Sub-basierten Befehlen verwendet.	Anzahl
SetBasedCmds	Gesamtanzahl der Set-basierten Befehle. Dies wird aus der OSS-Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Sätze einwirken (scard, sdiff, sadd, sunion usw.).	Anzahl
SetBasedCmdsECPUs	ECPUs wird von mengenbasierten Befehlen verbraucht.	Anzahl

Metrik	Description	Einheit
SetTypeCmds	Gesamtanzahl der Schreibbe fehle. Dies wird aus der OSS-Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle mutativen Befehlsty pen summiert werden, die mit Daten arbeiten (set, hset, sadd, lpop usw.).	Anzahl
SetTypeCmdsECPUs	ECPUs wird von Schreibbe fehlen verbraucht.	Anzahl
SortedSetBasedCmds	Gesamtanzahl der Sorted Set-basierten Befehle. Dies wird aus der OSS-Comma ndstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf eine oder mehrere sortierte Mengen wirken (zcount, zrange, zrank, zadd usw.).	Anzahl
SortedSetBasedCmdsECPUs	ECPUs wird von sortierten Befehlen verbraucht.	Anzahl
StringBasedCmds	Gesamtanzahl der Zeichenfo lge-basierten Befehle. Dies wird aus der Commandstats-Statistik Valkey oder Redis OSS abgeleitet, indem alle Befehle summiert werden, die auf eine oder mehrere Zeichenketten einwirken (strlen, setex, setrange usw.).	Anzahl

Metrik	Description	Einheit
StringBasedCmdsECPUs	ECPUs wird von auf Zeichenketten basierenden Befehlen verbraucht.	Anzahl
StreamBasedCmds	Die Gesamtanzahl Stream-basierter Befehle. Dies wird aus der Commandstats-Statistik Valkey oder Redis OSS abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Stream-Datentypen (xrange, xlen, xadd, xdel usw.) einwirken.	Anzahl
StreamBasedCmdsECPUs	ECPUs wird von stream-basierten Befehlen verbraucht.	Anzahl

Serverless-Cache-Ereignisse

ElastiCache protokolliert Ereignisse, die sich auf Ihren serverlosen Cache beziehen. Zu diesen Informationen gehören Datum und Uhrzeit des Ereignisses, Quellenname und Quelltyp des Ereignisses sowie eine Beschreibung des Ereignisses. Mit der ElastiCache Konsole, dem Befehl AWS CLI `describe-events` oder der ElastiCache API-Aktion können Sie Ereignisse ganz einfach aus dem Protokoll abrufen. `DescribeEvents`

Sie können wählen, ob Sie ElastiCache Ereignisse mithilfe von Amazon EventBridge überwachen, aufnehmen, transformieren und darauf reagieren möchten. Weitere Informationen finden Sie im [Amazon-Leitfaden „EventBridge Erste Schritte“](#).

ElastiCache Ereignisse anzeigen (Konsole)

So zeigen Sie Ereignisse mit der ElastiCache Konsole an:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>

2. Um eine Liste aller verfügbaren Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Events (Ereignisse).
3. Im Bildschirm Ereignisse repräsentiert jede Zeile in der Liste ein Ereignis und zeigt die Ereignisquelle, den Ereignistyp, die Uhrzeit (GMT) und eine Beschreibung des Ereignisses an. Mit der Option Filter können Sie angeben, ob alle Ereignisse oder nur Ereignisse eines bestimmten Typs in der Ereignisliste enthalten sein sollen.

ElastiCache Ereignisse anzeigen (AWS CLI)

Um mit dem eine Liste von ElastiCache Ereignissen zu generieren AWS CLI, verwenden Sie den Befehl `describe-events`. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Mit dem folgenden Code werden bis zu 40 Serverless-Cache-Ereignisse aufgelistet.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

Mit dem folgenden Code werden alle Ereignisse der letzten 24 Stunden (1 440 Minuten) für Serverless-Caches aufgelistet.

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

Serverless-Ereignisse

In diesem Abschnitt werden die verschiedenen Arten von Ereignissen dokumentiert, die Sie möglicherweise für Ihre Serverless-Caches erhalten.

Ereignisse bei der Erstellung eines Serverless-Caches

detail-type	Description	Einheit	Quelle	Fehlermeldung
Cache wurde erstellt	Cache arn	Erstellung	Serverless-Cache	Der Cache <cache-name> wurde erstellt und ist einsatzbereit.
Cache wurde erstellt	Cache arn	Erstellung	Serverless-Cache	Der Cache <cache-name>

detail-type	Description	Einheit	Quelle	Fehlermeldung
	Snapshot arn			wird erstellt und die Daten wurden aus dem Snapshot wiederhergestellt. Ihr Cache ist einsatzbereit.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Nicht genügend freie IP-Adressen, um einen VPC-Endpunkt zu erstellen.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. In der Anfrage wurden ungültige Subnetze angegeben.

detail-type	Description	Einheit	Quelle	Fehlermeldung
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Das Kontingentlimit für die Erstellung eines VPC-Endpunkts wurde erreicht.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Sie sind nicht berechtigt, einen VPC-Endpunkt zu erstellen.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Ein Benutzer mit einer inkompatiblen Valkey- oder Redis OSS-Version ist in der Benutzergruppe < > vorhanden. user-group-name

detail-type	Description	Einheit	Quelle	Fehlermeldung
Cache-Erstellung fehlgeschlagen	Cache arn Cache snapshot arn	Ausfall	Serverless- Cache	Der Cache <cache-name> konnte nicht erstellt werden. Die angegebene Benutzergruppe < user-group- name > existiert nicht.

detail-type	Description	Einheit	Quelle	Fehlermeldung
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Die Datenwiederherstellung aus dem Snapshot ist fehlgeschlagen, weil <reason>. Gründe für das Fehlschlagen: • Die Datei konnte nicht aus S3 abgerufen werden. • Das erwartete md5 stimmt nicht mit dem tatsächlichen md5 überein. • Die Version der bereitgestellten RDB-Datei wird nicht unterstützt.

Serverlose Cache-Aktualisierungsereignisse (Valkey oder Redis OSS)

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	SecurityGroups <cache-name> für den Cache aktualisiert.
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	Die Tags für den Cache <cache-name> wurden aktualisiert.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. Ein Benutzer mit einer inkompatiblen Valkey- oder Redis OSS-Version ist in der Benutzergruppe < > user-group-name vorhanden.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	<cache-name> Eine Aktualisierung des Caches ist fehlgeschlagen. SecurityGroups Die

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
				Aktualisierung ist fehlgeschlagen.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. SecurityGroups Die Aktualisierung ist aufgrund unzureichender Berechtigungen fehlgeschlagen.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. SecurityGroups Die Aktualisierung ist fehlgeschlagen, weil SecurityGroups sie ungültig sind.

Ereignisse zum Löschen von Cache ohne Server (Valkey oder Redis OSS)

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache wurde gelöscht	Cache arn	Löschung	Serverless-Cache	Der Cache <cache-name> wurde gelöscht.

Ereignisse zur Begrenzung der Cache-Nutzung ohne Server (Valkey oder Redis OSS)

detail-type	Description	Einheit	Quelle	Fehlermeldung
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	Die Limits für den Cache <cache-name> wurden aktualisiert.
Cache-Limit nähert sich	Cache arn	Benachrichtigung	Serverless-Cache	Der Slot <X> verwendet mehr als <Y> % des Limits pro Slot von 32 GB. Der Slot 10 verwendet beispielsweise mehr als 90 % des Limits pro Slot von 32 GB.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist fehlgeschlagen, da der

detail-type	Description	Einheit	Quelle	Fehlermeldung
				Cache gelöscht wurde.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist aufgrund einer ungültigen Konfiguration fehlgeschlagen.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist fehlgeschlagen, da die aktuell zwischengespeicherten Daten die neuen Limits überschreiten. Löschen Sie einige Daten, bevor Sie die Limits anwenden.

Serverlose Cache-Snapshot-Ereignisse (Valkey oder Redis OSS)

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Snapshot wurde erstellt	Cache arn Snapshot arn	Erstellung	serverless-cache-snapshot	Der Snapshot <snapshot-name> wurde für den Cache <cache-name> erstellt.
Snapshot-Erstellung fehlgeschlagen	Cache arn Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht erstellt werden. Die Erstellung des Snapshots <snapshot-name> ist mit dem kundenverwalteten Schlüssel <key-id> fehlgeschlagen <reason>. Meldungen zur Fehlerursache: • Der kundenverwaltete Schlüssel wurde deaktiviert. • Der kundenverwaltete Schlüssel konnte nicht

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
				gefunden werden. • Das Zeitlimit für die Anfrage wurde überschritten.
Snapshot-Erstellung fehlgeschlagen	Cache arn Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht erstellt werden. Die Erstellung des Snapshots <snapshot-name> ist fehlgeschlagen, weil <reason>. Standardursache: • Es liegt ein interner Fehler vor.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket %s exportiert werden, da ElastiCache er keine Berechtigungen für den Bucket besitzt.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da sich bereits ein Objekt mit demselben Namen im Bucket befindet.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da sich die Konto-ID des Bucket-Eigentümers geändert hat.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da auf den S3-Bucket nicht zugegriffen werden kann.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da auf den Bucket nicht zugegriffen werden kann.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da der Bucket nicht existiert.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte mit dem kundenverwalteten Quell-Snapshot-Schlüssel %s nicht in den Bucket '%s' exportiert werden <reason>.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Snapshot konnte nicht kopiert werden	Snapshot arn-1 Snapshot arn-2	Ausfall	serverless-cache-snapshot	Der Snapshot <snapshot-name> konnte nicht kopiert werden. Der Snapshot '%s' konnte nicht in den Snapshot '%s' mit dem kundenverwalteten Quell-Snapshot-Schlüssel <key-id> kopiert werden <reason-name>.
Der Snapshot konnte nicht kopiert werden	Snapshot arn-1 Snapshot arn-2	Ausfall	serverless-cache-snapshot	Der Snapshot <snapshot-name> konnte nicht kopiert werden. Der Snapshot '%s' konnte nicht in den Snapshot '%s' mit dem kundenverwalteten Ziel-Snapshot-Schlüssel '%s' '%s' kopiert werden.

Metriken und Ereignisse für knotenbasierte Valkey- und Redis OSS-Cluster

ElastiCache bietet eine Vielzahl von Metriken und Ereignissen für die Überwachung knotenbasierter Valkey- und Redis-OSS-Cluster. Dazu gehören Metriken auf Hostebene, Metriken auf Befehlsebene und Ereignisprotokolle, die über den AWS CLI und den Amazon Simple Notification Service (SNS) verfügbar sind.

Themen

- [Metriken für knotenbasierte Cluster](#)
- [Ereignisse für knotenbasierte Cluster \(Valkey und Redis OSS\)](#)

Metriken für knotenbasierte Cluster

Wenn Sie einen knotenbasierten Cluster erstellen, werden Metriken auf jeder Knotenebene ElastiCache ausgegeben, einschließlich Metriken auf Host-Ebene und Cache-Metriken.

Weitere Informationen zu Metriken auf Host-Ebene finden Sie unter [Metriken auf Host-Ebene](#).

Weitere Informationen zu Metriken auf Knotenebene finden Sie unter [Metriken für Valkey und Redis OSS](#).

Ereignisse für knotenbasierte Cluster (Valkey und Redis OSS)

ElastiCache protokolliert Ereignisse, die sich auf Ihre knotenbasierten Cluster beziehen. Wenn Sie mit knotenbasierten Clustern arbeiten, können Sie Ihre Cluster-Ereignisse in der ElastiCache Konsole, mithilfe von oder mithilfe von Amazon Simple Notification Service (SNS) anzeigen. AWS CLI Knotenbasierte Cluster-Ereignisse werden nicht auf Amazon veröffentlicht. EventBridge

Zu den knotenbasierten Cluster-Ereignisinformationen gehören Datum und Uhrzeit des Ereignisses, Quellname und Quelltyp des Ereignisses sowie eine Beschreibung des Ereignisses. Mit der ElastiCache Konsole, dem Befehl AWS CLI describe-events oder der API-Aktion können Sie Ereignisse ganz einfach aus dem Protokoll abrufen. ElastiCache DescribeEvents

ElastiCache Ereignisse anzeigen (Konsole)

Das folgende Verfahren zeigt Ereignisse mithilfe der ElastiCache Konsole an.

Um Ereignisse mit der ElastiCache Konsole anzuzeigen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Um eine Liste aller verfügbaren Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Events (Ereignisse).
3. Im Bildschirm Ereignisse repräsentiert jede Zeile in der Liste ein Ereignis und zeigt die Ereignisquelle, den Ereignistyp, die Uhrzeit (GMT) und eine Beschreibung des Ereignisses an. Mit der Option Filter können Sie angeben, ob alle Ereignisse oder nur Ereignisse eines bestimmten Typs in der Ereignisliste enthalten sein sollen.

ElastiCache Ereignisse anzeigen (AWS CLI)

Um mit dem eine Liste von ElastiCache Ereignissen zu generieren AWS CLI, verwenden Sie den Befehl `describe-events`. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Der folgende Code listet bis zu 40 knotenbasierte Clusterereignisse auf.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

Der folgende Code listet alle Ereignisse für knotenbasierte Caches der letzten 24 Stunden (1440 Minuten) auf.

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

Knotenbasierte Cluster-Ereignisse

Dieser Abschnitt enthält die Liste der Ereignisse, mit denen Sie für Ihre knotenbasierten Cluster rechnen können.

Die folgenden ElastiCache Ereignisse lösen Amazon SNS SNS-Benachrichtigungen aus. Weitere Informationen zu Ereignisdetails finden Sie unter [ElastiCache Ereignisse anzeigen](#).

Ereignisname	Fehlermeldung	Description
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Ein Cache-Knoten wurde dem Cluster hinzugefügt und ist einsatzbereit.

Ereignisname	Fehlermeldung	Description
ElastiCache: AddCacheNodeFailed aufgrund unzureichender freier IP-Adressen	ElastiCache: AddCacheNodeFailed : <i>cluster-name</i>	Ein Cache-Knoten konnte nicht hinzugefügt werden, da nicht genug IP-Adressen verfügbar sind.
ElastiCache: CacheClusterParametersChanged	ElastiCache: CacheClusterParametersChanged : <i>cluster-name</i>	Ein oder mehrere Cluster-Parameter wurden geändert.
ElastiCache: CacheClusterProvisioningComplete	ElastiCache: CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	Die Bereitstellung eines Clusters ist abgeschlossen, und die Cache-Knoten im Cache-Cluster sind einsatzbereit.
ElastiCache: CacheClusterProvisioningFailed aufgrund eines inkompatiblen Netzwerkstatus	ElastiCache: CacheClusterProvisioningFailed : <i>cluster-name</i>	Es wurde versucht, einen neuen Cluster in einer nicht existierenden Virtual Private Cloud (VPC) zu starten.
ElastiCache: CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Die Skalierung für den Cache-Cluster wurde erfolgreich abgeschlossen.
ElastiCache: CacheClusterScalingFailed	ElastiCache: CacheClusterScalingFailed : <i>cluster-name</i>	Die Erweiterung des Cache-Clusters ist fehlgeschlagen.

Ereignisname	Fehlermeldung	Description
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	Eines der folgenden Ereignisse ist aufgetreten: • Die Liste der für den Cluster autorisierten Cache-Sicherheitsgruppen wurde geändert. • Eine oder mehrere neue EC2 Sicherheitsgruppen wurden für eine der Cache-Sicherheitsgruppen, die dem Cluster zugeordnet sind, autorisiert. • Eine oder mehrere EC2 Sicherheitsgruppen wurden aus einer der Cache-Sicherheitsgruppen, die dem Cluster zugeordnet sind, gesperrt.

Ereignisname	Fehlermeldung	Description
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	ElastiCache hat festgestellt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, heruntergestuft oder nicht erreichbar ist, und hat begonnen, den Cache-Knoten zu ersetzen.  Note Der DNS-Eintrag für den ersetzten Cache-Knoten bleibt unverändert. In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Client-Bibliotheken verwenden den Cache-Knoten möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.

Ereignisname	Fehlermeldung	Description
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	ElastiCache hat festgestellt, dass der Host, auf dem ein Cacheknoten ausgeführt wird, heruntergefahren oder nicht erreichbar ist, und hat den Austausch des Cacheknotens abgeschlossen.  Note Der DNS-Eintrag für den ersetzten Cache-Knoten bleibt unverändert. In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Client-Bibliotheken verwenden den Cache-Knoten möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt ElastiCache wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.

Ereignisname	Fehlermeldung	Description
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Mindestens ein Cache-Knoten wurde neu gestartet. Nachricht (Memcached): "Cache node %s shutdown" Dann eine zweite Nachricht: "Cache node %s restarted"
ElastiCache: CertificateRenewalComplete (Nur Valkey oder Redis OSS)	ElastiCache:CertificateRenewalComplete	Das Amazon CA-Zertifikat wurde erfolgreich erneuert.
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	Die Replikationsgruppe wurde erfolgreich erstellt.
ElastiCache>DeleteCacheClusterComplete	ElastiCache>DeleteCacheClusterComplete : <i>cluster-name</i>	Das Löschen eines Clusters und aller zugehörigen Cache-Knoten ist abgeschlossen.
ElastiCache: FailoverComplete (Nur Valkey oder Redis OSS)	ElastiCache:FailoverComplete : <i>mycluster</i>	Failover zu einem Replikationsknoten wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	Die Anzahl der Replikate im Cluster wurde erhöht.

Ereignisname	Fehlermeldung	Description
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	Der Vorgang für das Hinzufügen von Replikaten zu Ihrem Cluster hat begonnen.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Ein Knoten in Ihrem Cluster, der ersetzt werden sollte, soll nicht länger ersetzt werden.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	Für einen zu ersetzenden Knoten in Ihrem Cluster wurde eine spätere Ersetzung im neuen Fenster geplant, das in der Benachrichtigung angegeben wurde. Weitere Informationen zu den möglichen Aktionen erhalten Sie unter Knoten ersetzen (Valkey und Redis OSS).
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	Ein Knoten in Ihrem Cluster soll während des in der Benachrichtigung beschriebenen Fensters ersetzt werden. Weitere Informationen zu den möglichen Aktionen erhalten Sie unter Knoten ersetzen (Valkey und Redis OSS).
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Ein Cache-Knoten wurde aus dem Cluster entfernt.

Ereignisname	Fehlermeldung	Description
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	Die Aufskalierung der Replikationsgruppe wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	Die Aufskalierung der Replikationsgruppe ist fehlgeschlagen.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Für den Knoten ist ein Self-Service-Update verfügbar.
ElastiCache: SnapshotComplete (Nur Valkey oder Redis OSS)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Ein Cache-Snapshot wurde erfolgreich abgeschlossen.
ElastiCache: SnapshotFailed (Nur Valkey oder Redis OSS)	SnapshotFailed : <i>cluster-name</i>	Ein Cache-Snapshot ist fehlgeschlagen. Weitere Informationen zur Ursache finden Sie in den Cache-Ereignissen des Clusters. Wenn Sie den Snapshot beschreiben (siehe DescribeSnapshots), ist dessen Status failed.

Metriken und Ereignisse für Memcached-Caches und -Cluster

In diesem Abschnitt werden die Metriken und Ereignisse beschrieben, die Sie bei der Arbeit mit knotenbasierten und serverlosen Memcached-Caches überwachen können.

Themen

- [Metriken für Serverless für Memcached ElastiCache](#)
- [Ereignisse für Serverless for Memcached ElastiCache](#)

Metriken für Serverless für Memcached ElastiCache

In diesem Abschnitt werden die Metriken und Ereignisse beschrieben, die Sie bei der Arbeit mit ElastiCache Serverless for Memcached überwachen können.

Der AWS/ElastiCache Namespace umfasst die folgenden CloudWatch Metriken für Ihre serverlosen Memcached-Caches.

Metrik	Description	Einheit
BytesUsedForCache	Die Gesamtzahl der von den in Ihrem Cache gespeicherten Daten verwendeten Bytes.	Bytes
ElastiCacheProcessingUnits	Die Gesamtzahl der ElastiCacheProcessingUnits (ECPUs), die von den in Ihrem Cache ausgeführten Anfragen verbraucht wurden	Anzahl
SuccessfulReadRequestLatency	Latenz erfolgreicher Leseanfragen.	Mikrosekunden
SuccessfulWriteRequestLatency	Latenz erfolgreicher Schreibanforderungen	Mikrosekunden
TotalCmdsCount	Gesamtzahl aller Befehle, die in Ihrem Cache ausgeführt wurden	Anzahl

Metrik	Description	Einheit
CurrConnections	Die Anzahl der Client-Verbindungen mit Ihrem Cache.	Anzahl
ThrottledCmds	Die Anzahl der Anfragen, um die gedrosselt wurden, ElastiCache weil die Arbeitslast schneller skaliert wurde, als sie skaliert ElastiCache werden kann.	Anzahl
NewConnections	Gesamtanzahl der Verbindungen, die in diesem Zeitraum vom Server akzeptiert worden sind.	Anzahl
CurrItems	Anzahl der Elemente im Cache.	Anzahl
NetworkBytesIn	Gesamtzahl der in den Cache übertragenen Bytes	Bytes
NetworkBytesOut	Gesamtzahl der aus dem Cache übertragenen Bytes	Bytes
Evictions	Die Anzahl der Schlüssel, die vom Cache bereinigt wurden	Anzahl
Zurückgefordert	Die Anzahl der Schlüssel, die im Cache abgelaufen sind	Anzahl

Metriken auf Befehlsebene

ElastiCache gibt außerdem die folgenden Memcached-Metriken auf Befehlsebene aus

Metrik	Description	Einheit
CmdGet	Die Anzahl der get-Befehle, die der Cache erhalten hat.	Anzahl
CmdSet	Die Anzahl der Set-Befehle, die der Cache empfangen hat.	Anzahl
CmdTouch	Die Anzahl der touch-Befehle, die der Cache erhalten hat.	Anzahl
GetHits	Die Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
GetMisses	Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrHits	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrMisses	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl

Metrik	Description	Einheit
DecrHits	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
DecrMisses	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
DeleteHits	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
DeleteMisses	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
TouchHits	Die Anzahl der Schlüssel, auf die zugegriffen wurde und die mit einer neuen Ablaufzeit versehen wurden.	Anzahl
TouchMisses	Die Anzahl der Schlüssel, auf die zwar zugegriffen wurde, die aber nicht gefunden werden konnten.	Anzahl

Metrik	Description	Einheit
CasHits	Die Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel gefunden wurde und die CAS-Werte übereinstimmen.	Anzahl
CasMisses	Die Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
CasBadval	Die Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der CAS-Wert nicht mit dem gespeicherten CAS-Wert übereinstimmt.	Anzahl
CmdFlush	Anzahl der flush-Befehle, die der Cache erhalten hat.	Anzahl

Ereignisse für Serverless for Memcached ElastiCache

ElastiCache protokolliert Ereignisse, die sich auf Ihren serverlosen Cache beziehen. Zu diesen Informationen gehören Datum und Uhrzeit des Ereignisses, Quellename und Quelltyp des Ereignisses sowie eine Beschreibung des Ereignisses. Mit der ElastiCache Konsole, dem Befehl AWS CLI `describe-events` oder der ElastiCache API-Aktion können Sie Ereignisse ganz einfach aus dem Protokoll abrufen. `DescribeEvents`

Sie können wählen, ob Sie ElastiCache Ereignisse mithilfe von Amazon EventBridge überwachen, aufnehmen, transformieren und darauf reagieren möchten. Weitere Informationen finden Sie im [Amazon-Leitfaden „EventBridge Erste Schritte“](#).

ElastiCache Ereignisse anzeigen (Konsole)

So zeigen Sie Ereignisse mit der ElastiCache Konsole an:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Um eine Liste aller verfügbaren Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Events (Ereignisse).
3. Im Bildschirm Ereignisse repräsentiert jede Zeile in der Liste ein Ereignis und zeigt die Ereignisquelle, den Ereignistyp, die Uhrzeit (GMT) und eine Beschreibung des Ereignisses an. Mit der Option Filter können Sie angeben, ob alle Ereignisse oder nur Ereignisse eines bestimmten Typs in der Ereignisliste enthalten sein sollen.

ElastiCache Ereignisse anzeigen (AWS CLI)

Um mit dem eine Liste von ElastiCache Ereignissen zu generieren AWS CLI, verwenden Sie den Befehl `describe-events`. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Mit dem folgenden Code werden bis zu 40 Serverless-Cache-Ereignisse aufgelistet.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

Mit dem folgenden Code werden alle Ereignisse der letzten 24 Stunden (1 440 Minuten) für Serverless-Caches aufgelistet.

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

Serverless-Ereignisse

In diesem Abschnitt werden die verschiedenen Arten von Ereignissen dokumentiert, die Sie möglicherweise für Ihre Serverless-Caches erhalten.

Ereignisse bei der Erstellung eines Serverless-Caches

detail-type	Description	Einheit	Quelle	Fehlermeldung
Cache wurde erstellt	Cache arn	Erstellung	Serverless-Cache	Der Cache <cache-name> wurde erstellt

detail-type	Description	Einheit	Quelle	Fehlermeldung
				und ist einsatzbe reit.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless- Cache	Der Cache <cache-name> konnte nicht erstellt werden. Nicht genügend freie IP-Adress en, um einen VPC-Endpunkt zu erstellen.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless- Cache	Der Cache <cache-name> konnte nicht erstellt werden. In der Anfrage wurden ungültige Subnetze angegeben.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless- Cache	Der Cache <cache-name> konnte nicht erstellt werden. Das Kontingen tlimit für die Erstellung eines VPC-Endpunkts wurde erreicht.

detail-type	Description	Einheit	Quelle	Fehlermeldung
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Sie sind nicht berechtigt, einen VPC-Endpunkt zu erstellen.

Serverlose Cache-Aktualisierungsereignisse (Memcached)

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	SecurityGroups <cache-name> für den Cache aktualisiert.
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	Die Tags für den Cache <cache-name> wurden aktualisiert.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. SecurityGroups Die Aktualisierung ist fehlgeschlagen.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. Security Groups Die Aktualisierung ist aufgrund unzureichender Berechtigungen fehlgeschlagen.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. Security Groups Die Aktualisierung ist fehlgeschlagen, weil SecurityGroups sie ungültig sind.

Ereignisse zum Löschen von Cache ohne Server (Memcached)

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache wurde gelöscht	Cache arn	Löschung	Serverless-Cache	Der Cache <cache-name> wurde gelöscht.

Ereignisse zur Begrenzung der Cache-Nutzung ohne Server (Memcached)

detail-type	Description	Einheit	Quelle	Fehlermeldung
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	Die Limits für den Cache <cache-name> wurden aktualisiert.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist fehlgeschlagen, da der Cache gelöscht wurde.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist aufgrund einer ungültigen Konfiguration fehlgeschlagen.

Serverlose Cache-Snapshot-Ereignisse (Memcached)

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Snapshot wurde erstellt	Cache arn Snapshot arn	Erstellung	serverless-cache-snapshot	Der Snapshot <snapshot-name> wurde für den Cache <cache-name> erstellt.
Snapshot-Erstellung fehlgeschlagen	Cache arn Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht erstellt werden. Die Erstellung des Snapshots <snapshot-name> ist mit dem kundenverwalteten Schlüssel <key-id> fehlgeschlagen <reason>. Meldungen zur Fehlerursache: • Der kundenverwaltete Schlüssel wurde deaktiviert. • Der kundenverwaltete Schlüssel konnte nicht

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
				gefunden werden. • Das Zeitlimit für die Anfrage wurde überschritten.
Snapshot-Erstellung fehlgeschlagen	Cache arn Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht erstellt werden. Die Erstellung des Snapshots <snapshot-name> ist fehlgeschlagen, weil <reason>. Standardursache: • Es liegt ein interner Fehler vor.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket %s exportiert werden, da ElastiCache er keine Berechtigungen für den Bucket besitzt.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da sich bereits ein Objekt mit demselben Namen im Bucket befindet.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da sich die Konto-ID des Bucket-Eigentümers geändert hat.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da auf den S3-Bucket nicht zugegriffen werden kann.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da auf den Bucket nicht zugegriffen werden kann.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da der Bucket nicht existiert.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte mit dem kundenverwalteten Quell-Snapshot-Schlüssel %s nicht in den Bucket '%s' exportiert werden <reason>.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Snapshot konnte nicht kopiert werden	Snapshot arn-1 Snapshot arn-2	Ausfall	serverless-cache-snapshot	Der Snapshot <snapshot-name> konnte nicht kopiert werden. Der Snapshot '%s' konnte nicht in den Snapshot '%s' mit dem kundenverwalteten Quell-Snapshot-Schlüssel <key-id> kopiert werden <reason-name>.
Der Snapshot konnte nicht kopiert werden	Snapshot arn-1 Snapshot arn-2	Ausfall	serverless-cache-snapshot	Der Snapshot <snapshot-name> konnte nicht kopiert werden. Der Snapshot '%s' konnte nicht in den Snapshot '%s' mit dem kundenverwalteten Ziel-Snapshot-Schlüssel '%s' '%s' kopiert werden.

Protokollierung Amazon ElastiCache Amazon-API-Aufrufen mit AWS CloudTrail

Amazon ElastiCache ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon ausgeführt wurden ElastiCache. CloudTrail erfasst alle API-Aufrufe für Amazon ElastiCache als Ereignisse, einschließlich Aufrufe von der ElastiCache Amazon-Konsole und von Codeaufrufen an die ElastiCache Amazon-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Amazon ElastiCache. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon gestellt wurde ElastiCache, 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](#).

ElastiCache Amazon-Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn in Amazon Aktivitäten auftreten ElastiCache, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie 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 Veranstaltungen für Amazon ElastiCache, 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 standardmäßig für alle Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. 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 hier:

- [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 ElastiCache Amazon-Aktionen werden von der [ElastiCache API-Referenz](#) protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe von `DescribeCacheCluster` und `ModifyCacheCluster` Aktionen Einträge in den CloudTrail Protokolldateien.

`CreateCacheCluster`

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](#).

ElastiCache Amazon-Protokolldateieinträge verstehen

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 `CreateCacheCluster` Aktion demonstriert.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "elasticache-allow"
```

```
    },
    "eventTime": "2014-12-01T22:00:35Z",
    "eventSource": "elasticache.amazonaws.com",
    "eventName": "CreateCacheCluster",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.01",
    "userAgent": "AWS CLI/ElastiCache 1.10 API 2014-12-01",
    "requestParameters": {
      "numCacheNodes": 2,
      "cacheClusterId": "test-memcached",
      "engine": "memcached",
      "aZMode": "cross-az",
      "cacheNodeType": "cache.m1.small",
    },
    "responseElements": {
      "engine": "memcached",
      "clientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "cacheParameterGroup": {
        "cacheParameterGroupName": "default.memcached1.4",
        "cacheNodeIdsToReboot": {
        },
        "parameterApplyStatus": "in-sync"
      },
      "preferredAvailabilityZone": "Multiple",
      "numCacheNodes": 2,
      "cacheNodeType": "cache.m1.small",

      "cacheClusterStatus": "creating",
      "autoMinorVersionUpgrade": true,
      "preferredMaintenanceWindow": "thu:05:00-thu:06:00",
      "cacheClusterId": "test-memcached",
      "engineVersion": "1.4.14",
      "cacheSecurityGroups": [
        {
          "status": "active",
          "cacheSecurityGroupName": "default"
        }
      ],
      "pendingModifiedValues": {
      }
    },
    "requestID": "104f30b3-3548-11e4-b7b8-6d79ffe84edd",
```

```
"eventID": "92762127-7a68-42ce-8787-927d2174cde1"
}
```

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `DescribeCacheCluster` Aktion demonstriert. Beachten Sie, dass bei allen Amazon ElastiCache Describe-Aufrufen (Describe*) der `ResponseElements` Abschnitt entfernt wird und als `null` angezeigt wird.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "elasticache-allow"
  },
  "eventTime": "2014-12-01T22:01:00Z",
  "eventSource": "elasticache.amazonaws.com",
  "eventName": "DescribeCacheClusters",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.01",
  "userAgent": "AWS CLI/ElastiCache 1.10 API 2014-12-01",
  "requestParameters": {
    "showCacheNodeInfo": false,
    "maxRecords": 100
  },
  "responseElements": null,
  "requestID": "1f0b5031-3548-11e4-9376-c1d979ba565a",
  "eventID": "a58572a8-e81b-4100-8e00-1797ed19d172"
}
```

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der eine `ModifyCacheCluster` Aktion aufzeichnet.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId": "123456789012",

```

```
    "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
    "userName":"elasticache-allow"
  },
  "eventTime":"2014-12-01T22:32:21Z",
  "eventSource":"elasticache.amazonaws.com",
  "eventName":"ModifyCacheCluster",
  "awsRegion":"us-west-2",
  "sourceIPAddress":"192.0.2.01",
  "userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
  "requestParameters":{
    "applyImmediately":true,
    "numCacheNodes":3,
    "cacheClusterId":"test-memcached"
  },
  "responseElements":{
    "engine":"memcached",
    "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/home#client-download:",
    "cacheParameterGroup":{
      "cacheParameterGroupName":"default.memcached1.4",
      "cacheNodeIdsToReboot":{
      },
      "parameterApplyStatus":"in-sync"
    },
    "cacheClusterCreateTime":"Dec 1, 2014 10:16:06 PM",
    "preferredAvailabilityZone":"Multiple",
    "numCacheNodes":2,
    "cacheNodeType":"cache.m1.small",
    "cacheClusterStatus":"modifying",
    "autoMinorVersionUpgrade":true,
    "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
    "cacheClusterId":"test-memcached",
    "engineVersion":"1.4.14",
    "cacheSecurityGroups":[
      {
        "status":"active",
        "cacheSecurityGroupName":"default"
      }
    ],
    "configurationEndpoint":{
      "address":"test-memcached.example.cfg.us1prod.cache.amazonaws.com",
      "port":11211
    },
    "pendingModifiedValues":{
```

```
        "numCacheNodes":3
    },
    "requestID":"807f4bc3-354c-11e4-9376-c1d979ba565a",
    "eventID":"e9163565-376f-4223-96e9-9f50528da645"
}
```

Amazon SNS SNS-Überwachung von Ereignissen ElastiCache

Wenn für einen Cluster wichtige Ereignisse eintreten, ElastiCache sendet eine Benachrichtigung an ein bestimmtes Amazon SNS SNS-Thema. Zu den wichtigen Ereignissen zählen beispielsweise das fehlgeschlagene Hinzufügen eines Knotens, das erfolgreiche Hinzufügen eines Knotens und die Änderung einer Sicherheitsgruppe. Durch die Überwachung wichtiger Schlüsselereignisse können Sie den aktuellen Status Ihrer Cluster erfahren und, je nach Ereignis, Korrekturen vornehmen.

Themen

- [Verwaltung von ElastiCache Amazon SNS SNS-Benachrichtigungen](#)
- [ElastiCache Ereignisse anzeigen](#)
- [Ereignisbenachrichtigungen und Amazon SNS](#)

Verwaltung von ElastiCache Amazon SNS SNS-Benachrichtigungen

Mit Amazon Simple Notification Service (Amazon SNS) können Sie konfigurieren ElastiCache , dass Benachrichtigungen für wichtige Cluster-Ereignisse gesendet werden. In diesen Beispielen konfigurieren Sie einen Cluster mit dem Amazon-Ressourcenname (ARN) eines Amazon-SNS-Themas, um Benachrichtigungen zu erhalten.

Note

- In diesem Thema wird davon ausgegangen, dass Sie sich bei Amazon SNS angemeldet und ein Amazon-SNS-Thema eingerichtet und abonniert haben. Informationen dazu finden Sie im [Entwicklerhandbuch zu Amazon Simple Notification Service](#).
- Standardmäßig API `modify-replication-group` betrifft das alle Gruppen in einer Region und nicht nur die aktuell angegebene Gruppe. Wenn Sie eine bestimmte Gruppe in einer Region anders konfigurieren möchten als die anderen Gruppen, können Sie die

--notification-topic-arn Option verwenden, um ein separates Thema für diese Gruppe zu erstellen.

Hinzufügen eines Amazon-SNS-Themas

In den folgenden Abschnitten erfahren Sie, wie Sie mithilfe der AWS Konsole, der oder der ElastiCache API ein Amazon SNS SNS-Thema hinzufügen.AWS CLI

Hinzufügen eines Amazon-SNS-Themas (Konsole)

Das folgende Verfahren zeigt Ihnen, wie Sie ein Amazon-SNS-Thema für einen Cluster hinzufügen. Wenn Sie Valkey oder Redis OSS verwenden, um in Schritt 2 ein Amazon SNS SNS-Thema für eine Replikationsgruppe hinzuzufügen, wählen Sie statt eines Clusters eine Replikationsgruppe aus. Folgen Sie dann den übrigen Schritten.

Note

Diese Vorgehensweise kann auch zum Ändern des Amazon-SNS-Themas verwendet werden.

So fügen Sie ein Amazon-SNS-Thema für einen Cluster hinzu oder ändern es (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie unter Cluster den Cluster aus, für den Sie einen Amazon-SNS-Thema-ARN hinzufügen oder ändern möchten.
3. Wählen Sie Ändern aus.
4. Wählen Sie im Feld Modify Cluster (Cluster ändern) unter Topic for SNS Notification (Thema für SNS-Benachrichtigung) das SNS-Thema aus, das Sie hinzufügen möchten, oder wählen Sie Manual ARN input (Manuelle ARN-Eingabe) aus und geben Sie den ARN des Amazon-SNS-Themas ein.
5. Wählen Sie Ändern aus.

Hinzufügen eines Amazon-SNS-Themas (AWS CLI)

Verwenden Sie den AWS CLI Befehl `modify-cache-cluster`, um ein Amazon SNS SNS-Thema für einen Cluster hinzuzufügen oder zu ändern.

Das folgende Codebeispiel fügt einen Amazon-SNS-Themen-ARN zu `my-cluster` hinzu.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --notification-topic-arn arn:aws:sns:us-west-2:123456789xxx:ElastiCacheNotifications
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

Weitere Informationen finden Sie unter [modify-cache-cluster](#).

Hinzufügen eines Amazon SNS SNS-Themas (ElastiCache API)

Rufen Sie zum Hinzufügen oder Ändern eines Amazon-SNS-Themas für einen Cluster die Aktion `ModifyCacheCluster` mit folgenden Parametern auf:

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

Example

```
https://elasticache.amazon.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=false  
  &CacheClusterId=my-cluster  
  &NotificationTopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications  
  &Version=2014-12-01  
  &SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie unter [ModifyCacheCluster](#).

Aktivieren und Deaktivieren von Amazon-SNS-Benachrichtigungen

Sie können Benachrichtigungen für einen Cluster aktivieren oder deaktivieren. Das folgende Verfahren zeigt, wie Sie Amazon-SNS-Benachrichtigungen deaktivieren.

Aktivieren und Deaktivieren von Amazon-SNS-Benachrichtigungen (Konsole)

Um Amazon SNS SNS-Benachrichtigungen zu deaktivieren, verwenden Sie den AWS-Managementkonsole

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste der Cluster anzuzeigen, die Memcached ausführen, wählen Sie im Navigationsbereich Memcached.

Um eine Liste Ihrer Cluster anzuzeigen, auf denen Valkey oder Redis OSS ausgeführt werden, wählen Sie im Navigationsbereich Valkey oder Redis OSS aus.

3. Aktivieren Sie das Kontrollkästchen links neben dem Cluster, für den die Benachrichtigung geändert werden soll.
4. Wählen Sie Ändern aus.
5. Wählen Sie im Feld Modify Cluster unter Topic for SNS Notification die Option Disable Notifications aus.
6. Wählen Sie Ändern aus.

Aktivieren und Deaktivieren von Amazon-SNS-Benachrichtigungen (AWS CLI)

Verwenden Sie zum Deaktivieren von Amazon-SNS-Benachrichtigungen den Befehl `modify-cache-cluster` mit folgenden Parametern:

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --notification-topic-status inactive
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --notification-topic-status inactive
```

 Note

Wenn der Cluster zu einer Replikationsgruppe gehört, müssen Sie den CLI-Befehl verwenden, `modify-replication-group` um SNS-Benachrichtigungen zu aktivieren oder zu deaktivieren.

Amazon SNS SNS-Benachrichtigungen (ElastiCache API) aktivieren und deaktivieren

Rufen Sie zum Deaktivieren von Amazon-SNS-Benachrichtigungen die `ModifyCacheCluster`-Aktion mit folgenden Parametern auf:

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

Diese Aktion führt zu folgender oder einer ähnlichen Ausgabe:

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&ApplyImmediately=false  
&CacheClusterId=my-cluster  
&NotificationTopicStatus=inactive  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z
```

```
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

ElastiCache Ereignisse anzeigen

ElastiCache protokolliert Ereignisse, die sich auf Ihre Cluster-Instances, Sicherheitsgruppen und Parametergruppen beziehen. Zu diesen Informationen gehören Datum und Uhrzeit des Ereignisses, Quellname und Quelltyp des Ereignisses sowie eine Beschreibung des Ereignisses. Sie können Ereignisse mithilfe der ElastiCache Konsole, des AWS CLIdescribe-events Befehls oder der ElastiCache API-Aktion ganz einfach aus dem Protokoll abrufenDescribeEvents.

Die folgenden Verfahren zeigen Ihnen, wie Sie alle ElastiCache Ereignisse der letzten 24 Stunden (1440 Minuten) anzeigen können.

ElastiCache Ereignisse anzeigen (Konsole)

Das folgende Verfahren zeigt Ereignisse mithilfe der ElastiCache Konsole an.

Um Ereignisse mit der ElastiCache Konsole anzuzeigen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste aller verfügbaren Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Events (Ereignisse).

Auf dem Bildschirm Ereignisse steht jede Zeile der Liste für ein Ereignis und zeigt die Ereignisquelle, den Ereignistyp (Cache-Cluster, cache-parameter-group cache-security-group, oder cache-subnet-group), die GMT-Zeit des Ereignisses und eine Beschreibung des Ereignisses an.

Mit der Option Filter können Sie angeben, ob alle Ereignisse oder nur Ereignisse eines bestimmten Typs in der Ereignisliste enthalten sein sollen.

ElastiCache Ereignisse anzeigen ()AWS CLI

Um mit dem eine Liste von ElastiCache Ereignissen zu generieren AWS CLI, verwenden Sie den Befehldescribe-events. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Der folgende Code listet bis zu 40 Cluster-Ereignisse auf.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

Mit dem folgenden Code werden alle Ereignisse der letzten 24 Stunden (1 440 Minuten) aufgelistet.

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

Die Ausgabe des Befehls `describe-events` sieht in etwa wie folgt aus:

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
{
  "Events": [
    {
      "SourceIdentifier": "my-mem-cluster",
      "SourceType": "cache-cluster",
      "Message": "Finished modifying number of nodes from 1 to 3",
      "Date": "2020-06-09T02:01:21.772Z"
    },
    {
      "SourceIdentifier": "my-mem-cluster",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0002 in availability zone us-west-2a",
      "Date": "2020-06-09T02:01:21.716Z"
    },
    {
      "SourceIdentifier": "my-mem-cluster",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0003 in availability zone us-west-2a",
      "Date": "2020-06-09T02:01:21.706Z"
    },
    {
      "SourceIdentifier": "my-mem-cluster",
      "SourceType": "cache-cluster",
      "Message": "Increasing number of requested nodes",
      "Date": "2020-06-09T01:58:34.178Z"
    },
    {
      "SourceIdentifier": "mycluster-0003-004",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0001 in availability zone us-west-2c",
      "Date": "2020-06-09T01:51:14.120Z"
    },
    {
      "SourceIdentifier": "mycluster-0003-004",
      "SourceType": "cache-cluster",

```

```

      "Message": "This cluster does not support persistence (ex: 'appendonly').
Please use a different instance type to enable persistence.",
      "Date": "2020-06-09T01:51:14.095Z"
    },
    {
      "SourceIdentifier": "mycluster-0003-004",
      "SourceType": "cache-cluster",
      "Message": "Cache cluster created",
      "Date": "2020-06-09T01:51:14.094Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-005",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0001 in availability zone us-west-2b",
      "Date": "2020-06-09T01:42:55.603Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-005",
      "SourceType": "cache-cluster",
      "Message": "This cluster does not support persistence (ex: 'appendonly').
Please use a different instance type to enable persistence.",
      "Date": "2020-06-09T01:42:55.576Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-005",
      "SourceType": "cache-cluster",
      "Message": "Cache cluster created",
      "Date": "2020-06-09T01:42:55.574Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-004",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0001 in availability zone us-west-2b",
      "Date": "2020-06-09T01:28:40.798Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-004",
      "SourceType": "cache-cluster",
      "Message": "This cluster does not support persistence (ex: 'appendonly').
Please use a different instance type to enable persistence.",
      "Date": "2020-06-09T01:28:40.775Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-004",

```



```
        "SourceType": "cache-cluster",
        "Message": "Cache cluster created",
        "Date": "2020-06-09T01:28:40.773Z"
    }
]
```

Weitere Informationen z. B. zu den verfügbaren Parametern und den zulässigen Parameterwerten finden Sie unter [describe-events](#).

ElastiCache Ereignisse anzeigen (ElastiCache API)

Verwenden Sie die DescribeEvents Aktion, um mithilfe der ElastiCache API eine Liste von ElastiCache Ereignissen zu generieren. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Mit dem folgenden Code werden die letzten 40 Cache-Cluster-Ereignisse aufgelistet.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeEvents
&MaxRecords=40
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&SourceType=cache-cluster
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Mit dem folgenden Code werden alle Cache-Cluster-Ereignisse der letzten 24 Stunden (1 440 Minuten) aufgelistet.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeEvents
&Duration=1440
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&SourceType=cache-cluster
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Die Ausgabe der oben angegebenen Aktionen sollte in etwa wie folgt aussehen:

```
<DescribeEventsResponse xmlns="http://elasticache.amazonaws.com/doc/2015-02-02/">
  <DescribeEventsResult>
    <Events>
      <Event>
        <Message>Cache cluster created</Message>
        <SourceType>cache-cluster</SourceType>
        <Date>2015-02-02T18:22:18.202Z</Date>
        <SourceIdentifier>mem01</SourceIdentifier>
      </Event>
      (...output omitted...)
    </Events>
  </DescribeEventsResult>
  <ResponseMetadata>
    <RequestId>e21c81b4-b9cd-11e3-8a16-7978bb24ffdf</RequestId>
  </ResponseMetadata>
</DescribeEventsResponse>
```

Weitere Informationen z. B. zu den verfügbaren Parametern und den zulässigen Parameterwerten finden Sie unter [DescribeEvents](#).

Ereignisbenachrichtigungen und Amazon SNS

ElastiCache kann Nachrichten mit Amazon Simple Notification Service (SNS) veröffentlichen, wenn wichtige Ereignisse in einem Cluster auftreten. Mit dieser Funktion können die Serverlisten auf Client-Computern aktualisiert werden, die mit einzelnen Cache-Knotenendpunkten eines Cache-Clusters verbunden sind.

Note

Weitere Informationen zum Amazon Simple Notification Service (SNS) sowie Informationen zu Preisen und Links zur Amazon-SNS-Dokumentation finden Sie auf der [Produktseite zu Amazon SNS](#).

Benachrichtigungen werden in einem bestimmten Amazon-SNS-Thema veröffentlicht. Für Benachrichtigungen müssen folgende Voraussetzungen erfüllt sein:

- Für ElastiCache Benachrichtigungen kann nur ein Thema konfiguriert werden.
- Das AWS Konto, dem das Amazon SNS SNS-Thema gehört, muss dasselbe Konto sein, das den Cluster besitzt, auf dem Benachrichtigungen aktiviert sind.
- Das Amazon-SNS-Thema, in dem Sie veröffentlichen, kann nicht verschlüsselt werden.

Note

Es ist möglich, ein verschlüsselt (im Ruhezustand befindliches) Amazon-SNS-Thema an den Cluster anzuhängen. Der Status des Themas in der ElastiCache Konsole wird jedoch als inaktiv angezeigt, wodurch das Thema effektiv vom Cluster getrennt wird, wenn Nachrichten an das ElastiCache Thema weitergeleitet werden.

- Das Amazon SNS SNS-Thema muss sich in derselben Region wie der ElastiCache Cluster befinden.

ElastiCache Events

Die folgenden ElastiCache Ereignisse lösen Amazon SNS SNS-Benachrichtigungen aus. Weitere Informationen zu Ereignisdetails finden Sie unter [ElastiCache Ereignisse anzeigen](#).

Ereignisname	Fehlermeldung	Description
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Ein Cache-Knoten wurde dem Cluster hinzugefügt und ist einsatzbereit.
ElastiCache: AddCacheNodeFailed aufgrund unzureichender freier IP-Adressen	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	Ein Cache-Knoten konnte nicht hinzugefügt werden, da nicht genug IP-Adressen verfügbar sind.
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	Ein oder mehrere Cluster-Parameter wurden geändert.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioning	Die Bereitstellung eines Clusters ist abgeschlossen, und die Cache-Knoten im

Ereignisname	Fehlermeldung	Description
	Complete <i>cluster-name-0001-005</i>	Cache-Cluster sind einsatzbereit.
ElastiCache: CacheClusterProvisioningFailed aufgrund eines inkompatiblen Netzwerkstatus	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	Es wurde versucht, einen neuen Cluster in einer nicht existierenden Virtual Private Cloud (VPC) zu starten.
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Die Skalierung für den Cache-Cluster wurde erfolgreich abgeschlossen.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	Die Erweiterung des Cache-Clusters ist fehlgeschlagen.

Ereignisname	Fehlermeldung	Description
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	Eines der folgenden Ereignisse ist aufgetreten: • Die Liste der für den Cluster autorisierten Cache-Sicherheitsgruppen wurde geändert. • Eine oder mehrere neue EC2 Sicherheitsgruppen wurden für eine der Cache-Sicherheitsgruppen, die dem Cluster zugeordnet sind, autorisiert. • Eine oder mehrere EC2 Sicherheitsgruppen wurden aus einer der Cache-Sicherheitsgruppen, die dem Cluster zugeordnet sind, gesperrt.

Ereignisname	Fehlermeldung	Description
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	ElastiCache hat festgestellt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, heruntergestuft oder nicht erreichbar ist, und hat begonnen, den Cache-Knoten zu ersetzen.  Note Der DNS-Eintrag für den ersetzten Cache-Knoten bleibt unverändert. In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Client-Bibliotheken verwenden den Cache-Knoten möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.

Ereignisname	Fehlermeldung	Description
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	ElastiCache hat festgestellt, dass der Host, auf dem ein Cacheknoten ausgeführt wird, heruntergefahren oder nicht erreichbar ist, und hat den Austausch des Cacheknotens abgeschlossen.  Note Der DNS-Eintrag für den ersetzten Cache-Knoten bleibt unverändert. In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Client-Bibliotheken verwenden den Cache-Knoten möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt ElastiCache wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.

Ereignisname	Fehlermeldung	Description
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Mindestens ein Cache-Knoten wurde neu gestartet. Nachricht (Memcached): "Cache node %s shutdown" Dann eine zweite Nachricht: "Cache node %s restarted"
ElastiCache: CertificateRenewalComplete (Nur Valkey oder Redis OSS)	ElastiCache:CertificateRenewalComplete	Das Amazon CA-Zertifikat wurde erfolgreich erneuert.
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	Die Replikationsgruppe wurde erfolgreich erstellt.
ElastiCache>DeleteCacheClusterComplete	ElastiCache>DeleteCacheClusterComplete : <i>cluster-name</i>	Das Löschen eines Clusters und aller zugehörigen Cache-Knoten ist abgeschlossen.
ElastiCache: FailoverComplete (Nur Valkey oder Redis OSS)	ElastiCache:FailoverComplete : <i>mycluster</i>	Failover zu einem Replikationsknoten wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	Die Anzahl der Replikate im Cluster wurde erhöht.

Ereignisname	Fehlermeldung	Description
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	Der Vorgang für das Hinzufügen von Replikaten zu Ihrem Cluster hat begonnen.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Ein Knoten in Ihrem Cluster, der ersetzt werden sollte, soll nicht länger ersetzt werden.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	Für einen zu ersetzenden Knoten in Ihrem Cluster wurde eine spätere Ersetzung im neuen Fenster geplant, das in der Benachrichtigung angegeben wurde. Weitere Informationen zu den möglichen Aktionen erhalten Sie unter Knoten ersetzen (Valkey und Redis OSS).
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	Ein Knoten in Ihrem Cluster soll während des in der Benachrichtigung beschriebenen Fensters ersetzt werden. Weitere Informationen zu den möglichen Aktionen erhalten Sie unter Knoten ersetzen (Valkey und Redis OSS).
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Ein Cache-Knoten wurde aus dem Cluster entfernt.

Ereignisname	Fehlermeldung	Description
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	Die Aufskalierung der Replikationsgruppe wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	Die Aufskalierung der Replikationsgruppe ist fehlgeschlagen.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Für den Knoten ist ein Self-Service-Update verfügbar.
ElastiCache: SnapshotComplete (Nur Valkey oder Redis OSS)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Ein Cache-Snapshot wurde erfolgreich abgeschlossen.
ElastiCache: SnapshotFailed (Nur Valkey oder Redis OSS)	SnapshotFailed : <i>cluster-name</i>	Ein Cache-Snapshot ist fehlgeschlagen. Weitere Informationen zur Ursache finden Sie in den Cache-Ereignissen des Clusters. Wenn Sie den Snapshot beschreiben (siehe DescribeSnapshots), ist dessen Status failed.

Verwandte Themen

- [ElastiCache Ereignisse anzeigen](#)

Protokollzustellung

Note

Slow Log wird für Valkey 7.x und höher sowie für Redis OSS-Cluster und Replikationsgruppen ab der Engine-Version 6.0 unterstützt.
Engine Log wird für Valkey 7.x und höher sowie für Redis OSS-Cluster und Replikationsgruppen ab Engine-Version 6.2 unterstützt.

Mit der Protokollzustellung können Sie [SLOWLOG oder Engine Log an](#) eines von zwei Zielen streamen:

- Amazon Data Firehose
- CloudWatch Amazon-Protokolle

Sie aktivieren und konfigurieren die Protokollzustellung, wenn Sie einen Cluster erstellen oder ändern mit ElastiCache APIs. Jeder Protokolleintrag wird in einem von zwei Formaten an das angegebene Ziel übermittelt: JSON oder TEXT.

Eine feste Anzahl von Slow-Protokolleinträgen wird regelmäßig von der Engine abgerufen. Je nach dem für den Engine-Parameter `slowlog-max-len` angegebenen Wert werden zusätzliche langsame Protokolleinträge möglicherweise nicht an das Ziel übermittelt.

Sie können die Versandkonfigurationen ändern oder die Protokollzustellung jederzeit deaktivieren. Verwenden Sie dazu die AWS Konsole oder eine der Optionen zum Ändern APIs, entweder [modify-cache-cluster](#) oder [modify-replication-group](#).

Sie müssen den `apply-immediately`-Parameter für alle Änderungen der Protokollzustellung setzen.

Note

Amazon CloudWatch Logs-Gebühren fallen an, wenn die Protokollzustellung aktiviert ist, auch wenn Protokolle direkt an Amazon Data Firehose geliefert werden. Weitere Informationen finden Sie im Abschnitt Verkaufte Logs in der [CloudWatch Amazon-Preisübersicht](#).

Inhalt eines langsamen Protokolleintrags

Das Slow Log enthält die folgenden Informationen:

- **CacheClusterId**— Die ID des Clusters
- **CacheNodeId**— Die ID des Cache-Knotens
- **Id** – Eine eindeutige progressive Kennung für jeden langsamen Protokoll-Eintrag
- **Timestamp** – Der Unix-Zeitstempel, zu dem der protokollierte Befehl verarbeitet wurde
- **Duration (Dauer)** – Die für die Ausführung benötigte Zeit in Mikrosekunden
- **Command** – Der vom Client verwendete Befehl. Zum Beispiel, `set foo bar` wo `foo` ist der Schlüssel und `bar` ist der Wert. ElastiCache ersetzt den tatsächlichen Schlüsselnamen und -wert durch, `(2 more arguments)` um zu verhindern, dass vertrauliche Daten offengelegt werden.
- **ClientAddress**— IP-Adresse und Port des Clients
- **ClientName**— Clientname, falls er über den `CLIENT SETNAME` Befehl gesetzt wurde

Inhalte eines Engine-Protokolleintrags

Das ElastiCache Engine-Protokoll enthält die folgenden Informationen:

- **CacheClusterId**— Die ID des Clusters
- **CacheNodeId**— Die ID des Cache-Knotens
- **Protokollebene** — **LogLevel** kann eine der folgenden Optionen sein: `VERBOSE("-")`, `NOTICE("*")`, `WARNING("#")`.
- **Zeit** – Die UTC-Uhrzeit der protokollierten Nachricht. Die Zeit ist in folgendem Format angegeben: `"DD MMM YYYY hh:mm:ss.ms UTC"`
- **Rolle** – Rolle des Knotens, von dem das Protokoll ausgegeben wird. Dabei kann es sich um einen der folgenden Werte handeln: „M“ für Primär, „S“ für Replikat, „C“ für einen untergeordneten Writer-Prozess, an dem gearbeitet wird, RDB/AOF oder „X“ für Sentinel.
- **Nachricht** — Engine-Log-Meldung.

Berechtigungen zum Konfigurieren der Protokollierung

Sie müssen die folgenden IAM-Berechtigungen in Ihre user/role IAM-Richtlinie aufnehmen:

- logs:CreateLogDelivery
- logs:UpdateLogDelivery
- logs>DeleteLogDelivery
- logs:GetLogDelivery
- logs:ListLogDeliveries

Weitere Informationen finden Sie unter [Übersicht über die Zugriffsverwaltung: Berechtigungen und Richtlinien](#).

Spezifikationen des Protokolltyps und des Protokollformats

Slow-Protokoll

Das langsame Protokoll unterstützt sowohl JSON als auch TEXT

Im Folgenden sehen Sie ein Beispiel für das JSON-Format:

```
{
  "CacheClusterId": "logslowxxxxmsxj",
  "CacheNodeId": "0001",
  "Id": 296,
  "Timestamp": 1605631822,
  "Duration (us)": 0,
  "Command": "GET ... (1 more arguments)",
  "ClientAddress": "192.168.12.104:55452",
  "ClientName": "logslowxxxxmsxj##"
}
```

Im Folgenden sehen Sie ein Beispiel für das TEXT-Format:

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

Engine-Protokoll

Das Engine-Protokoll unterstützt sowohl JSON als auch TEXT

Im Folgenden sehen Sie ein Beispiel für das JSON-Format:

```
{
  "CacheClusterId": "xxxxxxxxxy-engine-log-test",
  "CacheNodeId": "0001",
  "LogLevel": "VERBOSE",
  "Role": "M",
  "Time": "12 Nov 2020 01:28:57.994 UTC",
  "Message": "Replica is waiting for next BGSAVE before synchronizing with the primary.
Check back later"
}
```

Im Folgenden sehen Sie ein Beispiel für das TEXT-Format:

```
xxxxxxxxxy-engine-log-test/0001:M 29 Oct 2020 20:12:20.499 UTC * A slow-running Lua
script detected that is still in execution after 1000 microseconds.
```

ElastiCache Ziele protokollieren

In diesem Abschnitt werden die Protokollierungsziele beschrieben, die Sie für Ihre ElastiCache Protokolle auswählen können. Jeder Abschnitt enthält Anleitungen zum Konfigurieren der Protokollierung für den Zieltyp und Informationen zu jedem Verhalten, das für den jeweiligen Zieltyp spezifisch ist. Nachdem Sie Ihr Protokollierungsziel konfiguriert haben, können Sie dessen Spezifikationen für die ElastiCache Protokollierungskonfiguration angeben, um mit der Protokollierung zu beginnen.

Themen

- [CloudWatch Amazon-Protokolle](#)
- [Amazon Data Firehose](#)

CloudWatch Amazon-Protokolle

- Sie geben eine CloudWatch Logs-Protokollgruppe an, in die die Protokolle geliefert werden.
- Protokolle von mehreren Valkey- oder Redis OSS-Clustern und Replikationsgruppen können an dieselbe Protokollgruppe übermittelt werden.
- Für jeden Knoten innerhalb eines Clusters oder einer Replikationsgruppe wird ein neuer Protokollstream erstellt, und die Protokolle werden an die jeweiligen Protokollstreams übermittelt. Der Name des Protokolldatenstroms verwendet das folgende Format: `elasticache/${engine-name}/${cache-cluster-id}/${cache-node-id}/${log-type}`

Berechtigungen zum Veröffentlichen von Protokollen in CloudWatch Logs

Sie müssen über die folgenden Berechtigungseinstellungen verfügen, um das Senden von Protokollen an eine CloudWatch Logs-Protokollgruppe konfigurieren ElastiCache zu können:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "ElastiCacheLogging"
    },
    {
      "Sid": "ElastiCacheLoggingCWL",
      "Action": [
        "logs:PutResourcePolicy",
        "logs:DescribeResourcePolicies",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Weitere Informationen finden Sie unter An [Protokolle gesendete CloudWatch Protokolle](#).

Amazon Data Firehose

- Sie geben einen Firehose-Lieferstream an, in den die Protokolle geliefert werden.
- Protokolle aus mehreren Valkey- oder Redis OSS-Clustern und Replikationsgruppen können an denselben Delivery Stream übermittelt werden.
- Protokolle von jedem Knoten innerhalb eines Clusters oder einer Replikationsgruppe werden an denselben Lieferstream übermittelt. Sie können Protokollnachrichten von verschiedenen Cache-Knoten basierend auf `cache-cluster-id` und `cache-node-id` in jeder Protokollnachricht unterscheiden.
- Die Lieferung von Baumstämmen an Firehose ist derzeit in der Region Asien-Pazifik (Osaka) nicht verfügbar.

Berechtigungen zum Veröffentlichen von Protokollen auf Firehose

Sie benötigen die folgenden Berechtigungen, um das Senden von Protokollen an einen Amazon Kinesis Data Firehose Firehose-Lieferstream zu konfigurieren ElastiCache .

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "ElastiCacheLogging"
    },
    {
      "Sid": "ElastiCacheLoggingFHSLR",
      "Action": [
```



```
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Sid": "ElastiCacheLoggingFH",
      "Action": [
        "firehose:TagDeliveryStream"
      ],
      "Resource": "arn:aws:iam::*:role/*",
      "Effect": "Allow"
    }
  ]
}
```

Angeben der Protokollzustellung mithilfe der Konsole

Mit dem können AWS-Managementkonsole Sie einen Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert) erstellen, indem Sie den Schritten unter folgen, [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#) oder einen Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert) mithilfe der Schritte unter erstellen. [Erstellen eines Valkey- oder Redis OSS-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#) In beiden Fällen konfigurieren Sie die Protokollzustellung wie folgt;

1. Wählen Sie unter Erweiterte Einstellungen die Option Logs aus und aktivieren Sie dann entweder Slow Logs oder Engine Logs.
2. Unter Protokollformat wählen Sie entweder Text oder JSON aus.
3. Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
4. Wählen Sie unter Protokollziel entweder Neu erstellen aus und geben Sie entweder Ihren Amazon S3 S3-Bucket-Namen, CloudWatchLogs Protokollgruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und dann entweder Ihren CloudWatch Logs-Gruppennamen oder Ihren Kinesis Data Firehose Firehose-Streamnamen aus.

Beim Ändern eines Clusters:

Sie können entweder die enable/disable Protokollzustellung protokollieren oder entweder den Zieltyp, das Format oder das Ziel ändern:

1. Melden Sie sich bei der Konsole an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder Redis OSS-Cluster aus.
3. Suchen Sie in der Liste der Cluster den Cluster aus, den Sie ändern möchten. Wählen Sie den Clusternamen und nicht das Kontrollkästchen daneben.
4. Wählen Sie auf der Seite Clusternamen die Registerkarte Protokolle.
5. Um Logs zu enable/disable verlangsamen, wählen Sie entweder Slow Logs aktivieren oder Slow Logs deaktivieren.
6. Wählen Sie für enable/disable Engine-Logs entweder Engine-Logs aktivieren oder Engine-Logs deaktivieren.
7. Um Ihre Konfiguration zu ändern, wählen Sie entweder Modify slow logs (Langsame Protokolle ändern) oder Modify engine logs (Engine-Protokolle ändern) aus:
 - Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen aus und geben Sie entweder Ihren CloudWatchLogs Protokollgruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen ein. Oder wählen Sie Select existing und wählen Sie dann entweder Ihren CloudWatchLogs Log-Gruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen.

Angeben der Protokollzustellung mit dem AWS CLI

Slow-Protokoll

Erstellen Sie eine Replikationsgruppe mit langsamer Protokollzustellung an CloudWatch Logs.

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id test-slow-log \  
  --replication-group-description test-slow-log \  
  --engine redis \  
  --cache-node-type cache.r5.large \  
  --num-cache-clusters 2 \  
  --log-delivery-configurations '{
```

```
"LogType":"slow-log",
"DestinationType":"cloudwatch-logs",
"DestinationDetails":{
  "CloudWatchLogsDetails":{
    "LogGroup":"my-log-group"
  }
},
"LogFormat":"json"
}'
```

Für Windows:

```
aws elasticache create-replication-group ^
--replication-group-id test-slow-log ^
--replication-group-description test-slow-log ^
--engine redis ^
--cache-node-type cache.r5.large ^
--num-cache-clusters 2 ^
--log-delivery-configurations '{
  "LogType":"slow-log",
  "DestinationType":"cloudwatch-logs",
  "DestinationDetails":{
    "CloudWatchLogsDetails":{
      "LogGroup":"my-log-group"
    }
  },
  "LogFormat":"json"
}'
```

Ändern Sie eine Replikationsgruppe, um das Protokoll langsam in CloudWatch Logs zu speichern

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
--replication-group-id test-slow-log \
--apply-immediately \
--log-delivery-configurations '{
  "LogType":"slow-log",
  "DestinationType":"cloudwatch-logs",
  "DestinationDetails":{
    "CloudWatchLogsDetails":{
```

```
    "LogGroup": "my-log-group"
  }
},
"LogFormat": "json"
}'
```

Für Windows:

```
aws elasticache modify-replication-group ^
--replication-group-id test-slow-log ^
--apply-immediately ^
--log-delivery-configurations '
{
  "LogType": "slow-log",
  "DestinationType": "cloudwatch-logs",
  "DestinationDetails": {
    "CloudWatchLogsDetails": {
      "LogGroup": "my-log-group"
    }
  },
  "LogFormat": "json"
}'
```

Ändern einer Replikationsgruppe, um die langsame Protokollzustellung zu deaktivieren

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
--replication-group-id test-slow-log \
--apply-immediately \
--log-delivery-configurations '
{
  "LogType": "slow-log",
  "Enabled": false
}'
```

Für Windows:

```
aws elasticache modify-replication-group ^
--replication-group-id test-slow-log ^
--apply-immediately ^
--log-delivery-configurations '
```

```
{
  "LogType":"slow-log",
  "Enabled":false
}'
```

Engine-Protokoll

Erstellen Sie eine Replikationsgruppe mit der Übermittlung von CloudWatch Engine-Protokollen an Logs.

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \
  --replication-group-id test-slow-log \
  --replication-group-description test-slow-log \
  --engine redis \
  --cache-node-type cache.r5.large \
  --num-cache-clusters 2 \
  --log-delivery-configurations '{
    "LogType":"engine-log",
    "DestinationType":"cloudwatch-logs",
    "DestinationDetails":{
      "CloudWatchLogsDetails":{
        "LogGroup":"my-log-group"
      }
    },
    "LogFormat":"json"
  }'
```

Für Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id test-slow-log ^
  --replication-group-description test-slow-log ^
  --engine redis ^
  --cache-node-type cache.r5.large ^
  --num-cache-clusters 2 ^
  --log-delivery-configurations '{
    "LogType":"engine-log",
    "DestinationType":"cloudwatch-logs",
    "DestinationDetails":{
      "CloudWatchLogsDetails":{
        "LogGroup":"my-log-group"
      }
    }
  }'
```

```
    }  
  },  
  "LogFormat":"json"  
}'
```

Ändern Sie eine Replikationsgruppe, um das Engine-Protokoll an Firehose zu übermitteln

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test-slow-log \  
  --apply-immediately \  
  --log-delivery-configurations '  
{  
  "LogType":"engine-log",  
  "DestinationType":"kinesis-firehose",  
  "DestinationDetails":{  
    "KinesisFirehoseDetails":{  
      "DeliveryStream":"test"  
    }  
  },  
  "LogFormat":"json"  
}'
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
{  
  "LogType":"engine-log",  
  "DestinationType":"kinesis-firehose",  
  "DestinationDetails":{  
    "KinesisFirehoseDetails":{  
      "DeliveryStream":"test"  
    }  
  },  
  "LogFormat":"json"  
}'
```

Ändern einer Replikationsgruppe zum Wechseln in das Engine-Format

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test-slow-log \  
  --apply-immediately \  
  --log-delivery-configurations '  
  {  
    "LogType":"engine-log",  
    "LogFormat":"json"  
  }'
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
  {  
    "LogType":"engine-log",  
    "LogFormat":"json"  
  }'
```

Ändern einer Replikationsgruppe zur Deaktivierung der Übermittlung von Engine-Protokollen

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test-slow-log \  
  --apply-immediately \  
  --log-delivery-configurations '  
  {  
    "LogType":"engine-log",  
    "Enabled":false  
  }'
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
  {
```

```
"LogType": "engine-log",  
"Enabled": false  
'
```

Überwachung der Nutzung mit CloudWatch Metrics

ElastiCache bietet Metriken, mit denen Sie Ihre Cluster überwachen können. Sie können auf diese Metriken zugreifen über CloudWatch. Weitere Informationen zu CloudWatch finden Sie in der [CloudWatch Dokumentation](#).

ElastiCache bietet sowohl Metriken auf Host-Ebene (z. B. CPU-Auslastung) als auch Metriken, die für die Cache-Engine-Software spezifisch sind (z. B. Cache-Abrufe und Cache-Fehlschläge). Diese Metriken werden für jeden Cache-Knoten in 60-Sekunden-Intervallen gemessen und veröffentlicht.

Important

Sie sollten erwägen, CloudWatch Alarme für bestimmte wichtige Messwerte einzurichten, damit Sie benachrichtigt werden, wenn sich die Leistung Ihres Clusters zu verschlechtern beginnt. Weitere Informationen finden Sie unter [Welche Metriken sollte ich überwachen?](#) in diesem Handbuch.

Themen

- [Metriken auf Host-Ebene](#)
- [Metriken für Valkey und Redis OSS](#)
- [Metriken für Memcached](#)
- [Welche Metriken sollte ich überwachen?](#)
- [Auswählen von Metrikstatistiken und -zeiträumen](#)
- [Überwachung von CloudWatch Cluster- und Knotenmetriken](#)

Metriken auf Host-Ebene

Der AWS/ElastiCache-Namespace enthält die folgenden Metriken auf Host-Ebene für einzelne Cache-Knoten. Diese Metriken werden für jeden Cache-Knoten in 60-Sekunden-Intervallen erfasst und veröffentlicht.

Weitere Informationen finden Sie auch unter:

- [Metriken für Valkey und Redis OSS](#)

Metrik	Beschreibung	Einheit
CPUUtilization	Der Prozentsatz der CPU-Nutzung für den gesamten Host. Da Valkey und Redis OSS Single-Threading verwenden, empfehlen wir Ihnen, die EngineCPUUtilization Metrik für Knoten mit 4 oder mehr v zu überwachen. CPUs	Prozent
CPUCreditBalance	Die Anzahl verdienter CPU-Guthaben, die eine Instance angesammelt hat, seit sie gestartet wurde. Bei T2 Standard beinhaltet der CPUCredit Saldo auch die Anzahl der angesammelten Startguthaben. Guthaben werden auf dem Guthaben-Konto angesammelt, nachdem sie verdient wurden, und davon entfernt, wenn sie verbraucht werden. Der Guthaben-Kontostand hat ein maximales Limit, das anhand der Instance-Größe bestimmt wird. Nachdem das Limit erreicht ist, verfallen alle neu verdienten Guthabepunkte. Für T2 Standard zählen Startguthaben nicht zum Limit. Die Guthaben im CPUCredit Saldo stehen der Instance zur Verfügung, um die CPU-Grundauslastung zu überschreiten. Die Metriken für CPU-Guthaben sind nur mit einer fünfminütigen Frequenz verfügbar. Diese Metrik ist nicht für T2-Instances mit Spitzenleistung verfügbar.	Guthaben (vCPU-Minuten)

Metrik	Beschreibung	Einheit
CPUCreditUsage	Die Anzahl der von der Instance für die CPU-Nutzung verbrauchten CPU-Guthaben. Ein CPU-Guthaben entspricht einer vCPU, die eine Minute lang bei 100% Auslastung läuft, oder einer äquivalenten Kombination aus vCPUs, Auslastung und Zeit (z. B. eine vCPU, die zwei Minuten lang bei 50% Auslastung läuft, oder zwei v, die zwei Minuten lang bei 25% Auslastung CPUs laufen). Die Metriken für CPU-Guthaben sind nur mit einer fünfminütigen Frequenz verfügbar. Wenn Sie einen Zeitraum von mehr als fünf Minuten angeben, verwenden Sie die Summen-Statistik anstelle der Durchschnitts-Statistik. Diese Metrik ist nicht für T2-Instances mit Spitzenleistung verfügbar.	Guthaben (vCPU-Minuten)
FreeableMemory	Größe des freien Arbeitsspeichers auf dem Host. Dies ist von RAM, Puffern und Cache abgeleitet, die vom Betriebssystem als freigebbar eingestuft werden.	Bytes
NetworkBytesIn	Anzahl der Byte, die der Host aus dem Netzwerk gelesen hat.	Bytes
NetworkBytesOut	Anzahl der von der Instance auf allen Netzwerkschnittstellen gesendeten Byte.	Bytes
NetworkPacketsIn	Anzahl der von der Instance auf allen Netzwerkschnittstellen empfangenen Pakete. Diese Metrik gibt das an eine einzelne Instance eingehende Netzwerkdatenvolumen an, ausgedrückt in Anzahl an Paketen.	Anzahl

Metrik	Beschreibung	Einheit
NetworkPacketsOut	Anzahl der von der Instance auf allen Netzwerkschnittstellen gesendeten Pakete. Diese Metrik gibt das von einer einzelnen Instance ausgehende Netzwerkdatenvolumen an, ausgedrückt in Anzahl an Paketen.	Anzahl
NetworkBandwidthInAllowanceExceeded	Die Anzahl der Pakete, die in die Warteschlange gestellt oder verworfen wurden, da die eingehende aggregierte Bandbreite das Maximum für die Instance überschritten hat.	Anzahl
NetworkConntrackAllowanceExceeded	Die Anzahl der verworfenen Pakete, weil die Verbindungsverfolgung das Maximum für die Instance überschritten hat und keine neuen Verbindungen hergestellt werden konnten. Dies kann zu einem Paketverlust für den Datenverkehr zur oder von der Instance führen.	Anzahl
NetworkBandwidthOutAllowanceExceeded	Die Anzahl der Pakete, die in die Warteschlange gestellt oder verworfen wurden, weil die ausgehende aggregierte Bandbreite das Maximum für die Instance überschritten hat.	Anzahl
NetworkPacketsPerSecondAllowanceExceeded	Die Anzahl der Pakete, die in Warteschlangen gestellt oder verworfen wurden, weil Anzahl der bidirektionalen Pakete pro Sekunde das Maximum für die Instance überschritten hat.	Anzahl
NetworkMaxBytesIn	Die maximale Anzahl an empfangenen Bytes pro Sekunde pro Sekunde pro Minute.	Bytes
NetworkMaxBytesOut	Die maximale Anzahl an übertragenen Byte pro Sekunde pro Sekunde pro Minute.	Bytes
NetworkMaxPacketsIn	Die maximale Anzahl pro Sekunde empfangener Pakete pro Sekunde pro Minute.	Anzahl

Metrik	Beschreibung	Einheit
NetworkMaxPacketsOut	Die maximale Anzahl an übertragenen Paketen pro Sekunde pro Sekunde pro Minute.	Anzahl
SwapUsage	Größe des belegten Auslagerungsspeichers auf dem Host.	Bytes

Metriken für Valkey und Redis OSS

Der Amazon ElastiCache Namespace umfasst die folgenden Valkey- und Redis-OSS-Metriken. Diese Metriken sind identisch, wenn die Valkey-Engine verwendet wird.

Mit Ausnahme von ReplicationLag, EngineCPUUtilization SuccessfulWriteRequestLatencySuccessfulReadRequestLatency, und werden diese Metriken aus dem info Befehl abgeleitet. Jede Metrik wird zu jeder Cache-Knotenebene berechnet.

Eine vollständige Dokumentation des info Befehls finden Sie unter <http://valkey.io/commands/info>.

Weitere Informationen finden Sie auch unter:

- [Metriken auf Host-Ebene](#)

Metrik	Description	Einheit
ActiveDefragHits	Die Anzahl der Werteneuzuweisungen pro Minute, die der aktive Defragmentierungsprozess durchführt. Dies wird aus den active_defrag_hits Statistiken von INFO abgeleitet.	Zahl
AuthenticationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche, sich mit dem AUTH-Befehl bei Valkey oder Redis OSS zu authentifizieren. Weitere Informationen zu einzelnen Authentifizierungsfehlern finden Sie mit dem Befehl ACL LOG . Wir empfehlen, hierauf einen Alarm zu	Anzahl

Metrik	Description	Einheit
	setzen, um unberechtigte Zugriffsversuche zu erkennen.	
	Die Gesamtzahl der Byte, die von Valkey oder Redis OSS für alle Zwecke zugewiesen wurden, einschließlich des Datensatzes, der Puffer usw.	Bytes
BytesUsedForCache	Dimension: Tier=Memory für Valkey- oder Redis-OSS-Cluster mit Datenzuweisung ElastiCache : Die Gesamtzahl der für den Cache verwendeten Byte pro Speicher. Dies ist der Wert der used_memory Statistik bei INFO.	Bytes
	Dimension: Tier=SSD für Valkey- oder Redis-OSS-Cluster mit Datenzuweisung ElastiCache : Die Gesamtzahl der Byte, die von SSD für den Cache verwendet werden.	Bytes
BytesReadFromDisk	Gesamtzahl der von der Festplatte pro Minute gelesenen Bytes. Wird nur für Cluster mit Datenzuweisung ElastiCache unterstützt.	Bytes
BytesWrittenToDisk	Gesamtzahl der Bytes, die pro Minute auf den Datenträger geschrieben werden. Wird nur für Cluster mit Datenzuweisung ElastiCache unterstützt.	Bytes
CacheHits	Die Anzahl der erfolgreichen schreibgeschützten Schlüsselsuchereignisse im Hauptverzeichnis. Dies wird aus der keyspace_hits Statistik von INFO abgeleitet.	Anzahl

Metrik	Description	Einheit
CacheMisses	Die Anzahl der nicht erfolgreichen schreibgeschützten Schlüsselsuchereignisse im Hauptverzeichnis. Dies wird aus den <code>keyspace_misses</code> Statistiken von INFO abgeleitet.	Anzahl
CommandAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, Befehle auszuführen, für deren Aufruf sie keine Berechtigung haben. Weitere Informationen zu einzelnen Authentifizierungsfehlern finden Sie mit dem Befehl ACL LOG . Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl
CacheHitRate	Zeigt die Nutzungseffizienz der Valkey- oder Redis OSS-Instanz an. Wenn das Cache-Verhältnis unter ~0,8 liegt, bedeutet dies, dass eine erhebliche Anzahl von Schlüsseln bereinigt wurde, abgelaufen oder nicht vorhanden ist. Dies wird mit <code>cache_hits</code> - und <code>cache_misses</code> -Statistiken auf folgende Weise berechnet: $\text{cache\_hits} / (\text{cache\_hits} + \text{cache\_misses})$.	Prozent
ChannelAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, auf Kanäle zuzugreifen, für die sie keine Zugriffsberechtigung haben. Weitere Informationen zu einzelnen Authentifizierungsfehlern finden Sie mit dem Befehl ACL LOG . Wir empfehlen, für diese Metrik einen Alarm festzulegen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl

Metrik	Description	Einheit
<code>CurrConnections</code>	Die Anzahl der Client-Verbindungen, ausgenommen Verbindungen von Read Replicas. ElastiCache verwendet jeweils 4 bis 6 der Verbindungen, um den Cluster zu überwachen. Dies wird aus der <code>connected_clients</code> Statistik bei INFO abgeleitet.	Anzahl
<code>CurrItems</code>	Anzahl der Elemente im Cache. Dies wird aus der <code>keyspace</code> Statistik abgeleitet, die alle Schlüssel im gesamten Schlüsselraum summiert.	Anzahl
	Dimension: Tier=Memory für Cluster mit Datenzuweisung ElastiCache . Anzahl der Elemente im Speicher.	Anzahl
	Dimension: Tier=SSD (Solid-State-Laufwerke) für Cluster mit Datenzuweisung ElastiCache . Anzahl der Elemente im SSD.	Anzahl
<code>CurrVolatileItems</code>	Gesamtzahl der Schlüssel in allen Datenbanken mit festgelegtem TTL. Dies wird aus der <code>expires</code> Statistik abgeleitet, bei der alle Schlüssel mit einem TTL-Satz im gesamten Schlüsselraum summiert werden.	Anzahl

Metrik	Description	Einheit
DatabaseCapacityUsagePercentage	Prozentsatz der gesamten Datenkapazität für den Cluster, die genutzt wird. <u>Bei Data-Tiered-Instances wird die Metrik wie folgt berechnet <code>used_memory</code> und $(used\_memory - mem\_not\_counted\_for\_evict + SSD\ used) / (maxmemory + SSD\ total\ capacity)$ <code>maxmemory</code> stammt aus INFO.</u> In allen anderen Fällen wird die Metrik anhand von <code>used_memory/maxmemory</code> berechnet.	Prozent
DatabaseCapacityUsageCountedForEvictionPercentage	Prozentsatz der gesamten Datenkapazität für den Cluster, die genutzt wird, ausschließlich des Speichers, der für Overhead und COB verwendet wird. Diese Metrik wird wie folgt berechnet: $used\_memory - mem\_not\_counted\_for\_evict / maxmemory$ Bei Daten-Tiering-Instances wird die Metrik wie folgt berechnet: $(used\_memory + SSD\ used) / (maxmemory + SSD\ total\ capacity)$ wo <code>used_memory</code> und <code>maxmemory</code> sind aus INFO entnommen	Prozent
DatabaseMemoryUsagePercentage	Prozentsatz des Speichers für den Cluster, der genutzt wird. Dies wird mit Hilfe <code>used_memory/maxmemory</code> von From INFO berechnet.	Prozent

Metrik	Description	Einheit
DatabaseMemoryUsageCountedForEvictPercentage	Prozentsatz des Speichers für den Cluster, der genutzt wird, ausschließlich des Speichers, der für Overhead und COB verwendet wird.. Dies wird mithilfe <code>used_memory-mem_not_counted_for_evict/maxmemory</code> von <code>from INFO</code> berechnet.	Prozent
DB0AverageTTL	Macht <code>avg_ttl DBO</code> aus dem Befehl <code>keyspace</code> Statistik des <code>INFO</code>-Befehls verfügbar. Bei Replikaten laufen die Schlüssel nicht ab, die Replikate warten, bis die Schlüssel auf den Primärknoten ablaufen. Wenn auf einem Primärknoten ein Schlüssel abläuft (oder der Schlüssel wegen LRU bereinigt wird), synthetisiert der Primärschlüssel einen <code>DEL</code> -Befehl, der an alle Replikate übertragen wird. Daher lautet <code>DB0AverageTTL</code> für Replikatknoten 0, da dort keine Schlüssel ablaufen und somit TTL nicht verfolgt wird.	Millisekunden

Metrik	Description	Einheit
EngineCPUUtilization	Stellt die CPU-Auslastung des Valkey- oder Redis-OSS-Engine-Threads bereit. Da Valkey und Redis OSS Single-Threading verwenden, können Sie diese Metrik verwenden, um die Auslastung des Prozesses selbst zu analysieren. Die EngineCPUUtilization Metrik bietet einen genaueren Überblick über den Prozess. Sie können dies in Verbindung mit der CPUUtilization -Metrik verwenden. CPUUtilization legt die CPU-Auslastung für die Server-Instance als Ganzes offen, einschließlich anderer Betriebssystem- und Verwaltungsprozesse. Verwenden Sie bei größeren Knotentypen mit vier V CPUs oder mehr die EngineCPUUtilization Metrik, um Schwellenwerte für die Skalierung zu überwachen und festzulegen.	Prozent

 **Note**

Auf einem ElastiCache Host überwachen Hintergrundprozesse den Host, um eine verwaltete Datenbankumgebung zu gewährleisten. Diese Hintergrundprozesse können einen erheblichen Teil der CPU-Workload beanspruchen. Bei größeren Hosts mit mehr als zwei V ist dies nicht signifikant CPUs. Es kann jedoch kleinere Hosts mit 2 V CPUs oder weniger betreffen. Wenn Sie nur die EngineCPUUtilization Metrik überwachen, sind Ihnen Situationen nicht bewusst, in denen der Host sowohl durch eine hohe CPU-Auslastung durch Valkey oder

Metrik	Description	Einheit
	Redis OSS als auch durch eine hohe CPU-Auslastung durch die Hintergrundüberwachungsprozesse überlastet ist. Daher empfehlen wir, die CPUUtilization Metrik für Hosts mit zwei V oder weniger zu überwachen. CPUs	
Evictions	Anzahl der Schlüssel, die infolge des maxmemory -Grenzwertes bereinigt worden sind. Dies wird aus der evicted_keys Statistik von INFO abgeleitet.	Anzahl
GlobalDatastoreReplicationLag	Dies ist die Verzögerung zwischen dem Primärknoten der sekundären Region und dem Primärknoten der primären Region. Bei aktiviertem Valkey oder Redis OSS im Clustermodus gibt die Verzögerung die maximale Verzögerung zwischen den Shards an.	Sekunden
IamAuthenticationExpirations	Die Gesamtzahl der abgelaufenen IAM-authentifizierten Valkey- oder Redis OSS-Verbindungen. Weitere Informationen über Authentifizieren mit IAM finden Sie im Benutzerhandbuch.	Anzahl
IamAuthenticationThrottling	Die Gesamtzahl der gedrosselten IAM-authentifizierten Valkey- oder Redis OSS AUTH- oder HELLO-Anfragen. Weitere Informationen über Authentifizieren mit IAM finden Sie im Benutzerhandbuch.	Anzahl
IsMaster	Gibt an, ob der Knoten der Primärknoten des aktuellen Shard/Clusters ist. Die Metrik kann entweder 0 (nicht primär) oder 1 (primär) sein.	Anzahl

Metrik	Description	Einheit
KeyAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, auf Schlüssel zuzugreifen, für die sie keine Zugriffsberechtigung haben. Weitere Informationen zu einzelnen Authentifizierungsfehlern finden Sie mit dem Befehl ACL LOG . Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl
KeysTracked	Die Anzahl der Schlüssel, die von Valkey oder Redis OSS Key Tracking verfolgt werden, als Prozentsatz von. <code>tracking-table-max-keys</code> Die Schlüsselverfolgung wird verwendet, um das clientseitige Caching zu unterstützen und Clients zu benachrichtigen, wenn Schlüssel geändert werden.	Anzahl
MemoryFragmentationRatio	Gibt die Effizienz bei der Speicherzuweisung der Valkey- oder Redis OSS-Engine an. Bestimmte Schwellenwerte weisen auf unterschiedliche Verhaltensweisen hin. Der empfohlene Wert ist eine Fragmentierung über 1,0. Dies wird anhand von INFO berechnet mem_fragmentation_ratio statistic .	Zahl

Metrik	Description	Einheit
NewConnections	Gesamtanzahl der Verbindungen, die in diesem Zeitraum vom Server akzeptiert worden sind. Dies wird aus der <code>total_connections_received</code> Statistik bei INFO abgeleitet.  Note Wenn Sie ElastiCache für Redis OSS Version 5 oder niedriger verwenden, werden zwischen zwei und vier der von dieser Metrik gemeldeten Verbindungen ElastiCache zur Überwachung des Clusters verwendet. ElastiCache Bei Verwendung von Redis OSS Version 6 oder höher sind die Verbindungen, die ElastiCache zur Überwachung des Clusters verwendet werden, jedoch nicht in dieser Metrik enthalten.	Anzahl
NumItemsReadFromDisk	Die Gesamtzahl der pro Minute von der Festplatte abgerufenen Elemente. Wird nur für Cluster mit Datenzuweisung ElastiCache unterstützt.	Anzahl
NumItemsWrittenToDisk	Die Gesamtzahl der pro Minute auf die Festplatte geschriebenen Elemente. Wird nur für Cluster mit Datenzuweisung ElastiCache unterstützt.	Anzahl

Metrik	Description	Einheit
MasterLinkHealthStatus	Dieser Status kann zwei Werte annehmen: 0 oder 1. Der Wert 0 gibt an, dass die Daten im ElastiCache Primärknoten nicht synchron sind, wenn Valkey oder Redis OSS aktiviert sind. EC2 Der Wert 1 bedeutet, dass die Daten synchronisiert sind. Verwenden Sie den CompleteMigration API-Vorgang, um die Migration abzuschließen.	Boolesch
Reclaimed	Gesamtanzahl der Schlüsselablaufereignisse Dies wird aus der expired_keys Statistik von INFO abgeleitet.	Anzahl
ReplicationBytes	Für Knoten in einer replizierten Konfiguration gibt ReplicationBytes die Anzahl der Bytes an, die der Primärknoten an alle seine Replikationen sendet. Diese Metrik gibt die Schreiblast auf der Replikationsgruppe an. Dies wird aus der master_repl_offset Statistik bei INFO abgeleitet.	Bytes
ReplicationLag	Diese Metrik ist nur für einen als Read Replica laufenden Knoten verfügbar. Sie stellt die Zeitverzögerung in Sekunden dar, mit der die Replica die vom primären Knoten kommenden Änderungen anwendet. Für Valkey 7.2 und höher und ab Redis OSS 5.0.6 kann die Verzögerung in Millisekunden gemessen werden.	Sekunden

Metrik	Description	Einheit
SaveInProgress	Diese binäre Metrik gibt 1 zurück, wenn ein Hintergrundspeichervorgang (vergabelt oder unvergabelt) ausgeführt wird, andernfalls beträgt der Wert 0. Ein Hintergrundspeichervorgang wird normalerweise während Snapshots und Synchronisierungen verwendet. Diese Vorgänge können die Leistung beeinträchtigen. Mithilfe der SaveInProgress - Metrik können Sie diagnostizieren, ob die Leistungsbeeinträchtigung von einem Hintergrundspeichervorgang verursacht wurde. Dies wird aus der Statistik von INFO abgeleitet. rdb_bgsave_in_progress	Boolesch

Metrik	Description	Einheit
TrafficManagementActive	Gibt an, ob OSS ElastiCache für Redis den Datenverkehr aktiv verwaltet, indem der Datenverkehr, der eingehenden Befehlen zugewiesen wird, überwacht oder repliziert wird. Der Datenverkehr wird verwaltet, wenn mehr Befehle an den Knoten gesendet werden, als von Valkey oder Redis OSS verarbeitet werden können, und dient dazu, die Stabilität und den optimalen Betrieb der Engine aufrechtzuerhalten. Datenpunkte von 1 können darauf hinweisen, dass der Knoten für die bereitgestellte Workload unterdimensioniert ist.  Note Wenn diese Metrik aktiv bleibt, evaluieren Sie den Cluster, um zu entscheiden, ob eine Hoch- oder Aufskalierung erforderlich ist. Zu den verwandten Metriken gehören NetworkBandwidthOutAllowanceExceeded und EngineCPUUtilization .	Boolesch
SuccessfulWriteRequestLatency	Latenz erfolgreicher Schreibanforderungen. Gültige Statistiken: Durchschnitt, Summe, Min, Max, Anzahl der Stichproben, jedes Perzentil zwischen p0 und p100. Die Anzahl der Stichproben umfasst nur die Befehle, die erfolgreich ausgeführt wurden.	Mikrosekunden

Metrik	Description	Einheit
SuccessfulReadRequestLatency	Latenz erfolgreicher Leseanfragen. Gültige Statistiken: Durchschnitt, Summe, Min, Max, Stichprobenanzahl, jedes Perzentil zwischen p0 und p100. Die Anzahl der Stichproben umfasst nur die Befehle, die erfolgreich ausgeführt wurden.	Mikrosekunden
ErrorCount	Die Gesamtzahl der fehlgeschlagenen Befehle während des angegebenen Zeitraums. Gültige Statistiken: Durchschnitt, Summe, Min., Max	Anzahl
SearchNumberOfIndexes	Anzahl der erstellten Indizes	Anzahl
SearchTotalIndexedDocuments	Gesamtzahl der Schlüssel in allen Indizes	Anzahl
SearchUsedMemoryBytes	Anzahl der in allen Suchdatenstrukturen verbrauchten Speicherbytes	Bytes

CPUUtilization Verfügbarkeit der Engine

AWS Die unten aufgeführten Regionen sind für alle unterstützten Knotentypen verfügbar.

Region	Name der Region
us-east-2	US East (Ohio)
us-east-1	USA Ost (Nord-Virginia)
us-west-1	USA West (Nordkalifornien)
us-west-2	USA West (Oregon)

Region	Name der Region
ap-northeast-1	Asien-Pazifik (Tokio)
ap-northeast-2	Asien-Pazifik (Seoul)
ap-northeast-3	Asien-Pazifik (Osaka)
ap-east-1	Asien-Pazifik (Hongkong)
ap-south-1	Asien-Pazifik (Mumbai)
ap-southeast-1	Asien-Pazifik (Singapur)
ap-southeast-2	Asien-Pazifik (Sydney)
ap-southeast-3	Asien-Pazifik (Jakarta)
ca-central-1	Kanada (Zentral)
cn-north-1	China (Peking)
cn-northwest-2	China (Ningxia)
me-south-1	Naher Osten (Bahrain)
eu-central-1	Europe (Frankfurt)
eu-west-1	Europa (Irland)
eu-west-2	Europa (London)
eu-west-3	EU (Paris)
eu-south-1	Europa (Milan)
af-south-1	Afrika (Kapstadt)
eu-north-1	Europa (Stockholm)
sa-east-1	Südamerika (São Paulo)

Region	Name der Region
us-gov-west-1	AWS GovCloud (US-West)
us-gov-east-1	AWS GovCloud (US-Ost)

Im Folgenden finden Sie Zusammenfassungen bestimmter Befehle, die von `info commandstats` abgeleitet sind. Der Abschnitt „commandstats“ (Befehlsstatistiken) bietet Statistiken auf der Grundlage des Befehlstyps, einschließlich der Anzahl der Aufrufe, des gesamten durch diese Befehle verursachten CPU-Zeitaufwands und des durchschnittlichen CPU-Verbrauchs pro Befehlsausführung. Für jeden Befehlstyp wird die folgende Zeile hinzugefügt: `cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`.

[Die unten aufgeführten Latenzmetriken werden anhand der Commandstats-Statistik von INFO berechnet.](#) Diese werden auf folgende Weise berechnet: $\text{delta}(\text{usec})/\text{delta}(\text{calls})$. `delta` wird als Differenz innerhalb einer Minute berechnet. Die Latenz ist definiert als die CPU-Zeit, die für die Verarbeitung des ElastiCache Befehls benötigt wird. Beachten Sie, dass für Cluster, die Daten-Tiering verwenden, die zum Abrufen von Elementen vom SSD benötigte Zeit in diesen Messungen nicht enthalten ist.

Eine vollständige Liste der verfügbaren Befehle finden Sie in der Valkey-Dokumentation unter [Befehle](#).

Metrik	Description	Einheit
<code>ClusterBasedCmds</code>	Die Gesamtanzahl der Cluster-basierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die auf einen Cluster einwirken (<code>cluster slotcluster info</code> , usw.).	Anzahl
<code>ClusterBasedCmdsLatency</code>	Latenz von Cluster-basierten Befehlen.	Mikrosekunden
<code>EvalBasedCmds</code>	Die Gesamtzahl der Befehle für EVAL-basierte Befehle. Dies wird aus der <code>commandstats</code>	Anzahl

Metrik	Description	Einheit
	Statistik durch Summieren von, abgeleitet. eval evalsha	
EvalBasedCmdsLatency	Latenz von eval-basierten Befehlen.	Mikrosekunden
GeoSpatialBasedCmds	Die Gesamtzahl der Befehle für raumbezogene Befehle. Dies wird aus der commandstats Statistik abgeleitet. Es wird abgeleitet, indem alle Befehle des Geo-Typs summiert werden: geoadd, geodist, geohash, geopos, georadius und georadiusbymember.	Anzahl
GeoSpatialBasedCmdsLatency	Latenz von raumbezogenen Befehlen.	Mikrosekunden
GetTypeCmds	Gesamtanzahl der auf read-only basierenden Befehle. Dies wird aus der commandstats Statistik abgeleitet, indem alle read-only Typbefehle (get,, hget scardrange, usw.) summiert werden.	Anzahl
GetTypeCmdsLatency	Latenz von Lesebefehlen.	Mikrosekunden
HashBasedCmds	Gesamtanzahl der Hash-basierten Befehle. Dies wird aus der commandstats Statistik abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Hashes (hget,, hkeys hvalshdel, usw.) einwirken.	Anzahl
HashBasedCmdsLatency	Latenz von Hash-basierten Befehlen.	Mikrosekunden

Metrik	Description	Einheit
HyperLogLogBasedCmds	Gesamtanzahl der auf HyperLogLog basierenden Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehlstypen (<code>pfadd</code> , <code>pfcountpfmerge</code> , usw.) summiert werden. <code>pf</code>	Anzahl
HyperLogLogBasedCmdsLatency	Latenz von HyperLogLog basierten Befehlen.	Mikrosekunden
JsonBasedCmds	Die Gesamtzahl der JSON-Befehle, einschließlich Lese- und Schreibbefehlen. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle JSON-Befehle summiert werden, die sich auf JSON-Schlüssel beziehen.	Anzahl
JsonBasedCmdsLatency	Die Latenz der JSON-Befehle, einschließlich Lese- und Schreibbefehlen.	Mikrosekunden
JsonBasedGetCmds	Gesamtanzahl der JSON-Schreibschutzbefehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle JSON-Lesebefehle summiert werden, die sich auf JSON-Schlüssel auswirken.	Anzahl
JsonBasedGetCmdsLatency	Latenz der JSON-Schreibschutzbefehle.	Mikrosekunden
JsonBasedSetCmds	Gesamtanzahl der JSON-Schreibbefehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle JSON-Schreibbefehle summiert werden, die sich auf JSON-Schlüssel auswirken.	Anzahl
JsonBasedSetCmdsLatency	Latenz von JSON-Schreibbefehlen.	Mikrosekunden

Metrik	Description	Einheit
KeyBasedCmds	Gesamtanzahl der schlüsselbasierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Schlüssel in mehreren Datenstrukturen (<code>del</code> , <code>expire</code> , <code>rename</code> , usw.) einwirken.	Anzahl
KeyBasedCmdsLatency	Latenz von schlüsselbasierten Befehlen.	Mikrosekunden
ListBasedCmds	Gesamtanzahl der listenbasierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf eine oder mehrere Listen auswirken (<code>lindex</code> , <code>lrange</code> , <code>lpush</code> , <code>ltrim</code> , usw.).	Anzahl
ListBasedCmdsLatency	Latenz von listenbasierten Befehlen.	Mikrosekunden
NonKeyTypeCmds	Gesamtanzahl der nicht schlüsselbasierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die sich nicht auf eine Taste auswirken, z. B., <code>acl</code> , <code>dbsize</code> , <code>info</code> .	Anzahl
NonKeyTypeCmdsLatency	Latenz der Befehle. non-key-based	Mikrosekunden
PubSubBasedCmds	Die Gesamtzahl der Befehle für die pub/sub Funktionalität. Dies wird aus den <code>commandstats</code> Statistiken abgeleitet, indem alle für die pub/sub Funktionalität verwendeten Befehle summiert werden: <code>punsubscribe</code> , <code>publish</code> , <code>pubsub</code> , <code>punsubscribe</code> , <code>ssubscribe</code> , <code>unsubscribe</code> , <code>spublish</code> , <code>subscribe</code> , <code>undunsubscribe</code> .	Anzahl

Metrik	Description	Einheit
PubSubBasedCmdsLatency	Latenz von pub/sub-basierten Befehlen.	Mikrosekunden
SetBasedCmds	Gesamtanzahl der Set-basierten Befehle. Dies wird aus der commandstats Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf einen oder mehrere Sätze auswirken (scard,sdiff, saddsunion, usw.).	Anzahl
SetBasedCmdsLatency	Latenz von Set-basierten Befehlen.	Mikrosekunden
SetTypeCmds	Gesamtanzahl der auf write basierenden Befehle. Dies wird aus der commandstats Statistik abgeleitet, indem alle mutative Befehlstypen summiert werden, die mit Daten arbeiten (set,hset, saddlpop, usw.)	Anzahl
SetTypeCmdsLatency	Latenz von Schreibbefehlen.	Mikrosekunden
SortedSetBasedCmds	Gesamtanzahl der Sorted Set-basierten Befehle. Dies wird aus der commandstats Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf eine oder mehrere sortierte Sätze (zcount,, zrange zrankzadd, usw.) auswirken.	Anzahl
SortedSetBasedCmdsLatency	Latenz von Sortierungs-basierten Befehlen.	Mikrosekunden

Metrik	Description	Einheit
StringBasedCmds	Gesamtanzahl der Zeichenfolge-basierten Befehle. Dies wird aus der commandstats Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf eine oder mehrere Zeichenketten (strlen, setexsetrange, usw.) auswirken.	Anzahl
StringBasedCmdsLatency	Latenz von Zeichenfolgen-basierten Befehlen.	Mikrosekunden
StreamBasedCmds	Die Gesamtanzahl Stream-basierter Befehle. Dies wird aus der commandstats Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf einen oder mehrere Stream-Datentypen (xrange,, xlen xaddxdel, usw.) auswirken.	Anzahl
StreamBasedCmdsLatency	Latenz von Stream-basierten Befehlen.	Mikrosekunden
SearchBasedCmds	Die Gesamtzahl der Suchbefehle, einschließlich Lese- und Schreibbefehlen. Dies wird aus der Commandstats-Statistik abgeleitet, indem alle Suchbefehle summiert werden.	Anzahl
SearchBasedCmdsLatency	Latenz aller Suchbefehle, einschließlich Lese- und Schreibbefehlen.	Mikrosekunden
SearchBasedGetCmds	Die Gesamtzahl der schreibgeschützten Suchbefehle. Dies wird aus der Commandstats-Statistik abgeleitet, indem alle Lesebefehle von Search summiert werden.	Anzahl
SearchBasedGetCmdsLatency	Latenz der schreibgeschützten Search-Befehle.	Mikrosekunden

Metrik	Description	Einheit
SearchBasedSetCmds	Die Gesamtzahl der Such-Schreibbefehle. Dies wird aus der Commandstats-Statistik abgeleitet, indem alle Search-Schreibbefehle summiert werden.	Anzahl
SearchBasedSetCmds Latency	Latenz der Such-Schreibbefehle.	Mikrosekunden

Metriken für Memcached

Der AWS/ElastiCache-Namespace enthält die folgenden Memcache-Metriken.

Der ElastiCache Namespace AWS/enthält die folgenden Metriken, die vom Befehl Memcached stats abgeleitet sind. Jede Metrik wird zu jeder Cache-Knotenebene berechnet.

Informationen finden Sie auch unter:

- [Metriken auf Host-Ebene](#)

Metrik	Beschreibung	Einheit
BytesReadIntoMemcached	Anzahl der Byte, die vom Cache-Knoten aus dem Netzwerk gelesen wurden.	Bytes
BytesUsedForCacheItems	Anzahl der Byte, die zum Speichern von Cache-Elementen verwendet werden.	Bytes
BytesWrittenOutFromMemcached	Anzahl der Byte, die vom Cache-Knoten in das Netzwerk geschrieben wurden.	Bytes
CasBadval	Anzahl der CAS-Anforderungen (check and set), die vom Cache empfangen wurden, bei denen der CAS-Wert nicht mit dem gespeicherten CAS-Wert übereinstimmt.	Anzahl

Metrik	Beschreibung	Einheit
CasHits	Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel gefunden wurde und die CAS-Werte übereinstimmen.	Anzahl
CasMisses	Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
CmdFlush	Anzahl der flush-Befehle, die der Cache erhalten hat.	Anzahl
CmdGet	Die Anzahl der get-Befehle, die der Cache erhalten hat.	Anzahl
CmdSet	Die Anzahl der Set-Befehle, die der Cache empfangen hat.	Anzahl

Metrik	Beschreibung	Einheit
CurrConnections	Zählt die Anzahl der Verbindungen, die zu einem bestimmten Zeitpunkt mit dem Cache verbunden sind. ElastiCache verwendet 2 bis 3 der Verbindungen zur Überwachung des Clusters. Zusätzlich zu den oben genannten erstellt memcached eine Anzahl von internen Verbindungen, die dem Doppelten der für den Knotentyp verwendeten Threads entsprechen. Die Thread-Anzahl für die verschiedenen Knotentypen ist in <code>Nodetype Specific Parameters</code> der entsprechenden Parametergruppe zu sehen. Die Gesamtzahl der Verbindungen ist die Summe der Clientverbindungen, der zu überwachenden Verbindungen und der oben genannten internen Verbindungen.	Anzahl
CurrItems	Die Anzahl der zu einem bestimmten Zeitpunkt mit dem Cache verbundenen Verbindungen.	Anzahl
DecrHits	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
DecrMisses	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
DeleteHits	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl

Metrik	Beschreibung	Einheit
DeleteMisses	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
Evictions	Die Anzahl nicht abgelaufener Elemente, die vom Cache bereinigt wurden, um Platz für neue Schreibvorgänge zu schaffen.	Anzahl
GetHits	Die Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
GetMisses	Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrHits	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrMisses	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
Reclaimed	Die Anzahl abgelaufener Elemente, die vom Cache bereinigt wurden, um Platz für neue Schreibvorgänge zu schaffen.	Anzahl

Für Memcached 1.4.14 stehen die folgenden zusätzlichen Metriken zur Verfügung.

Metrik	Beschreibung	Einheit
BytesUsedForHash	Die Anzahl der Byte, die derzeit von Hash-Tabellen verwendet werden.	Bytes
CmdConfigGet	Die kumulative Anzahl an config get-Anforderungen.	Anzahl
CmdConfigSet	Die kumulative Anzahl an config set-Anforderungen.	Anzahl
CmdTouch	Die kumulative Anzahl an touch-Anforderungen.	Anzahl
CurrConfig	Die aktuelle Anzahl gespeicherter Konfigurationen.	Anzahl
EvictedUnfetched	Die Anzahl der aus dem LRU-Cache (Last Recently Used, "am längsten nicht verwendet") entfernten gültigen Elemente, auf die nie zugegriffen wurde, nachdem sie festgelegt worden waren.	Anzahl
ExpiredUnfetched	Die Anzahl der aus dem LRU-Cache wiedergewonnenen abgelaufenen Elemente, auf die nie zugegriffen wurde, nachdem sie festgelegt worden waren.	Anzahl
SlabsMoved	Die Gesamtanzahl der Slab Pages, die verschoben worden sind.	Anzahl
TouchHits	Die Anzahl der Schlüssel, auf die zugegriffen wurde und die mit einer neuen Ablaufzeit versehen wurden.	Anzahl
TouchMisses	Die Anzahl der Elemente, auf die zwar zugegriffen wurde, die aber nicht gefunden werden konnten.	Anzahl

Der ElastiCache Namespace AWS/enthält die folgenden berechneten Metriken auf Cache-Ebene.

Metrik	Beschreibung	Einheit
NewConnections	Die Anzahl der neuen Verbindungen, die der Cache erhalten hat. Dies wird von der memcached-Statistik total_connections abgeleitet, indem die Änderung in total_connections über einen bestimmten Zeitraum aufgezeichnet wird. Dieser Wert wird immer mindestens 1 sein, da eine Verbindung für a reserviert ist. ElastiCache	Anzahl
NewItems	Die Anzahl der neuen Elemente, die im Cache gespeichert wurden. Dies wird von der memcached-Statistik total_items abgeleitet, indem die Änderung in total_items über einen bestimmten Zeitraum aufgezeichnet wird.	Anzahl
UnusedMemory	Die Größe des nicht von Daten belegten Speichers. Dies wird von den Memcached-Statistiken limit_maxbytes und bytes abgeleitet, indem bytes von limit_maxbytes subtrahiert werden. Da der Memcached-Overhead zusätzlich zu dem für Daten verwendeten Speicher belegt, UnusedMemory sollte dieser Wert nicht als die Menge an Speicher angesehen werden, die für zusätzliche Daten zur Verfügung steht. Möglicherweise treten Bereinigungen auch dann auf, wenn noch etwas ungenutzter Speicherplatz vorhanden ist. Detailliertere Informationen erhalten Sie unter Memcached item memory usage.	Bytes

Welche Metriken sollte ich überwachen?

Die folgenden CloudWatch Kennzahlen bieten einen guten Einblick in die ElastiCache Leistung. In den meisten Fällen empfehlen wir, CloudWatch Alarme für diese Kennzahlen einzurichten, damit Sie Korrekturmaßnahmen ergreifen können, bevor Leistungsprobleme auftreten.

Zu überwachende Metriken

- [CPUUtilization](#)
- [Modul CPUUtilization](#)
- [SwapUsage \(Valkey und Redis OSS\)](#)
- [Evictions](#)
- [CurrConnections](#)
- [Speicher \(Valkey und Redis OSS\)](#)
- [Netzwerk](#)
- [Latenz](#)
- [Replikation](#)
- [Verkehrsmanagement \(Valkey und Redis OSS\)](#)

CPUUtilization

Diese Metrik auf Hostebene wird in Prozent angegeben. Weitere Informationen finden Sie unter [Metriken auf Host-Ebene](#).

Valkey und Redis OSS

Verwenden Sie bei kleineren Knotentypen mit 2 V CPUs oder weniger die CPUUtilization Metrik, um Ihre Arbeitslast zu überwachen.

Generell empfehlen wir, den Schwellenwert auf 90 % der verfügbaren CPU-Kapazität festzulegen. Da Valkey und Redis OSS beide Single-Threading verwenden, sollte der tatsächliche Schwellenwert als Bruchteil der Gesamtkapazität des Knotens berechnet werden. Angenommen, Sie verwenden einen Knotentyp mit zwei Kernen. In diesem Fall CPUUtilization wäre der Schwellenwert für 90/2 oder 45%.

Sie müssen eigene Grenzwerte basierend auf der Anzahl der Kerne im verwendeten Cache-Knoten festlegen. Wenn Sie diesen Schwellenwert überschreiten und Ihre Hauptlast aus Leseanfragen

besteht, skalieren Sie Ihren Cluster, indem Sie Read Replicas hinzufügen. Wenn der Workload hauptsächlich aus Schreibanfragen stammt, empfehlen wir Ihnen abhängig von Ihrer Cluster-Konfiguration:

- Valkey- oder Redis OSS-Cluster (Clustermodus deaktiviert): Skalieren Sie, indem Sie einen größeren Cache-Instance-Typ verwenden.
- Valkey- oder Redis OSS-Cluster (Clustermodus aktiviert): Fügen Sie weitere Shards hinzu, um die Schreiblast auf mehr Primärknoten zu verteilen.

Tip

Anstatt die Metrik auf Host-Ebene zu verwenden `CPUUtilization`, können Valkey- und Redis OSS-Benutzer möglicherweise die Metrik verwenden `EngineCPUUtilization`, die den Prozentsatz der Nutzung auf dem Valkey- oder Redis OSS-Engine-Kern angibt. Um zu sehen, ob diese Metrik auf Ihren Knoten verfügbar ist, und weitere Informationen finden Sie unter [Metriken](#) für Valkey und Redis OSS.

Für größere Knotentypen mit 4 V CPUs oder mehr können Sie die `EngineCPUUtilization` Metrik verwenden, die den Prozentsatz der Nutzung auf dem Valkey- oder Redis OSS-Engine-Kern angibt. Um zu sehen, ob diese Metrik auf Ihren Knoten verfügbar ist, und weitere Informationen finden Sie unter [Metriken für Redis OSS](#).

Memcached

Da Memcached mit mehreren Threads arbeitet, darf diese Metrik bis zu 90 % erreichen. Wenn Sie diesen Schwellenwert überschreiten, skalieren Sie Ihren Cluster, indem Sie einen größeren Cache-Knotentyp verwenden, oder skalieren Sie ihn, indem Sie weitere Cache-Knoten hinzufügen.

Modul CPUUtilization

Für größere Knotentypen mit 4 V CPUs oder mehr möchten Sie möglicherweise die `EngineCPUUtilization` Metrik verwenden, die den Prozentsatz der Nutzung auf dem Kern der Redis OSS-Engine angibt. Um zu sehen, ob diese Metrik auf Ihren Knoten verfügbar ist, und weitere Informationen finden Sie unter [Metriken für Valkey und Redis OSS](#).

Weitere Informationen finden Sie im CPUsAbschnitt [Bewährte Methoden zur Überwachung von Amazon ElastiCache für Redis OSS mithilfe von Amazon CloudWatch](#).

SwapUsage (Valkey und Redis OSS)

Diese Metrik auf Hostebene wird in Bytes angegeben. Weitere Informationen finden Sie unter [Metriken auf Host-Ebene](#).

Liegt die FreeableMemory CloudWatch Metrik nahe 0 (d. h. unter 100 MB) oder ist sie größer als die SwapUsage FreeableMemory Metrik, deutet dies darauf hin, dass ein Knoten unter Speicherauslastung steht. Beachten Sie in diesem Fall folgende Themen

- [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Valkey- oder Redis OSS-Snapshot zu erstellen](#)
- [Verwaltung von reserviertem Speicher für Valkey und Redis OSS](#)

Evictions

Dies ist eine Metrik für die Cache-Engine. Wir empfehlen Ihnen, einen eigenen Grenzwert für diese Metrik basierend auf den Anforderungen Ihrer Anwendung zu bestimmen.

Wenn Sie Memcached verwenden und den von Ihnen gewählten Schwellenwert überschreiten, skalieren Sie Ihren Cluster, indem Sie einen größeren Knotentyp verwenden, oder skalieren Sie, indem Sie weitere Knoten hinzufügen.

CurrConnections

Dies ist eine Metrik für die Cache-Engine. Wir empfehlen Ihnen, einen eigenen Grenzwert für diese Metrik basierend auf den Anforderungen Ihrer Anwendung zu bestimmen.

Eine zunehmende Anzahl von CurrConnections kann auf ein Problem mit Ihrer Anwendung hinweisen. Um dieses Problem zu beheben, müssen Sie das Verhalten der Anwendung untersuchen.

Weitere Informationen finden Sie im Abschnitt Verbindungen unter [Bewährte Methoden zur Überwachung von Amazon ElastiCache für Redis OSS mithilfe von Amazon CloudWatch](#).

Speicher (Valkey und Redis OSS)

Speicher ist ein Kernaspekt von Valkey und Redis OSS. Es ist notwendig, die Speicherauslastung Ihres Clusters zu verstehen, um Datenverluste zu vermeiden und das zukünftige Wachstum Ihres Datasets berücksichtigen zu können. Statistiken über die Speicherauslastung eines Knotens sind im Speicherbereich des [INFO-Befehls](#) verfügbar.

Weitere Informationen finden Sie im Abschnitt Speicher unter [Bewährte Methoden zur Überwachung mit Amazon ElastiCache für Redis OSS mithilfe von Amazon CloudWatch](#).

Netzwerk

Einer der entscheidenden Faktoren für die Kapazität der Netzwerkbandbreite Ihres Clusters ist der von Ihnen ausgewählte Knotentyp. Weitere Informationen zur Netzwerkkapazität Ihres Nodes finden Sie unter [ElastiCache Amazon-Preise](#).

Weitere Informationen finden Sie im Abschnitt Netzwerk unter [Bewährte Methoden zur Überwachung von Amazon ElastiCache für Redis OSS mithilfe von Amazon CloudWatch](#).

Latenz

Die Messung der Antwortzeit für eine ElastiCache for Valkey-Instance kann je nach erforderlicher Granularität auf unterschiedliche Weise erfolgen. Die wichtigsten Phasen, die zur gesamten serverseitigen Reaktionszeit von ElastiCache for Valkey beitragen, sind die Befehlsvorverarbeitung, Befehlsausführung und Befehlsnachverarbeitung.

Befehlsspezifische Latenzmetriken, die aus dem Befehl Valkey [INFO](#) abgeleitet wurden, wie GetTypeCmdsLatency z. B. SetTypeCmdsLatency Metriken, konzentrieren sich speziell auf die Ausführung der zentralen Befehlslogik für den Befehl Valkey. Diese Metriken sind hilfreich, wenn Sie in Ihrem Anwendungsfall die Ausführungszeit von Befehlen oder die aggregierten Latenzen pro Datenstruktur bestimmen möchten.

Die Latenzmetriken SuccessfulWriteRequestLatency und SuccessfulReadRequestLatency messen die Gesamtzeit, die die ElastiCache for Valkey-Engine benötigt, um auf eine Anfrage zu antworten.

Note

Überhöhte Werte für SuccessfulWriteRequestLatency und SuccessfulReadRequestLatency Metriken können auftreten, wenn Valkey-Pipelining verwendet wird und CLIENT REPLY auf dem Valkey-Client aktiviert ist. Valkey-Pipelining ist eine Technik zur Leistungssteigerung, indem mehrere Befehle gleichzeitig ausgegeben werden, ohne auf die Antwort auf jeden einzelnen Befehl warten zu müssen. [Um überhöhte Werte zu vermeiden, empfehlen wir, Ihren Valkey-Client so zu konfigurieren, dass er Befehle weiterleitet, wenn CLIENT REPLY OFF ist.](#)

Weitere Informationen finden Sie im Abschnitt Latenz unter [Bewährte Methoden zur Überwachung mit Amazon ElastiCache mithilfe von Amazon CloudWatch](#).

Replikation

Das Datenvolumen, das repliziert wird, ist über die `ReplicationBytes`-Metrik ersichtbar. Obwohl diese Metrik für die Schreiblast der Replikationsgruppe repräsentativ ist, gibt sie keine Einblicke in den Replikationsstatus. Für diesen Zweck können Sie die `ReplicationLag`-Metrik verwenden.

Weitere Informationen finden Sie im Abschnitt Replikation unter [Bewährte Methoden zur Überwachung von Amazon ElastiCache für Redis OSS mithilfe von Amazon CloudWatch](#).

Verkehrsmanagement (Valkey und Redis OSS)

ElastiCache für Redis OSS verwaltet automatisch den Datenverkehr für einen Knoten, wenn mehr eingehende Befehle an den Knoten gesendet werden, als von Valkey oder Redis OSS verarbeitet werden können. Dies geschieht, um den optimalen Betrieb und die Stabilität der Engine aufrechtzuerhalten.

Wenn der Datenverkehr auf einem Knoten aktiv verwaltet wird, gibt die Metrik `TrafficManagementActive` den Datenpunkt 1 aus. Dies weist darauf hin, dass der Knoten für den bereitgestellten Workload möglicherweise unterdimensioniert ist. Wenn diese Metrik über einen längeren Zeitraum 1 bleibt, evaluieren Sie den Cluster, um zu entscheiden, ob eine Hoch- oder Aufskalierung erforderlich ist.

Sehen Sie sich die Metrik `TrafficManagementActive` auf der Seite [Metriken](#) an, um mehr zu erfahren.

Auswählen von Metrikstatistiken und -zeiträumen

CloudWatch Sie können zwar eine beliebige Statistik und einen beliebigen Zeitraum für jede Metrik auswählen, aber nicht alle Kombinationen sind nützlich. Beispielsweise CPUUtilization sind die Statistiken „Durchschnitt“, „Minimum“ und „Maximum“ nützlich, die Summenstatistik jedoch nicht.

Alle ElastiCache Samples werden für einen Zeitraum von 60 Sekunden für jeden einzelnen Cache-Knoten veröffentlicht. Eine Metrik für einen Cache-Knoten enthält für einen 60-Sekunden-Zeitraum immer nur eine Stichprobe.

Weitere Informationen zum Abrufen von Metriken für Cache-Knoten finden Sie unter [Überwachung von CloudWatch Cluster- und Knotenmetriken](#).

Überwachung von CloudWatch Cluster- und Knotenmetriken

ElastiCache und CloudWatch sind integriert, sodass Sie eine Vielzahl von Metriken sammeln können. Sie können diese Metriken überwachen mit CloudWatch.

Note

Für die folgenden Beispiele sind die CloudWatch Befehlszeilentools erforderlich. Weitere Informationen zu den Entwicklertools CloudWatch und zum Herunterladen finden Sie auf der [CloudWatch Produktseite](#).

Die folgenden Verfahren zeigen Ihnen, wie Sie Speicherplatzstatistiken für einen Cluster für die letzte Stunde sammeln können. CloudWatch

Note

Die Werte für StartTime und EndTime in diesen Beispielen unten dienen nur zur Veranschaulichung. Sie müssen die entsprechenden Werte für den Start- und Endzeitpunkt Ihrer Cache-Knoten einsetzen.

Informationen zu ElastiCache Grenzwerten finden Sie unter [AWS Service Limits](#) für ElastiCache.

Überwachung von CloudWatch Cluster- und Knotenmetriken (Konsole)

So erfassen Sie CPU-Nutzungsstatistiken für einen Cache-Cluster

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie die Cache-Knoten aus, für die Sie Metriken anzeigen möchten.

 Note

Bei der Auswahl von mehr als 20 Knoten wird die Ansicht von Metriken auf der Konsole deaktiviert.

- a. Klicken Sie auf der Seite Cache-Cluster der AWS Management Console auf den Namen eines oder mehrerer Cluster.

Die Detailseite für den Cluster wird angezeigt.

- b. Klicken Sie oben im Fenster auf die Registerkarte Nodes.
- c. Wählen Sie auf der Registerkarte Nodes des Detailfensters die Cache-Knoten aus, für die Sie Metriken anzeigen möchten.

Eine Liste der verfügbaren CloudWatch Metriken wird unten im Konsolenfenster angezeigt.

- d. Klicken Sie auf die Metrik CPU Utilization.

Die CloudWatch Konsole wird geöffnet und zeigt Ihre ausgewählten Metriken an. Sie können die Dropdown-Listenfelder Statistic und Period und die Registerkarte Time Range verwenden, um die angezeigten Metriken zu ändern.

Überwachen von CloudWatch Cluster- und Knotenmetriken mit der CloudWatch CLI

So erfassen Sie CPU-Nutzungsstatistiken für einen Cache-Cluster

- Für Linux, macOS oder Unix:

```
aws cloudwatch get-metric-statistics \
  --namespace AWS/ElastiCache \
  --metric-name CPUUtilization \
  --dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' \
  --statistics=Average \
  --start-time 2018-07-05T00:00:00 \
  --end-time 2018-07-06T00:00:00 \
```

```
--period=3600
```

Für Windows:

```
aws cloudwatch get-metric-statistics ^  
  --namespace AWS/ElastiCache ^  
  --metric-name CPUUtilization ^  
  --dimensions='[{"Name":"CacheClusterId","Value":"test"},  
{"Name":"CacheNodeId","Value":"0001"}]' ^  
  --statistics=Average ^  
  --start-time 2018-07-05T00:00:00 ^  
  --end-time 2018-07-06T00:00:00 ^  
  --period=3600
```

Überwachung von CloudWatch Cluster- und Knotenmetriken mithilfe der CloudWatch API

So erfassen Sie CPU-Nutzungsstatistiken für einen Cache-Cluster

- Rufen Sie die CloudWatch API `GetMetricStatistics` mit den folgenden Parametern auf (beachten Sie, dass die Start- und Endzeiten nur als Beispiele angezeigt werden; Sie müssen Ihre eigenen entsprechenden Start- und Endzeiten ersetzen):
 - `Statistics.member.1=Average`
 - `Namespace=AWS/ElastiCache`
 - `StartTime=2013-07-05T00:00:00`
 - `EndTime=2013-07-06T00:00:00`
 - `Period=60`
 - `MeasureName=CPUUtilization`
 - `Dimensions=CacheClusterId=mycacheclass,CacheNodeId=0002`

Example

```
http://monitoring.amazonaws.com/  
  ?Action=GetMetricStatistics  
  &SignatureVersion=4
```

```
&Version=2014-12-01
&StartTime=2018-07-05T00:00:00
&EndTime=2018-07-06T23:59:00
&Period=3600
&Statistics.member.1=Average
&Dimensions.member.1="CacheClusterId=mycachecluster"
&Dimensions.member.2="CacheNodeId=0002"
&Namespace=&AWS;/ElastiCache
&MeasureName=CPUUtilization
&Timestamp=2018-07-07T17%3A48%3A21.746Z
&AWS;AccessKeyId=<&AWS; Access Key ID>
&Signature=<Signature>
```

Kontingente für ElastiCache

Ihr AWS Konto verfügt über Standardkontingente, die früher als Limits bezeichnet wurden, für jeden AWS Dienst. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region. Sie können Erhöhungen für einige Kontingente beantragen und andere Kontingente können nicht erhöht werden.

Um die Kontingente für anzuzeigen ElastiCache, öffnen Sie die [Konsole Service Quotas](#). Wählen Sie im Navigationsbereich AWS -Services und dann ElastiCache aus.

Informationen zur Erhöhung eines Kontingents finden Sie unter [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch zu Service Quotas. Wenn das Kontingent unter Service Quotas noch nicht in verfügbar ist, verwenden Sie das [Formular zur Erhöhung des Service-Limits](#).

Ihr AWS Konto hat die folgenden Kontingente für ElastiCache.

Name	Standardwert	Beschreibung	Metrikname
Knoten pro Region	300	Maximale Anzahl Knoten in allen Clustern in einer Region. Dieses Kontingent gilt für Ihre nicht reservierten Knoten in der angegebenen Region. Sie können bis zu 300 nicht reservierte Knoten in derselben Region haben.	NodesPerRegion
Knoten pro Cluster (Memcached)	60	Maximale Anzahl Knoten in einem einzelnen Memcached-Cluster	Memcached NodesPerCluster

Name	Standardwert	Beschreibung	Metrikname
Knoten pro Cluster (Clustermodus aktiviert)	90	Die maximale Anzahl von Knoten in einem einzelnen Redis OSS- oder Valkey-Cluster.	NodesPerCluster
Parametergruppen pro Region	300	Die maximale Anzahl von Parameter gruppen, die Sie in einer Region erstellen können.	ParameterGroup
Subnetzgruppen pro Region	300	Die maximale Anzahl von Subnetzgruppen, die Sie in einer Region erstellen können.	SubnetGroup
Subnetze pro Subnetzgruppe	20	Maximale Anzahl Subnetze, die Sie für eine Subnetzgruppe definieren können	SubnetsPerSubnetGroup
Serverlose Caches pro Region	40	Die maximale Anzahl serverloser Caches in einer Region.	ServerlessCache
Serverless-Snapshots pro Tag und Cache	24	Die maximale Anzahl serverloser Snapshots , die Sie pro Tag für jeden serverlos en Cache erstellen können.	SnapshotsPerDayPer Cache

Name	Standardwert	Beschreibung	Metrikname
Benutzer pro Region	2000	Die maximale Gesamtzahl von Benutzern, die Sie in einer Region erstellen können.	Benutzer
Benutzergruppen pro Region	200	Die maximale Anzahl von Benutzergruppen, die Sie in einer Region erstellen können.	UserGroup
Benutzer pro Benutzergruppe	100	Die maximale Anzahl von Benutzern, die Sie für eine Benutzergruppe definieren können.	UsersPerUserGroup

Referenz

Die Themen in diesem Abschnitt behandeln die Arbeit mit der ElastiCache Amazon-API und den ElastiCache Abschnitt der AWS CLI. In diesem Abschnitt sind auch allgemeine Fehlermeldungen und Servicemeldungen enthalten.

- [Die ElastiCache API verwenden](#)
- [ElastiCache API Reference](#)
- [ElastiCache Abschnitt der AWS CLI Referenz](#)
- [ElastiCache Amazon-Fehlermeldungen](#)
- [Benachrichtigungen](#)

Die ElastiCache API verwenden

Dieser Abschnitt enthält aufgabenorientierte Beschreibungen der Verwendung und Implementierung ElastiCache von Vorgängen. Eine vollständige Beschreibung dieser Vorgänge finden Sie in der [Amazon ElastiCache API-Referenz](#).

Themen

- [Verwenden der Abfrage-API](#)
- [Verfügbare Bibliotheken](#)
- [Fehlerbehebung bei Anwendungen](#)

Verwenden der Abfrage-API

Abfrageparameter

HTTP-Query-basierte Anfragen sind HTTP-Anfragen, die das HTTP-Verb GET oder POST und einen Query-Parameter namens `Action` verwenden.

Jede Query-Anfrage muss einige allgemeine Parameter enthalten, um die Authentifizierung und Auswahl einer Aktion zu bearbeiten.

Einige Operationen verwenden Parameterlisten. Diese Listen werden mit der Notation `param.n` definiert. Werte von *n* sind ganze Zahlen, die bei 1 beginnen.

Authentifizierung von Abfrageanforderungen

Sie können Abfrageanforderungen nur über HTTPS senden und müssen in jede Abfrageanforderung eine Signatur einschließen. In diesem Abschnitt wird beschrieben, wie Sie die Signatur erstellen. Die in den folgenden Schritten beschriebene Methode wird als Signaturversion 4 bezeichnet.

Die folgenden grundlegenden Schritte dienen der Authentifizierung von Anfragen an AWS. Dies setzt voraus, dass Sie registriert sind AWS und über eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel verfügen.

Abfrageauthentifizierungsprozess

1. Der Absender erstellt eine Anfrage an AWS.
2. Der Sender berechnet die Anforderungssignatur, ein Keyed-Hashing for Hash-based Message Authentication Code (HMAC) mit einer SHA-1-Hash-Funktion, wie im nächsten Abschnitt dieses Themas beschrieben.
3. Der Absender der Anfrage sendet die Anforderungsdaten, die Signatur und die Access Key ID (die Schlüssel-ID des verwendeten Secret Access Keys) an. AWS
4. AWS verwendet die Zugriffsschlüssel-ID, um den geheimen Zugriffsschlüssel nachzuschlagen.
5. AWS generiert eine Signatur aus den Anforderungsdaten und dem geheimen Zugriffsschlüssel unter Verwendung desselben Algorithmus, der zur Berechnung der Signatur in der Anfrage verwendet wurde.
6. Wenn die Signaturen übereinstimmen, wird die Anforderung als authentisch betrachtet. Falls der Vergleich fehlschlägt, wird die Anforderung verworfen, und AWS gibt eine Fehlerantwort zurück.

Note

Wenn eine Anforderung einen `Timestamp`-Parameter enthält, läuft die für die Anforderung berechnete Signatur 15 Minuten nach dessen Wert ab.

Wenn eine Anforderung einen `Expires`-Parameter enthält, läuft die Signatur an dem durch den `Expires`-Parameter angegebenen Zeitpunkt ab.

So wird die Anfragesignatur berechnet

1. Erstellen Sie eine vereinheitlichte Abfragezeichenfolge, die Sie später bei dieser Prozedur benötigen:

- a. Sortieren Sie die UTF-8-Abfragezeichenkomponenten nach Parameternamen in natürlicher Bytereihenfolge. Die Parameter können aus der GET-URI oder aus dem POST-Hauptteil stammen (wenn Content-Type auf x-www-form-urlencoded application/ gesetzt ist).
 - b. Führen Sie eine URL-Codierung des Parameternamens und -werts nach folgenden Regeln durch:
 - i. Führen Sie keine URL-Codierung der nicht reservierten, von RFC 3986 definierten Zeichen durch. Folgende Zeichen sind nicht reserviert: A – Z, a – z, 0 – 9, Bindestrich (-), Unterstrich (_), Punkt (.)
 - ii. Versehen Sie alle anderen Zeichen mit Prozentcode (%XY), wobei X und Y für Hexadezimalzeichen, d. h. 0-9 und die Großbuchstaben A-F, steht.
 - iii. Versehen Sie alle erweiterten UTF-8-Zeichen mit Prozentcode im Format %XY%ZA....
 - iv. Versehen Sie das Leerzeichen mit dem Prozentcode %20 (und nicht + wie in herkömmlichen Codierungsschemata).
 - c. Trennen Sie die codierten Parameternamen mit dem Gleichzeichen (=) (ASCII-Zeichen 61) von den zugehörigen codierten Werten, auch wenn der Parameterwert leer ist.
 - d. Trennen Sie diese Namen-Wert-Paare durch ein kaufmännisches Und (&) (ASCII-Code 38).
2. Erstellen Sie die Zeichenfolge für die Signatur anhand der folgenden Pseudogrammatik (das "\n" steht in ASCII für eine neue Zeile).

```
StringToSign = HTTPVerb + "\n" +  
ValueOfHostHeaderInLowercase + "\n" +  
HTTPRequestURI + "\n" +  
CanonicalizedQueryString <from the preceding step>
```

Die HTTPRequest URI-Komponente ist die absolute HTTP-Pfadkomponente des URI bis zur Abfragezeichenfolge, aber nicht einschließlich, der Abfragezeichenfolge. Wenn der HTTPRequest URI leer ist, verwenden Sie einen Schrägstrich (/).

3. Berechnen Sie einen RFC 2104-konformen HMAC mit der gerade erstellten Zeichenfolge, Ihrem geheimen Zugriffsschlüssel als Schlüssel und/oder als Hash-Algorithmus SHA256 . SHA1

Weitere Informationen finden Sie unter <https://www.ietf.org/rfc/rfc2104.txt>.

4. Konvertieren Sie den resultierenden Wert zu base64.
5. Schließen Sie den Wert als Wert des Signature-Parameters in die Anforderung ein.

Das folgende Beispiel zeigt eine Anforderung (die Zeilenumbrüche wurden aus Gründen der Übersichtlichkeit hinzugefügt).

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterIdentifier=myCacheCluster  
  &SignatureMethod=HmacSHA256  
  &SignatureVersion=4  
  &Version=2014-12-01
```

Für die vorangehende Abfragezeichenfolge berechnen Sie die HMAC-Signatur über die folgende Zeichenfolge.

```
GET\n  
  elasticache.amazonaws.com\n  
  Action=DescribeCacheClusters  
  &CacheClusterIdentifier=myCacheCluster  
  &SignatureMethod=HmacSHA256  
  &SignatureVersion=4  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Credential=AKIADQKE4SARGYLE%2F20140523%2Fus-west-2%2Felasticache  
%2Faws4_request  
  &X-Amz-Date=20141201T223649Z  
  &X-Amz-SignedHeaders=content-type%3Bhost%3Buser-agent%3Bx-amz-content-sha256%3Bx-  
amz-date  
    content-type:  
    host:elasticache.us-west-2.amazonaws.com  
    user-agent:CacheServicesAPICommand_Client  
  x-amz-content-sha256:  
  x-amz-date:
```

Das Ergebnis ist die folgende signierte Anforderung.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterIdentifier=myCacheCluster  
  &SignatureMethod=HmacSHA256  
  &SignatureVersion=4  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
```

```
&X-Amz-Credential=AKIADQKE4SARGYLE/20141201/us-west-2/elasticache/aws4_request
&X-Amz-Date=20141201T223649Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=2877960fced9040b41b4feaca835fd5cfeb9264f768e6a0236c9143f915ffa56
```

Ausführliche Informationen über das Signieren und Berechnen der Anforderungssignatur finden Sie im Thema [Signaturprozess mit Signature Version 4](#) und in den entsprechenden Unterthemen.

Verfügbare Bibliotheken

AWS bietet Software Development Kits (SDKs) für Softwareentwickler, die es vorziehen, Anwendungen mit sprachspezifischen APIs statt mit der Query API zu erstellen. Diese SDKs bieten grundlegende Funktionen (nicht im Lieferumfang enthalten APIs) wie Anforderungsauthentifizierung, Wiederholungsversuche von Anfragen und Fehlerbehandlung, sodass der Einstieg erleichtert wird. SDKs und zusätzliche Ressourcen sind für die folgenden Programmiersprachen verfügbar:

- [Java](#)
- [Windows und .NET](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

Weitere Informationen zu anderen Sprachen finden Sie unter [Beispiel-Code und Bibliotheken](#).

Fehlerbehebung bei Anwendungen

ElastiCache bietet spezifische und beschreibende Fehler, um Ihnen bei der Behebung von Problemen bei der Interaktion mit der ElastiCache API zu helfen.

Fehler bei Abrufen

In der Regel sollte Ihre Anwendung überprüfen, ob eine Anforderung einen Fehler verursacht hat, bevor Sie Zeit für die Verarbeitung von Ergebnissen aufwenden. Der einfachste Weg, um herauszufinden, ob ein Fehler aufgetreten ist, besteht darin, in der Antwort der ElastiCache API nach einem `Error` Knoten zu suchen.

XPath Die Syntax bietet eine einfache Möglichkeit, nach dem Vorhandensein eines `Error` Knotens zu suchen und den Fehlercode und die Fehlermeldung abzurufen. Der folgende Codeausschnitt

verwendet Perl und das XPath Modul XML::, um festzustellen, ob bei einer Anfrage ein Fehler aufgetreten ist. Wenn ein Fehler aufgetreten ist, gibt der Code den ersten Fehlercode und die erste Fehlermeldung in der Antwort an.

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ( $xp->find("//Error") )
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Tipps zur Problembehebung

Die folgenden Prozesse werden empfohlen, um Probleme mit der ElastiCache -API zu diagnostizieren und zu beheben.

- Stellen Sie sicher, dass es korrekt ElastiCache läuft.

Öffnen Sie dazu einfach ein Browserfenster und senden Sie eine Abfrageanfrage an den ElastiCache Dienst (z. B. <https://elasticache.amazonaws.com>). Ein interner Serverfehler MissingAuthenticationTokenException oder 500 bestätigt, dass der Dienst verfügbar ist und auf Anfragen reagiert.

- Überprüfen Sie die Struktur Ihrer Anforderung.

Für jeden ElastiCache Vorgang gibt es eine Referenzseite in der ElastiCache API-Referenz. Prüfen Sie nochmals, dass Sie die Parameter korrekt verwenden. Die Beispielanforderungen oder Benutzerszenarien zeigen Ihnen, ob ähnliche Operationen ausgeführt werden, und vermitteln Ihnen eine Vorstellung von möglichen Fehlern.

- Sehen Sie im Forum nach.

ElastiCache hat ein Diskussionsforum, in dem Sie nach Lösungen für Probleme suchen können, auf die andere unterwegs gestoßen sind. Weitere Informationen zur Anzeige des Forums finden Sie unter

<https://forums.aws.amazon.com/> .

Einrichtung der ElastiCache Befehlszeilenschnittstelle

In diesem Abschnitt wird beschrieben, welche Voraussetzungen für die Ausführung der Befehlszeilen-Tools erforderlich sind, wo Sie die Befehlszeilen-Tools herunterladen können und wie die Tools und die Umgebung eingerichtet werden. Zudem werden gängige Beispiele für die Nutzung der Tools vorgestellt.

Folgen Sie den Anweisungen in diesem Thema nur, wenn Sie AWS CLI zum Forum wechseln ElastiCache.

Important

Die Amazon ElastiCache Command Line Interface (CLI) unterstützt keine ElastiCache Verbesserungen nach API-Version 2014-09-30. Um neuere ElastiCache Funktionen von der Befehlszeile aus zu verwenden, verwenden Sie die [AWS Befehlszeilenschnittstelle](#).

Themen

- [Voraussetzungen](#)
- [Aufrufen der Befehlszeilen-Tools](#)
- [Einrichten der Tools](#)
- [Bereitstellung von Anmeldeinformationen für die Tools](#)
- [Umgebungsvariablen](#)

Voraussetzungen

In diesem Dokument wird davon ausgegangen, dass Sie in einer Linux/UNIX oder einer Windows-Umgebung arbeiten können. Die ElastiCache Amazon-Befehlszeilentools funktionieren auch unter Mac OS X, einer UNIX-basierten Umgebung. In diesem Handbuch sind jedoch keine spezifischen Mac OS X-Anweisungen enthalten.

Als Konvention wird jeglichem Befehlszeilentext die generische Eingabeaufforderung **PROMPT>** vorangestellt. Der auf dem Rechner tatsächlich angezeigte Prompt sieht wahrscheinlich anders aus. Wir verwenden **\$** sie auch, um einen Linux/UNIX bestimmten Befehl und **C:\>** für einen Windows-spezifischen Befehl anzugeben. Die Beispielausgabe des Befehls wird direkt im Anschluss ohne Präfix angezeigt.

Die Java-Laufzeitumgebung

Für die in diesem Handbuch verwendeten Befehlszeilen-Tools muss Java Version 5 oder höher ausgeführt werden. Eine JRE- oder JDK-Installation ist zulässig. Informationen zum Ansehen und Herunterladen JREs für eine Reihe von Plattformen, einschließlich Linux/UNIX Windows, finden Sie unter [Java SE-Downloads](#).

Einrichten der Java-Home-Variablen

Die Befehlszeilen-Tools sind von einer Umgebungsvariablen (JAVA_HOME) abhängig, um die Java-Runtime zu finden. Diese Umgebungsvariable sollte auf den vollständigen Pfad des Verzeichnisses gesetzt werden, das ein Unterverzeichnis namens bin enthält, das wiederum die ausführbare java-Datei (unter Linux und UNIX) oder die ausführbare java.exe-Datei (unter Windows) enthält.

Einrichten der Java Home-Variablen

1. Einrichten der Java Home-Variablen.

- Geben Sie unter Linux und UNIX den folgenden Befehl ein:

```
$ export JAVA_HOME=<PATH>
```

- Geben Sie unter Windows den folgenden Befehl ein:

```
C:\> set JAVA_HOME=<PATH>
```

2. Bestätigen Sie die Pfadeinrichtung, indem Sie **\$JAVA_HOME/bin/java -version** ausführen und die Ausgabe überprüfen.

- Die Ausgabe unter Linux/UNIX entspricht weitgehend der Folgenden:

```
$ $JAVA_HOME/bin/java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

- Unter Windows sehen Sie eine Ausgabe vergleichbar der Folgenden:

```
C:\> %JAVA_HOME%\bin\java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

Aufrufen der Befehlszeilen-Tools

Die Befehlszeilen-Tools stehen als ZIP-Datei auf der Website [ElastiCache -Entwickler-Tools](#) zur Verfügung. Diese Tools sind in Java geschrieben und enthalten Shell-Skripts für Windows 2000/XP/Vista/Windows 7, Linux/UNIX und Mac OSX. Die ZIP-Datei ist unabhängig und erfordert keine Installation; laden Sie einfach die ZIP-Datei herunter und entpacken Sie sie in ein Verzeichnis auf Ihrem lokalen Rechner.

Einrichten der Tools

Die Befehlszeilen-Tools sind von einer Umgebungsvariablen (AWS_ELASTICACHE_HOME) abhängig, um unterstützende Bibliotheken zu finden. Sie müssen diese Umgebungsvariable festlegen, bevor Sie die Tools verwenden können. Setzen Sie sie auf den Pfad des Verzeichnisses, in das Sie die Befehlszeilen-Tools entpackt haben. Dieses Verzeichnis hat den Namen ElastiCacheCli-A.b.nnnn (A, B und n sind Versions-/Versionsnummern) und enthält Unterverzeichnisse mit den Namen bin und lib.

So AWS legen Sie die Umgebungsvariable _ELASTICACHE_HOME fest

- Öffnen Sie ein Befehlszeilenfenster und geben Sie einen der folgenden Befehle ein, um die Umgebungsvariable _ELASTICACHE_HOME festzulegen. AWS
 - Geben Sie unter Linux und UNIX den folgenden Befehl ein:

```
$ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

- Geben Sie unter Windows den folgenden Befehl ein:

```
C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

Um die Verwendung der Tools zu vereinfachen, empfehlen wir Ihnen, das BIN-Verzeichnis der Tools Ihrem System-Pfad hinzuzufügen. Der Rest dieser Anleitung geht davon aus, dass sich das BIN-Verzeichnis in Ihrem Systempfad befindet.

Das BIN-Verzeichnis der Tools Ihrem Systempfad hinzufügen

- Geben Sie die folgenden Befehle ein, um das BIN-Verzeichnis der Tools zu Ihrem System-PATH hinzuzufügen.
- Geben Sie unter Linux und UNIX den folgenden Befehl ein:

```
$ export PATH=$PATH:$&AWS;_ELASTICACHE_HOME/bin
```

- Geben Sie unter Windows den folgenden Befehl ein:

```
C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
```

Note

Die Windows-Umgebungsvariablen werden zurückgesetzt, wenn Sie das Befehlsfenster schließen. Möglicherweise wollen Sie sie permanent einrichten. Weitere Informationen finden Sie in der Dokumentation zu Ihrer Windows-Version.

Note

Pfade, die ein Leerzeichen enthalten, müssen in Anführungszeichen gesetzt werden, z. B.: "C:\Program Files\Java"

Bereitstellung von Anmeldeinformationen für die Tools

Die Befehlszeilentools benötigen den AWS Access Key und den Secret Access Key, die Sie mit Ihrem Konto erhalten haben. AWS Sie können sie über die Befehlszeile oder aus einer Anmeldeinformationsdatei auf Ihrem lokalen System beziehen.

Die Bereitstellung umfasst eine Vorlagendatei `$ {AWS_ELASTICACHE_HOME}/credential-file-path.template`, die Sie mit Ihren Informationen bearbeiten müssen. Nachfolgend finden Sie dem Inhalt der Template-Datei:

```
AWS AccessKeyId=<Write your AWS access ID>
```

```
AWS SecretKey=<Write your AWS secret key>
```

Important

Unter UNIX sollten Sie die Berechtigungen auf den Eigentümer der Anmeldeinformationsdatei beschränken:

```
$ chmod 600 <the file created above>
```

Bei der Einrichtung der Anmeldeinformationsdatei müssen Sie die Umgebungsvariable `AWS_CREDENTIAL_FILE` festlegen, damit die Tools Ihre Informationen finden können. ElastiCache

Um die Umgebungsvariable `_CREDENTIAL_FILE` festzulegen AWS

1. Legen Sie die -Umgebungsvariable fest:

- Aktualisieren Sie unter Linux und UNIX die Variable mit folgendem Befehl:

```
$ export &AWS;_CREDENTIAL_FILE=<the file created above>
```

- Legen Sie unter Windows die Variable mit folgendem Befehl fest:

```
C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
```

2. Überprüfen Sie, ob Ihre Einrichtung ordnungsgemäß funktioniert, und führen Sie den folgenden Befehl aus:

```
elasticache --help
```

Sie sollten die Verwendungsseite für alle ElastiCache-Befehle sehen.

Umgebungsvariablen

Umgebungsvariablen können nützlich sein, um Skripts zu erstellen, Standardwerte zu konfigurieren oder sie vorübergehend zu überschreiben.

Zusätzlich zur Umgebungsvariablen `AWS_CREDENTIAL_FILE` unterstützen die meisten API-Tools, die in der Befehlszeilenschnittstelle enthalten sind, die ElastiCache folgenden Variablen:

- `EC2_REGION` — Die zu verwendende Region. AWS
- `AWS_ELASTICACHE_URL` – Die für den Serviceaufruf zu verwendende URL. Es ist nicht erforderlich, einen anderen regionalen Endpunkt anzugeben, wenn `EC2_REGION` angegeben oder der Parameter `--region` übergeben wird.

Die folgenden Beispiele zeigen, wie die Umgebungsvariable `EC2_REGION` gesetzt wird, um die von den API-Tools verwendete Region zu konfigurieren:

Linux, OS X oder Unix

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

ElastiCache Amazon-Fehlermeldungen

Die folgenden Fehlermeldungen werden von Amazon zurückgegeben ElastiCache. Möglicherweise erhalten Sie weitere Fehlermeldungen, die von anderen AWS Diensten oder von ElastiCache Valkey, Memcached oder Redis OSS zurückgegeben werden. Beschreibungen von Fehlermeldungen aus anderen Quellen als finden Sie in der Dokumentation der Quelle ElastiCache, die die Fehlermeldung generiert hat.

- [Cluster node quota exceeded](#)
- [Customer's node quota exceeded](#)
- [Manual snapshot quota exceeded](#)
- [Insufficient cluster capacity](#)

Fehlermeldung: Das Cluster-Knoten-Kontingent ist erschöpft. Jeder Cluster in dieser Region kann maximal %n Knoten enthalten.

Ursache: Sie haben versucht, einen Cluster zu erstellen oder zu ändern, der dadurch mehr als %n Knoten enthalten würde.

Lösung: Ändern Sie die Anforderung dahingehend, dass der Cluster nicht mehr als %n Knoten enthält. Oder, wenn Sie mehr als %n Knoten benötigen, stellen Sie Ihre Anfrage über das [Amazon ElastiCache Node-Anfrageformular](#).

Weitere Informationen finden Sie unter [ElastiCache Amazon-Limits](#) unter Allgemeine Amazon Web Services-Referenz.

Fehlermeldungen: Kundenknotenquote überschritten. Sie können maximal %n Knoten in dieser Region haben. Oder: Sie haben Ihr Kontingent an %s-Knoten in dieser Region bereits erreicht.

Ursache: Sie haben versucht, einen Cluster zu erstellen oder zu ändern, und Ihr Konto würde dadurch mehr als %n Knoten in allen Clustern in dieser Region enthalten.

Lösung: Ändern Sie die Anforderung dahingehend, dass die Gesamtanzahl der Knoten in der Region für alle Cluster in diesem Konto nicht mehr als &n beträgt. Oder, wenn Sie mehr als %n Knoten benötigen, stellen Sie Ihre Anfrage über das [Amazon ElastiCache Node-Anfrageformular](#).

Weitere Informationen finden Sie unter [ElastiCache Amazon-Limits](#) unter Allgemeine Amazon Web Services-Referenz.

Fehlermeldungen: Die maximale Anzahl an manuellen Snapshots für dieses Cluster innerhalb von 24 Stunden wurde erreicht. oder Die maximale Anzahl an manuellen Snapshots für diesen Konten innerhalb von 24 Stunden für das Kontingent von %n wurde erreicht.

Ursache: Sie haben versucht einen manuellen Snapshot eines Clusters zu erstellen, während die maximale Anzahl an manuellen Snapshots für einen 24-Stunden-Zeitraum bereits erreicht war.

Lösung: Warten Sie 24 Stunden, bevor Sie erneut versuchen, einen manuellen Snapshot des Clusters zu erstellen. Wenn Sie den manuellen Snapshot direkt benötigen, erstellen Sie den Snapshot für einen anderen Konten mit denselben Daten, z. B. für einen anderen Knoten in diesem Cluster.

Fehlermeldungen: InsufficientCacheClusterCapacity

Ursache: Die verfügbare On-Demand-Kapazität von AWS reicht derzeit nicht aus, um Ihre Anfrage zu bearbeiten.

Solution (Lösung):

- Warten Sie einige Minuten und senden Sie Ihre Anfrage erneut. Die Kapazität kann häufig schwanken.
- Senden Sie eine neue Anfrage mit einer geringeren Anzahl von Knoten oder Shards (Knotengruppen). Wenn Sie z. B. eine einzelne Anfrage zum Starten von 15 Knoten senden möchten, versuchen Sie stattdessen, 3 Anfragen für 5 Knoten oder 15 Anfragen für 1 Knoten zu erstellen.
- Wenn Sie einen Cluster starten, senden Sie eine neue Anfrage ohne Angabe einer Availability Zone.
- Wenn Sie einen Cluster starten, senden Sie eine neue Anfrage unter Verwendung eines anderen Knotentyps (die Skalierung können Sie später erweitern). Weitere Informationen finden Sie unter [Skalierung ElastiCache](#).

Benachrichtigungen

Dieses Thema behandelt ElastiCache Benachrichtigungen, die Sie interessieren könnten. Eine Benachrichtigung ist eine Situation oder ein Ereignis, die bzw. das in der Regel temporär ist und nur solange andauert, bis eine Lösung gefunden und implementiert wurde. Benachrichtigungen haben normalerweise ein Anfangsdatum und ein Lösungsdatum, nach dem die Benachrichtigung nicht mehr relevant ist. Sie entscheiden selbst, welche Benachrichtigungen für Sie relevant sind. Wir empfehlen eine Implementierungsrichtlinie, durch die sich die Leistung Ihres Clusters verbessern lässt.

Benachrichtigungen kündigen keine neuen oder verbesserten ElastiCache Features oder Funktionen an.

Allgemeine ElastiCache Benachrichtigungen

Derzeit gibt es keine ausstehenden ElastiCache Benachrichtigungen, die nicht maschinenspezifisch sind.

ElastiCache für Memcached-Benachrichtigungen

Die folgenden ElastiCache Benachrichtigungen sind spezifisch für die Memcached-Engine.

ElastiCache für Memcached-spezifische Benachrichtigungen

- [Warnung: Memcached-LRU-Crawler verursacht Segmentierungsfehler](#)

Warnung: Memcached-LRU-Crawler verursacht Segmentierungsfehler

 Datum der Warnung: 28. Februar 2017

Unter bestimmten Umständen kann ein Cluster durch einen Segmentierungsfehler im Memcached-LRU-Crawler instabil werden. Dieses Problem tritt bereits seit einiger Zeit in der Memcached-Engine auf. Es trat erstmals in Memcached 1.4.33 auf, wenn der LRU-Crawler standardmäßig aktiviert wurde.

Wenn dieses Problem bei Ihnen auftritt, empfehlen wir, den LRU-Crawler zu deaktivieren, bis das Problem behoben wurde. Führen Sie dazu den Befehl `lru_crawler disable` auf der Befehlszeile aus oder ändern Sie den Parameterwert `lru_crawler` (empfohlen).

Lösungsdatum:

Auflösung

ElastiCache für Redis OSS-spezifische Benachrichtigungen

Derzeit gibt es keine ausstehenden Benachrichtigungen ElastiCache für Redis OSS.

ElastiCache Geschichte der Dokumentation

- API-Version: 2015-02-02
- Letzte Aktualisierung der Dokumentation: 27. November 2023

In der folgenden Tabelle werden wichtige Änderungen in den einzelnen Versionen des ElastiCache Benutzerhandbuchs nach März 2018 beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie den RSS-Feed abonnieren.

Aktuelle ElastiCache Updates

Änderung	Beschreibung	Datum
Support ElastiCache für vertikale Skalierung auf Abruf von Memcached und automatische horizontale Skalierung	ElastiCache mit Memcached unterstützt jetzt vertikale Skalierung auf Abruf und automatisierte horizontale Skalierung. Dies beinhaltet eine Aktualisierung der Richtlinien für die Aktion <code>elasticache:ModifyCacheCluster</code>	10. April 2025
Support für ElastiCache mit Valkey	ElastiCache unterstützt jetzt Valkey. Valkey 7.2.6 ist mit Redis OSS 7.2 kompatibel. Weitere Informationen finden Sie unter Valkey.	8. Oktober 2024
Größe: flexible reservierte Knoten	ElastiCache unterstützt jetzt Size flexible reservierte Knoten . Weitere Informationen finden Sie unter ElastiCache Amazon-Preise .	1. Oktober 2024
Support ElastiCache für Memcached 1.6.22	ElastiCache für Memcached unterstützt jetzt Memcached	10. Januar 2024

1.6.22. Es enthält kumulative Fehlerkorrekturen der Version [1.6.18](#). Weitere Informationen finden Sie unter [Memcached Version 1.6.22](#).

[ElastiCache für Redis OSS wurde Unterstützung für zusätzliche C7gn-Knot engrößen hinzugefügt](#)

ElastiCache für Redis OSS wurde Unterstützung für zusätzliche C7gn-Knot engrößen hinzugefügt.

10. Januar 2024

[ElastiCache für Redis
unterstützt OSS jetzt die
Erstellung von serverlosen
Caches](#)

8. November 2023

Sie können jetzt Serverless-Caches erstellen, die das Cache-Management vereinfachen und sofort skalieren, um die anspruchsvollsten Anwendungen zu unterstützen. Weitere Informationen finden Sie unter [Auswählen aus den Bereitstellungsoptionen](#). Im Rahmen dieser Funktion wurden [neue Berechtigungen](#) zu `ElastiCacheServiceRolePolicy` und `AmazonElastiCacheFullAccess` hinzugefügt, um die Zuordnung von Serverless-Caches zu verwalteten VPC-Endpunkten zu ermöglichen. Darüber hinaus wurden Berechtigungen hinzugefügt, um ein überarbeitetes Konsolenerlebnis mithilfe der Richtlinie `AmazonElastiCacheFullAccess` zu unterstützen.

[ElastiCache for Memcached unterstützt jetzt die Erstellung von serverlosen Caches](#)

Sie können jetzt Serverless-Caches erstellen, die das Cache-Management vereinfachen und sofort skalieren, um die anspruchsvollsten Anwendungen zu unterstützen. Weitere Informationen finden Sie unter [Auswählen aus den Bereitstellungsoptionen](#). Im Rahmen dieser Funktion wurden [neue Berechtigungen](#) zu `ElastiCacheServiceRolePolicy` und `AmazonElastiCacheFullAccess` hinzugefügt, um die Zuordnung von Serverless-Caches zu verwalteten VPC-Endpunkten zu ermöglichen. Darüber hinaus wurden Berechtigungen hinzugefügt, um ein überarbeitetes Konsolenerlebnis mithilfe der Richtlinie `AmazonElastiCacheFullAccess` zu unterstützen.

8. November 2023

[ElastiCache für Redis unterstützt OSS jetzt das Ändern des Clustermodus](#)

Sie können Cluster jetzt vom deaktivierten Clustermodus (CMD) zum aktivierten Clustermodus (CME) migrieren. Weitere Informationen finden Sie unter [Ändern des Clustermodus](#).

11. Mai 2023

[ElastiCache für Redis unterstützt OSS jetzt das Ändern der Verschlüsselungseinstellungen bei der Übertragung](#)

Sie können jetzt die TLS-Konfiguration Ihrer Redis OSS-Cluster ändern, ohne die Cluster neu erstellen oder bereitstellen zu müssen oder die Anwendungsverfügbarkeit zu beeinträchtigen. Weitere Informationen finden Sie unter [Aktivieren der Verschlüsselung während der Übertragung für einen vorhandenen Cluster](#).

28. Dezember 2022

[ElastiCache für Redis unterstützt OSS jetzt die Authentifizierung von Benutzern mithilfe von IAM](#)

Mit der IAM-Authentifizierung können Sie eine Verbindung zu Redis OSS mithilfe von ElastiCache IAM-Identitäten authentifizieren. AWS Auf diese Weise können Sie Ihr Sicherheitsmodell stärken und viele administrative Sicherheitssaufgaben vereinfachen. Weitere Informationen finden Sie unter [Authentifizierung mit IAM](#).

16. November 2022

[ElastiCache für Redis OSS unterstützt jetzt Redis OSS 7](#)

Diese Version bietet Amazon mehrere neue Funktionen ElastiCache für Redis OSS: Redis OSS-Funktionen, ACL-Verbesserungen und Sharded Pub/Sub. [Weitere Informationen finden Sie unter Redis OSS Version 7.0 ElastiCache](#).

08. November 2022

[ElastiCache für Redis OSS unterstützt jetzt IPv6](#)

07. November 2022

ElastiCache unterstützt die Internetprotokoll-Versionen 4 und 6 (IPv4 und IPv6), sodass Sie Ihren Cluster so konfigurieren können, dass er nur IPv4 Verbindungen, nur Verbindungen oder beides IPv4 und IPv6 IPv6 Verbindungen (Dual-Stack) akzeptiert. IPv6 [wird für Workloads unterstützt, die die Redis OSS-Engine ab Version 6.2 auf allen auf dem Nitro-System basierenden Instanzen verwenden.](#) Für den Zugriff über fallen keine zusätzlichen Gebühren an. ElastiCache IPv6 Weitere Informationen zu Netzwerktypen finden Sie unter [Auswählen eines Netzwerktyps.](#)

[ElastiCache for Memcached unterstützt jetzt IPV6](#)

07. November 2022

ElastiCache unterstützt die Internetprotokoll-Versionen 4 und 6 (IPv4 und IPv6), sodass Sie Ihren Cluster so konfigurieren können, dass er nur IPv4 Verbindungen, nur Verbindungen oder beides IPv4 und IPv6 IPv6 Verbindungen (Dual-Stack) akzeptiert. [IPv6 wird für Workloads unterstützt, die die Memcached Engine ab Version 1.6.6 auf allen auf dem Nitro-System basierenden Instanzen verwenden.](#) Für den Zugriff über fallen keine zusätzlichen Gebühren an. ElastiCache IPv6 Weitere Informationen zu Netzwerktypen finden Sie unter [Auswählen eines Netzwerktyps.](#)

[ElastiCache for Memcached unterstützt jetzt die Verschlüsselung bei der Übertragung](#)

Die Verschlüsselung während der Übertragung ist eine optionale Funktion, mit der Sie die Sicherheit Ihrer Daten an ihrer verwundbarsten Stelle verbessern können – bei der Übertragung von einem Speicherort an einen anderen. Sie wird in Memcached-Versionen 1.6.12 und höher unterstützt. Weitere Informationen finden Sie unter [Verschlüsselung bei der ElastiCache Übertragung \(TLS\)](#).

26. Mai 2022

[ElastiCache für Redis unterstützt OSS jetzt das native Format JavaScript Object Notation \(JSON\)](#)

Das native Format JavaScript Object Notation (JSON) ist eine einfache, schemalose Methode, um komplexe Datensätze innerhalb von Redis OSS-Clustern zu kodieren. Sie können Daten mithilfe des JSON-Formats (JavaScript Object Notation) nativ in Redis OSS-Clustern speichern und darauf zugreifen und die in diesen Clustern gespeicherten JSON-Daten aktualisieren, ohne benutzerdefinierten Code für die Serialisierung und Deserialisierung verwalten zu müssen. Weitere Informationen finden Sie unter [Erste Schritte mit JSON](#).

25. Mai 2022

[ElastiCache unterstützt jetzt PrivateLink](#)

AWS PrivateLink ermöglicht Ihnen den privaten Zugriff auf ElastiCache API-Operationen ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder AWS Direct Connect-Verbindung. Weitere Informationen finden Sie unter [Amazon ElastiCache API and interface VPC endpoints \(AWS PrivateLink\)](#) for Redis OSS oder [Amazon ElastiCache API and interface VPC endpoints](#) () for Memcached.AWS PrivateLink

24. Januar 2022

[ElastiCache für Redis OSS unterstützt jetzt Redis OSS 6.2 und Data Tiering](#)

Amazon ElastiCache for Redis OSS stellt die nächste Version der Redis OSS-Engine vor, die von Amazon unterstützt wird. ElastiCache ElastiCache für Redis OSS 6.2 beinhaltet Leistungsverbesserungen für TLS-fähige Cluster, die x86-Knotentypen mit 8 V CPUs oder mehr oder Graviton2-Knotentypen mit 4 V oder mehr verwenden. CPUs ElastiCache für Redis führt OSS auch Daten-Tiering ein. Sie können Daten-Tiering als kostengünstigere Methode verwenden, um Ihre Cluster auf bis zu Hunderte von Terabyte Kapazität zu skalieren.

[Weitere Informationen finden Sie unter ElastiCache Redis OSS Version 6.2 \(erweitert\) und unter Data Tiering.](#)

23. November 2021

[Support für Auto Scaling](#)

ElastiCache für Redis OSS unterstützt jetzt Auto Scaling. ElastiCache für Redis OSS ist Auto Scaling die Fähigkeit, die gewünschten Shards oder Replikate in Ihrem ElastiCache for Redis OSS-Service automatisch zu erhöhen oder zu verringern. ElastiCache nutzt den Application Auto Scaling Scaling-Dienst, um diese Funktionalität bereitzustellen. Weitere Informationen finden Sie unter [Auto Scaling ElastiCache für Redis OSS-Cluster](#).

19. August 2021

[Support für die Bereitstellung von Redis OSS Slow-Logs](#)

ElastiCache ermöglicht es Ihnen jetzt, Redis OSS SLOWLOG an eines von zwei Zielen zu streamen: Amazon Data Firehose oder Amazon Logs. CloudWatch Weitere Informationen finden Sie unter [Protokollauslieferung](#).

22. April 2021

[Support für die Markierung von Ressourcen und Bedingungsschlüsseln](#)

ElastiCache unterstützt jetzt Tagging, um Sie bei der Verwaltung Ihrer Cluster und anderer Ressourcen zu unterstützen. ElastiCache Weitere Informationen finden Sie unter [Taggen Ihrer ElastiCache Ressource](#)[n](#). ElastiCache führt auch Unterstützung für Bedingungsschlüssel ein. Sie können Bedingungen angeben, die bestimmen, wie eine IAM-Richtlinie wirksam wird. Weitere Informationen finden Sie unter [Verwenden von Bedingungsschlüssel](#).

7. April 2021

[Support für die Markierung von Ressourcen und Bedingungsschlüsseln](#)

ElastiCache unterstützt jetzt Tagging, um Sie bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen zu unterstützen. Weitere Informationen finden Sie unter [Taggen Ihrer ElastiCache Ressource](#)[n](#). ElastiCache führt auch Unterstützung für Bedingungsschlüssel ein. Sie können Bedingungen angeben, die bestimmen, wie eine IAM-Richtlinie wirksam wird. Weitere Informationen finden Sie unter [Verwenden von Bedingungsschlüssel](#).

7. April 2021

[ElastiCache ist jetzt auf AWS Outposts verfügbar](#)

[AWS Outposts](#) bieten native AWS Dienste, Infrastrukturen und Betriebsmodelle für praktisch jedes Rechenzentrum, jeden Colocation-Bereich oder jede lokale Einrichtung. Sie können ElastiCache auf Outposts bereitstellen, um Cache vor Ort einzurichten, zu betreiben und zu verwenden, genau wie in der Cloud. Weitere Informationen finden Sie unter [Outposts verwenden](#).

8. Oktober 2020

[ElastiCache unterstützt jetzt Redis OSS 6](#)

Amazon ElastiCache for Redis OSS stellt die nächste Version der Redis OSS-Engine vor, die von Amazon unterstützt wird. ElastiCache Diese Version umfasst die [Authentifizierung von Benutzern mit rollenbasierter Zugriffskontrolle](#), versionlosen Support, clientseitiges Caching und erhebliche betriebliche Verbesserungen. Weitere Informationen finden Sie unter [ElastiCache Redis OSS Version 6.0 \(erweitert\)](#).

7. Oktober 2020

[ElastiCache unterstützt jetzt Local Zones](#)

Eine lokale Zone ist eine Erweiterung einer AWS Region, die sich geografisch in der Nähe Ihrer Benutzer befindet. Sie können jede Virtual Private Cloud (VPC) von einer übergeordneten AWS Region auf Local Zones erweitern, indem Sie ein neues Subnetz erstellen und es einer lokalen Zone zuweisen. Weitere Informationen finden Sie unter [Local Zones nutzen](#).

25. September 2020

[ElastiCache for Redis OSS unterstützt jetzt die Skalierung Ihrer Redis OSS-Cluster-Umgebung auf bis zu 500 Knoten oder 500 Shards](#)

Der Redis OSS-Clustermodus ermöglicht Konfigurationen, mit denen Sie Ihre Daten auf mehrere Shards partitionieren können, und bietet eine bessere Skalierbarkeit, Leistung und Verfügbarkeit. Diese Funktion ist auf Amazon ElastiCache für Redis OSS ab Version 5.0.6 in allen AWS Regionen und für alle bestehenden und neuen Cluster-Umgebungen ElastiCache für Redis OSS verfügbar. Weitere Informationen finden Sie unter [Redis OSS](#) Nodes and Shards.

13. August 2020

[ElastiCache unterstützt jetzt Berechtigungen auf Ressourceebene](#)

Sie können jetzt den Umfang der Benutzerberechtigungen einschränken, indem Sie ElastiCache Ressourcen in einer AWS Identity and Access Management(IAM-) Richtlinie angeben. Weitere Informationen finden Sie unter [Berechtigungen auf Ressourceebene](#).

12. August 2020

[ElastiCache für Redis OSS fügt zusätzliche CloudWatch Amazon-Metriken hinzu](#)

ElastiCache für Redis unterstützt OSS jetzt neue CloudWatch Metriken, einschließlich PubSubCmds und. HyperLogLogBasedCmds Eine vollständige Liste finden Sie unter [Metrics for Redis OSS](#).

10. Juni 2020

[ElastiCache unterstützt jetzt die automatische Aktualisierung von Clustern ElastiCache](#)

Amazon unterstützt ElastiCache jetzt die automatische Aktualisierung von ElastiCache Clustern, nachdem das „empfohlene Verfallsdatum“ des Service-Updates abgelaufen ist. ElastiCache verwendet Ihr Wartungsfenster, um die automatische Aktualisierung der entsprechenden Cluster zu planen. Weitere Informationen finden Sie unter [Self-Service-Updates](#).

13. Mai 2020

[ElastiCache für Redis OSS
unterstützt jetzt Global
Datastore für Redis OSS](#)

Die Funktion Global Datastore for Redis OSS bietet eine vollständig verwaltete, schnelle, zuverlässige und sichere Replikation über Regionen hinweg. AWS Mit dieser Funktion können Sie regionsübergreifende Read Replica-Cluster ElastiCache für Redis OSS erstellen, um Lesevorgänge mit niedriger Latenz und regionsübergreifende Disaster Recovery zu ermöglichen. AWS Sie können einen globalen Datenspeicher erstellen, ändern und beschreiben. Sie können auch AWS Regionen zu Ihrem globalen Datenspeicher hinzufügen oder daraus entfernen und eine Region innerhalb eines globalen Datenspeichers als primäre AWS Region heraufstufen. Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation](#) mithilfe von Global Datastore.

16. März 2020

[ElastiCache für Redis OSS
unterstützt jetzt Redis OSS
Version 5.0.6](#)

Weitere Informationen finden Sie unter [ElastiCache Redis OSS Version 5.0.6](#) (erweitert).

18. Dezember 2019

[Amazon unterstützt ElastiCache jetzt T3-Standard-Cache-Knoten](#)

Sie können jetzt die Allzweck-Burstable-T3-Standard-Cache-Knoten der nächsten Generation in Amazon starten. ElastiCache EC2Die T3-Standard-Instances von Amazon bieten eine grundlegende CPU-Leistung und können die CPU-Auslastung jederzeit erhöhen, bis das aufgelaufene Guthaben aufgebraucht ist. Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

12. November 2019

[Amazon unterstützt ElastiCache jetzt die Änderung des AUTH-Tokens auf einem vorhandenen ElastiCache OSS-Server für Redis](#)

ElastiCache für Redis OSS 5.0.6 können Sie jetzt Authentifizierungstoken ändern, indem Sie neue Token setzen und rotieren. Sie können aktive Token jetzt ändern, während sie in Gebrauch sind. Sie können vorhandenen Clustern mit aktivierter Verschlüsselung während der Übertragung, die zuvor ohne Authentifizierungstoken eingerichtet wurden, auch ganz neue Token hinzufügen. Dies ist ein zweistufiger Prozess, mit dem Sie das Token festlegen und rotieren können, ohne die Client-Anforderungen zu unterbrechen. Diese Funktion wird derzeit nicht unterstützt auf CloudFormation. Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit dem Redis OSS AUTH-Befehl](#).

30. Oktober 2019

[Amazon unterstützt ElastiCache jetzt die Online-Datenmigration von Redis OSS auf Amazon EC2](#)

Sie können jetzt Online Migration verwenden, um Ihre Daten von selbst gehosteten Redis OSS auf Amazon EC2 zu migrieren. Weitere Informationen finden Sie unter [Online-Migration](#) zu ElastiCache.

28. Oktober 2019

[ElastiCache für Redis OSS führt vertikale Online-Skalierung für den Redis OSS-Clustermodus ein.](#)

Sie können Ihren Sharded Redis OSS-Cluster jetzt bei Bedarf hoch- oder runterskalieren. ElastiCache for Redis OSS ändert die Größe Ihres Clusters, indem der Knotentyp geändert wird, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Weitere Informationen finden Sie unter [Vertikales Online-Skalieren durch Ändern des Knotentyps.](#)

20. August 2019

[ElastiCache for Redis OSS ermöglicht es Benutzern jetzt, einen einzigen Reader-Endpoint für Ihren Amazon ElastiCache for Redis OSS-Cluster zu verwenden.](#)

Mit dieser Funktion können Sie den gesamten Lesetraffic über einen einzigen Endpunkt auf Clusterebene an Ihren ElastiCache for Redis OSS-Cluster weiterleiten, um den Lastenausgleich und die höhere Verfügbarkeit zu nutzen. Weitere Informationen finden Sie unter [Finden von Verbindungsendpunkten.](#)

13. Juni 2019

[ElastiCache für Redis OSS
können Benutzer jetzt Service-
Updates nach ihrem eigenen
Zeitplan anwenden](#)

Dank dieser Funktion können Sie verfügbare Service-Updates zu einem Zeitpunkt Ihrer Wahl anwenden, nicht nur in Wartungsfenstern. Auf diese Weise werden Dienstunterbrechungen, insbesondere bei hohen Geschäftsabläufen, minimiert und Sie können sicherstellen, dass Sie die Vorschriften einhalten, wenn Ihr Cluster an ElastiCache unterstützten Compliance-Programmen teilnimmt. Weitere Informationen finden Sie unter [Self-Service-Updates bei Amazon ElastiCache](#) und [Compliance-Validierung für Amazon ElastiCache](#).

4. Juni 2019

[ElastiCache Standardangebote für Reserved Instances: Teilweise Vorauszahlung, vollständige Vorauszahlung und keine Vorauszahlung.](#)

Reserved Instances bieten Ihnen die Flexibilität, je nach ElastiCache Instance-Typ und AWS Region eine Amazon-Instance für einen Zeitraum von einem oder drei Jahren zu reservieren. Weitere Informationen finden Sie unter [Verwalten der Kosten mit reservierten Knoten](#).

18. Januar 2019

[ElastiCache für Redis OSS-
Unterstützung für bis zu 250
Knoten pro Redis OSS-Cluster](#)

Das Knoten- oder Shard-Limit kann auf maximal 250 pro ElastiCache Redis OSS-Cluster erhöht werden. Weitere Informationen finden Sie unter [Shards](#).

19. November 2018

[ElastiCache für Redis OSS-
Unterstützung für Autofailover
und Sicherung und Wiederherstellung auf allen T2-Knoten](#)

ElastiCache für Redis bietet OSS Unterstützung für Autofailover, Erstellung von Snapshots sowie Sicherung und Wiederherstellung auf allen T2-Knoten. Weitere Informationen finden Sie unter [ElastiCache für Redis OSS Backup and Restore und Snapshot](#).

19. November 2018

[ElastiCache für Redis OSS-
Unterstützung für M5- und R5-
Knoten](#)

ElastiCache für Redis unterstützt OSS jetzt M5- und R5-Knoten, universelle und speicheroptimierte Instance-Typen, die auf dem Nitro-System basieren. AWS Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

23. Oktober 2018

[Unterstützung für eine dynamische Änderung der Anzahl von Read Replicas](#)

ElastiCache für Redis bietet OSS zusätzliche Unterstützung für das Hinzufügen und Entfernen von Read Replicas aus beliebigen Clustern ohne Cluster-Ausfallzeiten. Weitere Informationen zu diesen und anderen Änderungen in dieser Version finden Sie unter [Ändern der Anzahl der Replikate im Benutzerhandbuch ElastiCache](#) für Redis OSS. Siehe auch [DecreaseReplicaCount](#) und [IncreaseReplicaCount](#) in der ElastiCache API-Referenz.

17. September 2018

[Zertifizierung der FedRAMP-Compliance](#)

ElastiCache for Redis OSS ist jetzt für die FedRAMP-Konformität zertifiziert. Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

30. August 2018

[Valkey- oder Redis OSS-Engine-Upgrades \(Cluster-Modus aktiviert\)](#)

Amazon ElastiCache for Redis OSS hat Unterstützung für Upgrades von Valkey- oder Redis OSS-Engine-Versionen (Cluster-Modus aktiviert) hinzugefügt. Weitere Informationen finden Sie unter [Upgrade von Engine-Versionen](#).

20. August 2018

[PCI DSS Compliance-Zertifizierung](#)

ElastiCache for Redis OSS ist jetzt für die PCI-DSS-Konformität zertifiziert. Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

5. Juli 2018

[Support ElastiCache für Redis OSS 4.0.10](#)

ElastiCache für Redis OSS unterstützt jetzt Redis OSS 4.0.10, einschließlich Verschlüsselung und Online-Clustergrößenänderung in einer einzigen Version. Weitere Informationen finden Sie unter [Redis OSS Version ElastiCache 4.0.10](#) (erweitert).

14. Juni 2018

[Neu strukturiertes Benutzerhandbuch](#)

[Das einheitliche ElastiCache Benutzerhandbuch ist jetzt neu strukturiert, sodass es separate Benutzerhandbücher für Redis OSS \(für das Redis OSS-Benutzerhandbuch\) und ElastiCache für Memcached \(für das Memcached-Benutzerhandbuch\) gibt. ElastiCache](#)
Die Dokumentationsstruktur im Abschnitt [AWS CLI Command Reference: elasticache](#) und in der [Amazon ElastiCache API-Referenz](#) bleibt unverändert.

20. April 2018

[Support für Engine CPU Utilization Metric](#)

ElastiCache für Redis OSS wurde eine neue Metrik hinzugefügt `EngineCPUUtilization`, die den Prozentsatz Ihrer CPU-Kapazität angibt, der derzeit genutzt wird. Weitere Informationen finden Sie unter [Metriken für Redis OSS](#).

9. April 2018

In der folgenden Tabelle werden die wichtigen Änderungen am ElastiCache Benutzerhandbuch vor März 2018 beschrieben.

Änderungen	Beschreibung	Änderungsdatum
Support für die Region Asien-Pazifik (Osaka – regional).	ElastiCache Unterstützung für die Region Asien-Pazifik (lokal in Osaka) hinzugefügt. Die Region Asien-Pazifik (Osaka) unterstützt derzeit eine einzige Availability Zone und ist nur auf Einladung zugänglich. Weitere Informationen finden Sie hier: • Unterstützte Regionen • Unterstützte Cache-Knotentypen	12. Februar 2018
Support für die Europa (Paris).	ElastiCache Unterstützung für die Region EU (Paris) hinzugefügt. Weitere Informationen finden Sie hier: • Unterstützte Regionen • Unterstützte Cache-Knotentypen	18. Dezember 2017
Support für die Region China (Ningxia)	Amazon ElastiCache hat Unterstützung für die Region China (Ningxia) hinzugefügt. Weitere Informationen finden Sie hier:	11. Dezember 2017

Änderungen	Beschreibung	Änderungsdatum
	• Unterstützte Regionen • Unterstützte Cache-Knotentypen	
Unterstützung für dienstverknüpfte Rollen	In dieser Version ElastiCache wurde Unterstützung für Service Linked Roles (SLR) hinzugefügt. Weitere Informationen finden Sie hier: • Verwenden von serviceverknüpften Rollen für Amazon ElastiCache • Richten Sie Ihre Berechtigungen ein (nur ElastiCache für neue Benutzer)	7. Dezember 2017
Unterstützung für R4-Knotentypen	Diese Version von ElastiCache hat Unterstützung für R4-Knotentypen in allen AWS Regionen hinzugefügt, die von unterstützt werden. ElastiCache R4-Knotentypen sind als On-Demand-Knoten oder als reservierte Cache-Knoten erhältlich. Weitere Informationen finden Sie hier: • Unterstützte Cache-Knotentypen • Spezifische Parameter des Memcached-Knotentyps • Redis OSS-Knotentyp-spezifische Parameter	20. November 2017

Änderungen	Beschreibung	Änderungsdatum
ElastiCache für Redis OSS 3.2.10 und Unterstützung für Online-Resharding	Amazon ElastiCache for Redis OSS bietet Unterstützung ElastiCache für Redis OSS 3.2.10. ElastiCache für Redis OSS wird auch die Online-Clustergrößenänderung eingeführt, um Shards zum Cluster hinzuzufügen oder daraus zu entfernen , während der Cluster weiterhin eingehende Anfragen bearbeitet. I/O Weitere Informationen finden Sie hier: • Online-Größenanpassung von Clustern • Online-Resharding für Valkey oder Redis OSS (Clustermodus aktiviert)	9. November 2017
HIPAA-Konformität	ElastiCache für Redis OSS ist Version 3.2.6 jetzt HIPAA-tauglich, wenn die Verschlüsselung auf Ihrem Cluster aktiviert ist. Weitere Informationen finden Sie hier: • Konformitätsvalidierung für Amazon ElastiCache • Datensicherheit bei Amazon ElastiCache	2. November 2017

Änderungen	Beschreibung	Änderungsdatum
ElastiCache für Redis OSS 3.2.6 und Unterstützung für Verschlüsselung	ElastiCache fügt Unterstützung ElastiCache für Redis OSS 3.2.6 hinzu, die zwei Verschlüsselungsfunktionen beinhaltet: • Die Verschlüsselung der Daten während der Übertragung verschlüsselt Ihre Daten bei jeder Übertragung, z. B. zwischen Knoten in einem Cluster oder zwischen einem Cluster und Ihrer Anwendung. • Bei der Verschlüsselung im Ruhezustand werden Ihre Daten auf dem Laufwerk bei Synchronisierungs- und Sicherungsoperationen verschlüsselt. Weitere Informationen finden Sie hier: • Datensicherheit bei Amazon ElastiCache • Unterstützte Engines und Versionen	25. Oktober 2017
Thema der Verbindungsmuster	ElastiCache Die Dokumentation fügt ein Thema hinzu, das verschiedene Muster für den Zugriff auf einen ElastiCache Cluster in einer Amazon VPC behandelt. Weitere Informationen finden Sie unter Zugriffsmuster für den Zugriff auf einen ElastiCache Cluster in einer Amazon VPC im ElastiCache -Benutzerhandbuch.	24. April 2017

Änderungen	Beschreibung	Änderungsdatum
Support für Memcached 1.4.34	ElastiCache fügt Unterstützung für Memcached Version 1.4.34 hinzu, die eine Reihe von Korrekturen für frühere Memcached-Versionen beinhaltet. Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.4.34 auf Memcached am. GitHub	10. April 2017
Unterstützung für das Testen des automatischen Failovers	ElastiCache fügt Unterstützung für das Testen von Automatic Failover auf Redis OSS-Clustern hinzu, die Replikation unterstützen. Weitere Informationen finden Sie hier: • Testen des automatischen Failovers im ElastiCache -Benutzerhandbuch. • TestFailover in der API-Referenz zu ElastiCache • test-failover in der AWS CLI-Referenz.	4. April 2017
Verbesserte Redis OSS-Wiederherstellung	ElastiCache fügt erweiterte Redis OSS-Backup- und Wiederherstellungsfunktionen mit Cluster-Größenänderung hinzu. Diese Funktion unterstützt die Wiederherstellung einer Sicherung in einem Cluster mit einer unterschiedlichen Anzahl an Shards als bei dem Cluster, der zur Erstellung der Sicherung verwendet wurde. (Bei der API und der CLI wird mit dieser Funktion statt einer unterschiedlichen Anzahl an Shards eine unterschiedliche Anzahl an Knotengruppen wiederhergestellt.) Dieses Update unterstützt auch verschiedene Redis OSS-Steckplatzkonfigurationen. Weitere Informationen finden Sie unter Wiederherstellen aus einem Backup in einen neuen Cache.	15. März 2017

Änderungen	Beschreibung	Änderungsdatum
Neuer Redis OSS-Speicherwartungsparameter	ElastiCache fügt einen neuen Redis OSS-Parameter <code>hzreserved-memory-percent</code> hinzu, der die Verwaltung Ihres reservierten Speichers erleichtert. Dieser Parameter ist in allen Versionen von ElastiCache für Redis OSS verfügbar. Weitere Informationen finden Sie hier: • Verwaltung von reserviertem Speicher für Valkey und Redis OSS • Neue Parameter für Redis OSS 3.2.4	15. März 2017
Unterstützung für Memcached 1.4.33	ElastiCache fügt Unterstützung für Memcached Version 1.4.33 hinzu. Weitere Informationen finden Sie hier: • ElastiCache Version 1.4.33 für Memcached • In Memcached 1.4.33 hinzugefügte Parameter	20. Dezember 2016
Unterstützung für die Region EU (London)	ElastiCache fügt Unterstützung für die Region EU (London) hinzu. Derzeit werden nur die Knotentypen T2 und M4 unterstützt. Weitere Informationen finden Sie hier: • Unterstützte Regionen • Unterstützte Cache-Knotentypen	13. Dezember 2016
Unterstützung für die Region Kanada (Montreal)	ElastiCache fügt Unterstützung für die Region Kanada (Montreal) hinzu. In dieser AWS Region werden derzeit nur die Knotentypen M4 und T2 unterstützt. Weitere Informationen finden Sie hier: • Unterstützte Regionen • Unterstützte Cache-Knotentypen	8. Dezember 2016

Änderungen	Beschreibung	Änderungsdatum
Unterstützung für Knotentypen M4 und R3	ElastiCache fügt Unterstützung für R3- und M4-Knotentypen in der Region Südamerika (São Paulo) und M4-Knotentypen in der Region China (Peking) hinzu. Weitere Informationen finden Sie hier: • Unterstützte Regionen • Unterstützte Cache-Knotentypen	1. November 2016
Unterstützung für die Region USA Ost 2 (Ohio)	ElastiCache fügt Unterstützung für die Region USA Ost (Ohio) (us-east-2) mit den Knotentypen M4, T2 und R3 hinzu. Weitere Informationen finden Sie hier: • Unterstützte Regionen • Unterstützte Cache-Knotentypen	17. Oktober 2016

Änderungen	Beschreibung	Änderungsdatum
Support für Redis OSS Cluster	ElastiCache fügt Unterstützung für Redis OSS Cluster (erweitert) hinzu. Kunden, die Redis OSS Cluster verwenden, können ihre Daten auf bis zu 15 Shards (Knotengruppen) partitionieren. Jeder Shard unterstützt Replikation mit bis zu 5 Lesereplikaten pro Shard. Die automatischen Failover-Zeiten des Redis OSS-Clusters sind etwa ein Viertel so lang wie bei früheren Versionen. Diese Version umfasst eine neu gestaltete Verwaltungskonsole, die Terminologie entsprechend der Branchennutzung verwendet. Weitere Informationen finden Sie hier: • Vergleich von Memcached und Redis OSS • ElastiCache Komponenten und Funktionen – Beachten Sie die Abschnitte zu Knoten, Shards, Clustern und Replikation. • ElastiCache Terminologie	12. Oktober 2016
Unterstützung für Knotentyp M4	ElastiCache fügt Unterstützung für die M4-Familie von Knotentypen in den meisten AWS Regionen hinzu, die von unterstützt werden. ElastiCache M4-Knotentypen sind als On-Demand-Knoten oder als reservierte Cache-Knoten erhältlich. Weitere Informationen finden Sie hier: • Unterstützte Cache-Knotentypen • Spezifische Parameter des Memcached-Knotentyps • Redis OSS-Knotentyp-spezifische Parameter	3. August 2016

Änderungen	Beschreibung	Änderungsdatum
Unterstützung der Region Mumbai	ElastiCache fügt Unterstützung für die Region Asien-Pazifik (Mumbai) hinzu. Weitere Informationen finden Sie hier: • Unterstützte Cache-Knotentypen • Spezifische Parameter des Memcached-Knotentyps • Redis OSS-Knotentyp-spezifische Parameter	27. Juni 2016
Snapshot-Export	ElastiCache fügt die Möglichkeit hinzu, einen Redis OSS-Snapshot zu exportieren, sodass Sie von außen ElastiCache darauf zugreifen können. Weitere Informationen finden Sie hier: • Exportieren einer Sicherung im ElastiCache Amazon-Benutzerhandbuch • CopySnapshot in der Amazon ElastiCache API-Referenz	26. Mai 2016
Vertikale Skalierung von Knotentypen	ElastiCache fügt die Möglichkeit hinzu, Ihren Redis OSS-Knotentyp zu skalieren. Weitere Informationen finden Sie unter Skalierung ElastiCache.	24. März 2016
Einfaches Upgraden der Engine	ElastiCache bietet die Möglichkeit, Ihre Redis OSS-Cache-Engine einfach zu aktualisieren. Weitere Informationen finden Sie unter Versionsverwaltung für ElastiCache.	22. März 2016
Unterstützung für R3-Knotentypen	ElastiCache fügt Unterstützung für R3-Knotentypen in den Regionen China (Peking) und Südamerika (São Paulo) hinzu. Weitere Informationen finden Sie unter Unterstützte Cache-Knotentypen.	16. März 2016

Änderungen	Beschreibung	Änderungsdatum
Zugreifen ElastiCache mit einer Lambda-Funktion	Es wurde ein Tutorial zur Konfiguration einer Lambda-Funktion für den Zugriff ElastiCache in einer Amazon VPC hinzugefügt. Weitere Informationen finden Sie unter Weitere ElastiCache Tutorials und Videos .	12. Februar 2016
Support für Redis OSS 2.8.24	ElastiCache fügt Unterstützung für Redis OSS Version 2.8.24 hinzu, wobei Verbesserungen seit Redis OSS 2.8.23 hinzugefügt wurden. Zu den Verbesserungen gehören Fehlerbehebungen und die Unterstützung einer Protokollierung von Adressen ungültiger Speicherzugriffe. Weitere Informationen finden Sie hier: • ElastiCache Version 2.8.24 für Redis OSS (erweitert) • Versionshinweise zu Redis OSS 2.8	20. Januar 2016
Support für die Region Asien-Pazifik (Seoul)	ElastiCache fügt Unterstützung für die Region Asien-Pazifik (Seoul) (ap-northeast-2) mit den Knotentypen t2, m3 und r3 hinzu.	6. Januar 2016
Änderung der ElastiCache Amazon-Konsole.	Da die neueren Redis OSS-Versionen eine bessere und stabilere Benutzererfahrung bieten, sind die Redis OSS-Versionen 2.6.13, 2.8.6 und 2.8.19 nicht mehr in der Management Console aufgeführt. ElastiCache Weitere Optionen und mehr Informationen finden Sie unter Unterstützte Engines und Versionen .	15. Dezember 2015

Änderungen	Beschreibung	Änderungsdatum
Support für Redis OSS 2.8.23.	ElastiCache fügt Unterstützung für Redis OSS Version 2.8.23 hinzu, wobei Verbesserungen seit Redis OSS 2.8.22 hinzugefügt wurden. Die Verbesserungen umfassen Fehlerkorrekturen sowie Support für den neuen Parameter <code>close-on-slave-write</code> der in aktiviertem Zustand die Verbindung zu Clients trennt, die versuchen, in ein schreibgeschütztes Replikat zu schreiben. Weitere Informationen finden Sie unter ElastiCache Version 2.8.23 für Redis OSS (erweitert) .	13. November 2015

Änderungen	Beschreibung	Änderungsdatum
Support für Redis OSS 2.8.22.	ElastiCache fügt Unterstützung für Redis OSS Version 2.8.22 mit ElastiCache zusätzlichen Erweiterungen und Verbesserungen seit Version 2.8.21 hinzu. Zu den Verbesserungen gehören: • Implementierung eines unvergabelten Speicherprozesses, der eine erfolgreiche Speicherung ermöglicht, wenn aufgrund wenig verfügbarem Arbeitsspeicher eine vergabelte Speicherung fehlschlagen würde. • Zusätzliche Metriken — und. CloudWatch <code>SaveInProgressReplicationBytes</code> • Um Teilsynchronisationen zu ermöglichen, gilt der Redis OSS-Parameter <code>repl-backlog-size</code> jetzt für alle Cluster. Eine komplette Liste der Änderungen und weitere Informationen finden Sie unter ElastiCache Version 2.8.22 für Redis OSS (erweitert). Diese Dokumentationsversion beinhaltet eine Neuorganisation der Dokumentation und die Entfernung der Dokumentation zur ElastiCache Befehlszeilenschnittstelle (CLI). Informationen zur Verwendung über die Befehlszeile finden Sie in der AWS Befehlszeile für ElastiCache.	28. September 2015

Änderungen	Beschreibung	Änderungsdatum
Unterstützung für Memcached 1.4.28	ElastiCache fügt Unterstützung für Memcached Version 1.4.24 und Memcached-Verbesserungen seit Version 1.4.14 hinzu. Diese Version bietet jetzt Unterstützung für LRU-Cache-Verwaltung (en: Least Recently Used) als Hintergrundaufgabe, die Auswahl von jenkins oder murmur3 als Hash-Algorithmus, neue Befehle und verschiedene Fehlerkorrekturen. Weitere Informationen finden Sie unter den Memcached-Versionshinweisen .	27. August 2015
Unterstützung von Memcached Auto Discovery mit PHP 5.6	Diese Version von Amazon ElastiCache bietet Unterstützung für den Memcached Auto Discovery -Client für PHP Version 5.6. Weitere Informationen finden Sie unter Den Quellcode für den ElastiCache Cluster-Client für PHP kompilieren .	29. Juli 2015
Support für Redis OSS 2.8.21	ElastiCache fügt Unterstützung für Redis OSS Version 2.8.21 und Redis OSS-Verbesserungen seit Version 2.8.19 hinzu. Diese Redis OSS-Version enthält mehrere Bugfixes. Weitere Informationen finden Sie in den Versionshinweisen zu Redis OSS 2.8 .	29. Juli 2015
Neues Thema: Zugriff ElastiCache von außen AWS	Es wurde ein neues Thema zum Zugriff auf ElastiCache Ressourcen von außen hinzugefügt AWS. Weitere Informationen finden Sie unter Zugriff auf ElastiCache Ressourcen von außen AWS .	9. Juli 2015

Änderungen	Beschreibung	Änderungsdatum
Nachrichten über Knotenersatz wurden hinzugefügt	ElastiCache fügt drei Meldungen hinzu, die sich auf den geplanten Austausch von Knoten beziehen: <code>ElastiCache: NodeReplacementScheduled</code>, <code>ElastiCache: NodeReplacementRescheduled</code>, <code>ElastiCache: NodeReplacementCanceled</code>. Weitere Informationen und Maßnahmen, die Sie ergreifen können, wenn der Austausch eines Knotens geplant ist, finden Sie unter ElastiCache: Ereignisbenachrichtigungen und Amazon SNS.	11. Juni 2015

Änderungen	Beschreibung	Änderungsdatum
Support für Redis OSS v. 2.8.19.	ElastiCache fügt Unterstützung für Redis OSS Version 2.8.19 und Redis OSS-Verbesserungen seit Version 2.8.6 hinzu. Diese Unterstützung enthält Unterstützung für: • Die HyperLogLog Datenstruktur mit den Redis OSS-Befehlen PFADD, PFCOUNT und PFMERGE. • Lexikografische Bereichsabfragen mit den neuen Befehlen ZRANGEBYLEX, ZLEXCOUNT und ZREMRANGEBYLEX • Es wurde eine Reihe von Fehlerkorrekturen vorgenommen, insbesondere um zu verhindern, dass ein Primärknoten veraltete Daten an Replikationsknoten sendet, indem der primäre SYNC fehlschlägt, wenn ein Hintergrundspeicherprozess (bgsave) unerwartet beendet wird. Weitere Informationen zu finden Sie unter Neue HyperLogLog Datenstruktur von Redis OSS: die HyperLogLog Weitere Informationen zu PFADD, PFCOUNT und PFMERGE finden Sie in der Redis OSS-Dokumentation und klicken Sie auf. HyperLogLog	11. März 2015
Unterstützung von Kostenzuordnungs-Tags	ElastiCache fügt Unterstützung für Kostenzuordnungs-Tags hinzu. Weitere Informationen finden Sie unter Überwachung von Kosten mit Kostenzuordnungs-Tags.	9. Februar 2015

Änderungen	Beschreibung	Änderungsdatum
Support für die AWS GovCloud Region (US-West)	ElastiCache fügt Unterstützung für die Region AWS GovCloud (US-West) (us-gov-west-1) hinzu.	29. Januar 2015
Support für die Region Europa (Frankfurt)	ElastiCache fügt Unterstützung für die Region Europa (Frankfurt) (eu-central-1) hinzu.	19. Januar 2015
Multi-AZ-Unterstützung für Redis OSS-Replikationsgruppen	ElastiCache fügt Unterstützung für Multi-AZ vom primären Knoten zu einer Read Replica in einer Redis OSS-Replikationsgruppe hinzu. ElastiCache überwacht den Zustand der Replikationsgruppe. Wenn das Primärreplikat ausfällt, ElastiCache wird ein Replikat automatisch zum Primärreplikat heraufgestuft und dann das Replikat ersetzt. Weitere Informationen finden Sie unter Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS .	24. Oktober 2014
AWS CloudTrail Protokollierung von API-Aufrufen wird unterstützt	ElastiCache fügt Unterstützung für AWS CloudTrail die Protokollierung aller ElastiCache API-Aufrufe hinzu. Weitere Informationen finden Sie unter Protokollierung Amazon ElastiCache Amazon-API-Aufrufen mit AWS CloudTrail .	15. September 2014
Es werden neue Instance-Größen unterstützt	ElastiCache fügt Unterstützung für zusätzliche General Purpose (T2) -Instances hinzu. Weitere Informationen finden Sie unter Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen .	11. September 2014

Änderungen	Beschreibung	Änderungsdatum
Flexibler Knotenersatz für Memcached wird unterstützt	ElastiCache fügt Unterstützung für die Erstellung von Memcached-Knoten in mehreren Availability Zones hinzu.	23. Juli 2014
Es werden neue Instance-Größen unterstützt	ElastiCache fügt Unterstützung für zusätzliche General Purpose (M3) -Instances und Memory Optimized (R3) -Instances hinzu. Weitere Informationen finden Sie unter Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen .	1. Juli 2014
PHP-Auto Discovery	Unterstützung für Auto Discovery von PHP-Version 5.5 wurde hinzugefügt.	13. Mai 2014
Backup und Wiederherstellung für Redis OSS-Cluster	In dieser Version ElastiCache können Kunden Snapshots ihrer Redis OSS-Cluster erstellen und mithilfe dieser Snapshots neue Cluster erstellen. Ein Backup ist eine Kopie des Clusters zu einem bestimmten Zeitpunkt und besteht aus Cluster-Metadaten und allen Daten im Redis OSS-Cache. Sicherungen werden in Amazon S3 gespeichert. Kunden können die Daten von einem Snapshot jederzeit in einem neuen Cluster wiederherstellen. Weitere Informationen finden Sie unter Snapshot und Wiederherstellung .	24. April 2014

Änderungen	Beschreibung	Änderungsdatum
Redis OSS 2.8.6	ElastiCache unterstützt Redis OSS 2.8.6, zusätzlich zu Redis OSS 2.6.13. Mit Redis OSS 2.8.6 können Kunden die Stabilität und Fehlertoleranz von Read Replicas verbessern, indem sie die teilweise Resynchronisierung unterstützen und eine benutzerdefinierte Mindestanzahl von Read Replicas festlegen, die jederzeit verfügbar sein müssen. Redis OSS 2.8.6 bietet auch volle Unterstützung für publish-and-subscribe, bei der Clients über Ereignisse auf dem Server informiert werden können.	13. März 2014
Redis OSS-Cache-Engine	ElastiCache bietet zusätzlich zu Memcached die Redis OSS-Cache-Engine-Software an. Kunden, die derzeit Redis OSS verwenden, können einen neuen Redis OSS-Cluster mit ihren vorhandenen Daten aus einer ElastiCache Redis OSS-Snapshots-Datei „starten“, was die Migration zu einer verwalteten Umgebung erleichtert. ElastiCache Zur Unterstützung der Redis OSS-Replikationsfunktionen unterstützt die ElastiCache API jetzt Replikationsgruppen. Kunden können eine Replikationsgruppe mit einem primären Redis OSS-Cache-Knoten erstellen und einen oder mehrere Read Replica-Knoten hinzufügen, die automatisch mit den Cache-Daten im primären Knoten synchronisiert bleiben. Leseintensive Anwendungen können in ein Lesereplikat ausgelagert werden, wodurch die Arbeitslast auf dem Primärknoten reduziert wird. Lesereplikate können auch vor Datenverlust im Fall eines Ausfalls des primären Cache-Knotens schützen.	3. September 2013

Änderungen	Beschreibung	Änderungsdatum
Support für standardmäßige Amazon Virtual Private Cloud (VPC)	In dieser Version ElastiCache ist es vollständig in Amazon Virtual Private Cloud (VPC) integriert. Für Neukunden werden Cluster standardmäßig in einer Amazon VPC erstellt. Weitere Informationen finden Sie unter Amazon VPCs und ElastiCache Sicherheit .	8. Januar 2013
PHP-Unterstützung für Auto Discovery für Cache-Knoten	Die ursprüngliche Version von Auto Discovery für Cache-Knoten bot Unterstützung für Java-Programme. In dieser Version wird die ElastiCache auto Erkennung von Cache-Knoten in PHP unterstützt.	2. Januar 2013
Support für Amazon Virtual Private Cloud (VPC)	In dieser Version können ElastiCache Cluster in Amazon Virtual Private Cloud (VPC) gestartet werden. Standardmäßig werden Cluster für neue Kunden automatisch in einer Amazon VPC erstellt. Bestandskunden können in ihrem eigenen Tempo zu Amazon VPC migrieren. Weitere Informationen finden Sie unter Amazon VPCs und ElastiCache Sicherheit .	20. Dezember 2012

Änderungen	Beschreibung	Änderungsdatum
Auto Discovery für Cache-Knoten und neue Cache-Engine-Version	ElastiCache bietet die auto von Cacheknoten — die Fähigkeit von Client-Programmen, automatisch alle Cache-Knoten in einem Cluster zu ermitteln und Verbindungen zu all diesen Knoten zu initiieren und aufrechtzuerhalten. Diese Version bietet außerdem eine neue Cache-Engine-Version, Memcached 1.4.14. Diese neue Cache-Engine bietet erweiterte Slab-Ausgleichsfunktionen, entscheidende Leistungs- und Skalierungsverbesserungen sowie mehrere Fehlerkorrekturen. Es gibt verschiedene neue Cache-Parameter, die konfiguriert werden können. Weitere Informationen finden Sie unter Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen.	28. November 2012
Neue Cache-Knotentypen	Diese Version stellt vier neue Cache-Knotentypen bereit.	13. November 2012
Reservierte Cache-Knoten	Diese Version bietet nun Unterstützung für reservierte Cache-Knoten.	5. April 2012
Neues Handbuch	Dies ist die erste Version des ElastiCache Amazon-Benutzerhandbuchs.	22. August 2011

AWS Glossar

Die neueste AWS Terminologie finden Sie im [AWS Glossar](#) in der AWS-Glossar Referenz.

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.