



Benutzer-Leitfaden

AWS Fehlerinjektionsservice



AWS Fehlerinjektionsservice: Benutzer-Leitfaden

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

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

Table of Contents

Was ist AWS FIS?	1
Konzepte	1
Aktionen	2
Targets (Ziele)	2
Bedingungen beenden	2
Unterstützt AWS-Services	3
Greifen AWS Sie auf FIS zu	3
Preisgestaltung	4
Planung Ihrer Experimente	5
Grundprinzipien und Richtlinien	5
Richtlinien für die Planung von Experimenten	7
Komponenten der Versuchsvorlage	9
Syntax der Vorlage	10
Erste Schritte	10
Aktionen	10
Syntax der Aktion	11
Aktions-Identifikatoren	12
Aktionsparameter	12
Ziele der Aktion	13
Dauer der Aktion	14
Beispielaktionen	14
Ziele	17
Zielsyntax	18
Ressourcentypen	19
Identifizieren Sie die Zielressourcen	21
Auswahlmodus	25
Beispielziele	26
Beispielfilter	27
Bedingungen beenden	32
Syntax der Stoppbedingungen	33
Weitere Informationen	33
Die Rolle des Experiments	34
Voraussetzungen	34

Option 1: Erstellen Sie eine Experimentrolle und fügen Sie eine AWS verwaltete Richtlinie hinzu	36
Option 2: Erstellen Sie eine Experimentrolle und fügen Sie ein integriertes Richtliniendokument hinzu	37
Konfiguration des Versuchsberichts	39
Syntax der Konfiguration des Versuchsberichts	41
Berechtigungen für Versuchsberichte	43
Bewährte Methoden für Versuchsberichte	45
Experimentieroptionen	46
Ausrichtung auf Konten	47
Leerer Zielauflösungsmodus	48
Modus „Aktionen“	48
Referenz zu Aktionen	50
Aktionen zur Fehlerinjektion	51
aws:fis:inject-api-internal-error	51
aws:fis:inject-api-throttle-error	52
aws:fis:inject-api-unavailable-error	53
Aktion zur Wiederherstellung	53
aws:arc:start-zonal-autoshift	53
Warte auf Aktion	55
aws:fis:wait	55
CloudWatch Amazon-Aktionen	55
aws:cloudwatch:assert-alarm-state	55
Amazon DynamoDB DynamoDB-Aktionen	56
aws:dynamodb:global-table-pause-replication	56
Amazon Aurora DSQL-Aktionen	60
aws:dsq:cluster-connection-failure	60
Amazon EBS-Aktionen	61
aws:ebs:pause-volume-io	61
aws:ebs:volume-io-latency	62
Amazon EC2 EC2-Aktionen	63
aws:ec2:api-insufficient-instance-capacity-error	63
aws:ec2:asg-insufficient-instance-capacity-error	64
aws:ec2:reboot-instances	65
aws:ec2:send-spot-instance-interruptions	66
aws:ec2:stop-instances	67

aws:ec2:terminate-instances	68
Amazon ECS-Aktionen	68
aws:ecs:drain-container-instances	69
aws:ecs:stop-task	70
aws:ecs:task-cpu-stress	70
aws:ecs:task-io-stress	71
aws:ecs:task-kill-process	72
aws:ecs:task-network-blackhole-port	73
aws:ecs:task-network-latency	74
aws:ecs:task-network-packet-loss	75
Amazon EKS-Aktionen	77
aws:eks:inject-kubernetes-custom-resource	77
aws:eks:pod-cpu-stress	78
aws:eks:pod-delete	80
aws:eks:pod-io-stress	81
aws:eks:pod-memory-stress	82
aws:eks:pod-network-blackhole-port	83
aws:eks:pod-network-latency	84
aws:eks:pod-network-packet-loss	86
aws:eks:terminate-nodgroup-instances	87
ElastiCache Amazon-Aktionen	88
aws:elasticache:replicationgroup-interrupt-az-power	88
Amazon Kinesis Data Streams Streams-Aktionen	89
aws:kinesis:stream-provisioned-throughput-exception	89
aws:kinesis:stream-expired-iterator-exception	90
AWS Lambda Aktionen	90
aws:lambda:invocation-add-delay	91
aws:lambda:invocation-error	92
aws:lambda:invocation-http-integration-response	92
Amazon MemoryDB-Aktion	93
aws:memorydb:multi-region-cluster-pause-replication	93
Netzwerkaktionen	94
aws:network:disrupt-connectivity	94
aws:network:route-table-disrupt-cross-region-connectivity	96
aws:network:transit-gateway-disrupt-cross-region-connectivity	98
aws:network:disrupt-vpc-endpoint	98

Amazon RDS-Aktionen	99
aws:rds:failover-db-cluster	100
aws:rds:reboot-db-instances	100
Amazon-S3-Aktionen	101
aws:s3:bucket-pause-replication	101
Systems Manager Manager-Aktionen	102
aws:ssm:send-command	102
aws:ssm:start-automation-execution	103
AWS Direct Connect Aktionen	104
aws:directconnect:virtual-interface-disconnect	104
Aktionen für SSM-Dokumente	105
Verwenden Sie die Aktion aws:ssm:send-command	106
Vorkonfigurierte AWS FIS SSM-Dokumente	107
Beispiele	118
Einschränkungen	118
Rollback-Skripten	118
Fehlerbehebung	120
ECS-Aufgabenaktionen	120
Aktionen	121
Einschränkungen	121
Voraussetzungen	122
Referenzversion des Skripts	125
Beispiel für eine Versuchsvorlage	127
EKS-Pod-Aktionen	128
Aktionen	129
Einschränkungen	130
Voraussetzungen	131
Erstellen Sie eine Experimentrolle	131
Das Kubernetes-Servicekonto konfigurieren	131
Gewähren Sie IAM-Benutzern und -Rollen Zugriff auf Kubernetes APIs	133
Pod-Container-Bilder	134
Beispiel für eine Versuchsvorlage	136
AWS Lambda Aktionen	137
Aktionen	138
Einschränkungen	138
Voraussetzungen	138

Konfigurieren von Lambda-Funktionen	140
Konfigurieren Sie ein AWS FIS Experiment	141
Protokollierung	141
Fortschrittliche Themen	143
AWS FIS Versionen der Lambda-Erweiterung	149
Verwaltung von Versuchsvorlagen	153
Erstellen Sie eine Versuchsvorlage	153
Vorlagen für Experimente anzeigen	156
Generieren Sie eine Zielvorschau	157
Starten Sie ein Experiment mit einer Vorlage	158
Aktualisieren Sie eine Experimentvorlage	159
Versuchs-Vorlagen mit Tags versehen	159
Löschen Sie eine Experimentvorlage	160
Beispiel--Vorlagen	160
Stoppen Sie EC2 Instanzen, die auf Filtern basieren	161
Stoppt eine bestimmte Anzahl von Instanzen EC2	162
Führen Sie ein vorkonfiguriertes AWS FIS SSM-Dokument aus	163
Führen Sie ein vordefiniertes Automatisierungs-Runbook aus	164
Drosseln Sie API-Aktionen auf EC2 Instances mit der IAM-Zielrolle	165
Stresstest der CPU von Pods in einem Kubernetes-Cluster	166
Bereitgestellte Durchsatzausnahme für die angegebene Anzahl von Kinesis Data Streams .	168
Beispiel für ein Experiment, Rollenberechtigungen	169
Verwaltung von Experimenten	171
Starten Sie ein Experiment	171
Sehen Sie sich Ihre Experimente an	172
Status des Experiments	173
Status der Aktion	173
Kennzeichnen Sie ein Experiment	174
Stoppen eines Experiments	174
Aufgelöste Ziele auflisten	175
Lernprogramme	176
Testinstanz beenden und starten	176
Voraussetzungen	176
Schritt 1: Erstellen Sie eine Versuchsvorlage	176
Schritt 2: Starten Sie das Experiment	180
Schritt 3: Verfolgen Sie den Fortschritt des Experiments	180

Schritt 4: Überprüfen Sie das Ergebnis des Experiments	181
Schritt 5: Bereinigen	181
CPU-Belastung auf einer Instanz ausführen	182
Voraussetzungen	182
Schritt 1: Erstellen Sie einen CloudWatch Alarm für eine Stopp-Bedingung	183
Schritt 2: Erstellen Sie eine Experimentvorlage	184
Schritt 3: Starten Sie das Experiment	187
Schritt 4: Verfolgen Sie den Fortschritt des Experiments	187
Schritt 5: Überprüfen Sie die Ergebnisse des Experiments	188
Schritt 6: Bereinigen	181
Testen Sie Spot-Instance-Unterbrechungen	190
Voraussetzungen	190
Schritt 1: Erstellen Sie eine Experimentvorlage	192
Schritt 2: Starten Sie das Experiment	194
Schritt 3: Verfolgen Sie den Fortschritt des Experiments	195
Schritt 4: Überprüfen Sie das Versuchsergebnis	195
Schritt 5: Bereinigen	196
Simulieren Sie ein Verbindungsereignis	197
Voraussetzungen	198
Schritt 1: Erstellen Sie eine AWS FIS-Versuchsvorlage	199
Schritt 2: Pingen Sie einen Amazon S3 S3-Endpunkt	200
Schritt 3: Starten Sie Ihr AWS FIS-Experiment	201
Schritt 4: Verfolgen Sie den Fortschritt Ihres AWS FIS-Experiments	202
Schritt 5: Überprüfen Sie die Amazon S3 S3-Netzwerkunterbrechung	202
Schritt 5: Bereinigen	202
Planen Sie ein wiederkehrendes Experiment	203
Voraussetzungen	204
Schritt 1: Erstellen Sie eine IAM-Rolle und -Richtlinie	204
Schritt 2: Erstellen Sie einen Amazon EventBridge Scheduler	206
Schritt 3: Verifizieren Sie Ihr Experiment	207
Schritt 4: Bereinigen	208
Arbeiten mit der Szenariobibliothek	209
Ein Szenario anzeigen	209
Verwenden eines Szenarios	210
Ein Szenario exportieren	211
Referenz zu Szenarien	211

AZ Availability: Power Interruption	215
AZ: Application Slowdown	230
Cross-AZ: Traffic Slowdown	237
Cross-Region: Connectivity	245
Arbeiten mit Experimenten mit mehreren Konten	260
Konzepte	261
Best Practices	261
Voraussetzungen	262
Berechtigungen	262
Bedingungen beenden (optional)	265
Sicherheitshebel für Experimente mit mehreren Konten (optional)	266
Erstellen Sie eine Versuchsvorlage für mehrere Konten	266
Aktualisieren Sie die Konfiguration eines Zielkontos	267
Löschen Sie eine Zielkontokonfiguration	268
Planung von Experimenten	270
Erstellen Sie eine Scheduler-Rolle	270
Erstellen Sie einen Zeitplan für Experimente	274
Um den Zeitplan über die Konsole zu aktualisieren	275
Aktualisieren Sie einen Experimentplan	276
Deaktivieren oder löschen Sie einen Experimentplan	276
Sicherheitshebel	278
Konzepte für Sicherheitshebel	278
Ressource für den Sicherheitshebel	279
Mit Sicherheitshebeln arbeiten	279
Sicherheitshebel anzeigen	279
Betätigen eines Sicherheitshebels	280
Einen Sicherheitshebel lösen	280
Überwachung von Experimenten	282
Überwachen Sie mit CloudWatch	283
Überwachen Sie AWS FIS-Experimente	284
AWS FIS-Nutzungsmetriken	284
Überwachen Sie mit EventBridge	285
Protokollierung von Experimenten	287
Berechtigungen	287
Protokollschema	287
Ziele protokollieren	289

Beispiel für Protokolldatensätze	289
Aktivieren Sie die Protokollierung der Experimente	294
Deaktivieren Sie die Protokollierung der Experimente	295
API-Aufrufe protokollieren mit AWS CloudTrail	296
Benutzen CloudTrail	296
Verstehen Sie die Einträge in AWS der FIS-Protokolldatei	297
Fehlerbehebung	302
Fehlercodes	302
Sicherheit	305
Datenschutz	305
Verschlüsselung im Ruhezustand	307
Verschlüsselung während der Übertragung	307
Identity and Access Management	307
Zielgruppe	308
Authentifizierung mit Identitäten	308
Verwalten des Zugriffs mit Richtlinien	310
So funktioniert der AWS Fault Injection Service mit IAM	311
Beispiele für Richtlinien	317
Serviceverknüpfte Rollen verwenden	328
AWS verwaltete Richtlinien	331
Sicherheit der Infrastruktur	336
AWS PrivateLink	337
Überlegungen	337
Erstellen eines Schnittstellen-VPC-Endpunkts	337
Erstellen einer VPC-Endpunktrichtlinie	337
Markieren Ihrer -Ressourcen	340
Tagging-Einschränkungen	340
Arbeite mit Tags	340
Kontingente und Einschränkungen	342
Dokumentverlauf	358
.....	ccclxvii

Was ist der AWS Fault Injection Service?

AWS Der AWS Fault Injection Service (FIS) ist ein verwalteter Service, mit dem Sie Fault-Injection-Experimente an Ihren AWS Workloads durchführen können. Die Fehlerinjektion basiert auf den Prinzipien der Chaos-Technik. Diese Experimente stressen eine Anwendung, indem sie störende Ereignisse erzeugen, sodass Sie beobachten können, wie Ihre Anwendung reagiert. Sie können diese Informationen dann verwenden, um die Leistung und Stabilität Ihrer Anwendungen zu verbessern, sodass sie sich wie erwartet verhalten.

Um AWS FIS zu verwenden, müssen Sie Experimente einrichten und durchführen, die Ihnen helfen, die realen Bedingungen zu schaffen, die erforderlich sind, um Anwendungsprobleme aufzudecken, die sonst schwer zu finden sind. AWS FIS bietet Vorlagen, die zu Störungen führen, sowie die Kontrollen und Leitplanken, die Sie für die Durchführung von Experimenten in der Produktion benötigen, z. B. das automatische Zurücksetzen oder Stoppen des Experiments, wenn bestimmte Bedingungen erfüllt sind.

Important

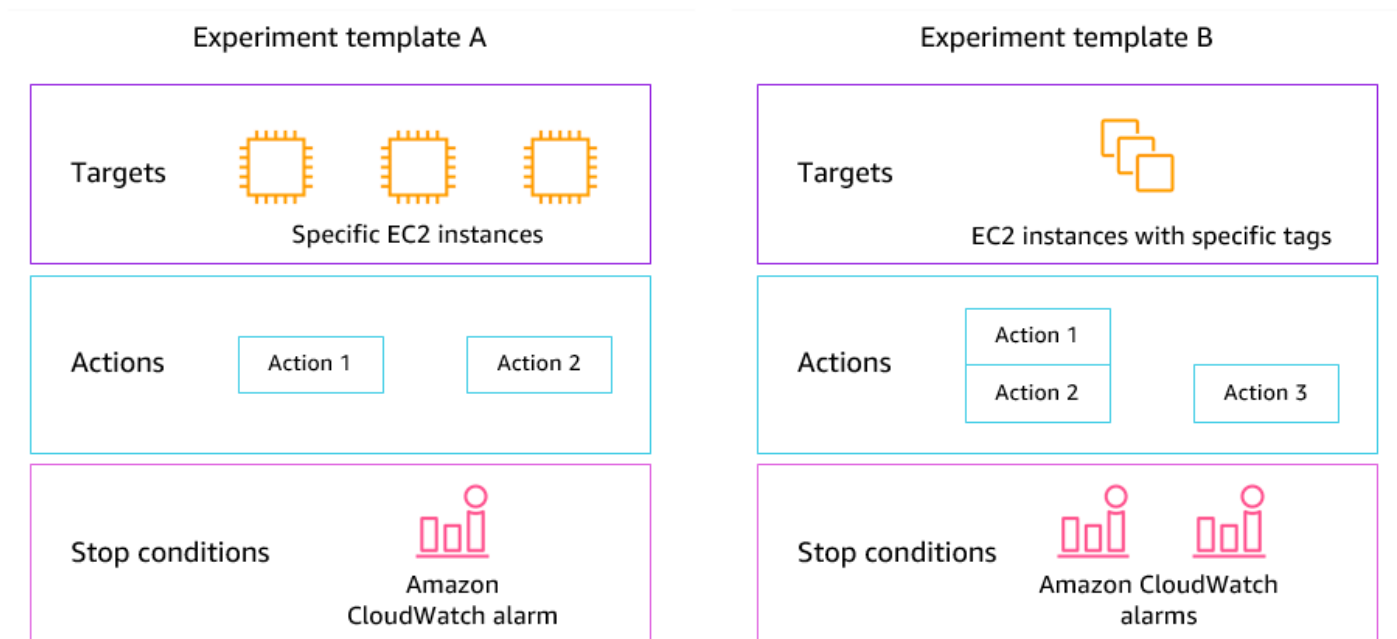
AWS FIS führt echte Aktionen an realen Ressourcen in Ihrem System durch. AWS Bevor Sie AWS FIS zur Durchführung von Experimenten in der Produktion verwenden, empfehlen wir Ihnen daher dringend, eine Planungsphase abzuschließen und die Experimente in einer Vorproduktionsumgebung durchzuführen.

Weitere Informationen zur Planung Ihres Experiments finden Sie unter [Testzuverlässigkeit und Planung Ihrer AWS FIS-Experimente](#) Weitere Informationen zu AWS FIS finden Sie unter [AWS Fault Injection Service](#).

AWS FIS-Konzepte

Um AWS FIS zu verwenden, führen Sie Experimente mit Ihren AWS Ressourcen durch, um Ihre Theorie zu testen, wie sich eine Anwendung oder ein System unter Fehlerbedingungen verhalten wird. Um Experimente durchzuführen, erstellen Sie zunächst eine Experimentvorlage. Eine Experimentvorlage ist der Bauplan Ihres Experiments. Sie enthält die Aktionen, Ziele und Stoppbedingungen für das Experiment. Nachdem Sie eine Versuchsvorlage erstellt haben, können Sie sie verwenden, um ein Experiment durchzuführen. Während Ihr Experiment läuft, können Sie

seinen Fortschritt verfolgen und seinen Status einsehen. Ein Experiment ist abgeschlossen, wenn alle Aktionen des Experiments ausgeführt wurden.



Aktionen

Eine Aktion ist eine Aktivität, die AWS FIS während eines Experiments an einer AWS Ressource durchführt. AWS FIS bietet eine Reihe von vorkonfigurierten Aktionen, die auf dem Ressourcentyp basieren. AWS Jede Aktion wird während eines Experiments für eine bestimmte Dauer ausgeführt oder bis Sie das Experiment beenden. Aktionen können sequentiell oder gleichzeitig (parallel) ausgeführt werden.

Targets (Ziele)

Ein Ziel ist eine oder mehrere AWS Ressourcen, auf denen AWS FIS während eines Experiments eine Aktion ausführt. Sie können bestimmte Ressourcen oder eine Gruppe von Ressourcen anhand bestimmter Kriterien wie Tags oder Status auswählen.

Bedingungen beenden

AWS FIS bietet die Kontrollen und Leitplanken, die Sie für die sichere Durchführung von Experimenten an Ihren Workloads benötigen. AWS Eine Stoppbedingung ist ein Mechanismus, um ein Experiment zu beenden, wenn es einen Schwellenwert erreicht, den Sie als CloudWatch Amazon-Alarm definieren. Wenn während des Experiments eine Stopp-Bedingung ausgelöst wird, stoppt AWS FIS das Experiment.

Unterstützt AWS-Services

AWS FIS bietet dienstübergreifend AWS vorkonfigurierte Aktionen für bestimmte Zieltypen. Eine Liste der unterstützten Dienste und ihrer Aktionen finden Sie in der Referenz zu [AWS FIS-Aktionen](#).

Bei Experimenten mit einem einzigen Konto müssen sich die Zielressourcen im selben Umfang AWS-Konto wie das Experiment befinden. Mithilfe von AWS FIS-Experimenten mit mehreren Konten können Sie AWS FIS-Experimente durchführen, die auf Ressourcen in einem anderen AWS-Konto Konto abzielen.

Weitere Informationen finden Sie unter [Aktionen für AWS FIS](#).

Greifen AWS Sie auf FIS zu

Sie können auf AWS eine der folgenden Arten mit FIS arbeiten:

- **AWS-Managementkonsole**— Stellt eine Weboberfläche bereit, über die Sie auf AWS FIS zugreifen können. Weitere Informationen finden Sie unter [Arbeiten mit der AWS-Managementkonsole](#).
- **AWS Command Line Interface (AWS CLI)** — Stellt Befehle für eine Vielzahl von AWS Diensten bereit, einschließlich AWS FIS, und wird unter Windows, macOS und Linux unterstützt. Weitere Informationen finden Sie unter [AWS Command Line Interface](#). Weitere Informationen zu den Befehlen für AWS FIS finden Sie unter [fis](#) in der AWS CLI Befehlsreferenz.
- **AWS CloudFormation**— Erstellen Sie Vorlagen, die Ihre AWS Ressourcen beschreiben. Mit den Vorlagen können Sie diese Ressourcen als Einheit bereitstellen und verwalten. Weitere Informationen finden Sie in der [Referenz zum Ressourcentyp des AWS Fault Injection Service](#).
- **AWS SDKs**— Stellt sprachspezifische Informationen bereit APIs und kümmert sich um viele Verbindungsdetails, wie z. B. die Berechnung von Signaturen, die Behandlung von Wiederholungsversuchen von Anfragen und die Behandlung von Fehlern. Weitere Informationen finden Sie unter [AWS SDKs](#).
- **HTTPS-API** — Stellt API-Aktionen auf niedriger Ebene bereit, die Sie mithilfe von HTTPS-Anfragen aufrufen können. Weitere Informationen finden Sie in der [AWS Fault Injection Service API-Referenz](#).

Preise für AWS FIS

Ihnen wird pro Minute berechnet, die eine Aktion von Anfang bis Ende ausgeführt wird, basierend auf der Anzahl der Zielkonten für Ihr Experiment. Weitere Informationen finden Sie unter [AWS FIS-Preise](#).

Planung Ihrer AWS FIS-Experimente

Bei der Fehlerinjektion wird eine Anwendung in Test- oder Produktionsumgebungen durch störende Ereignisse wie Serverausfälle oder API-Drosselung belastet. Wenn Sie beobachten, wie das System reagiert, können Sie dann Verbesserungen umsetzen. Wenn Sie Experimente an Ihrem System durchführen, kann es Ihnen helfen, systemische Schwächen kontrolliert zu identifizieren, bevor sich diese Schwächen auf die Kunden auswirken, die von Ihrem System abhängig sind. Anschließend können Sie die Probleme proaktiv angehen, um unvorhersehbare Ergebnisse zu vermeiden.

Bevor Sie mit der Durchführung von Experimenten zur Fehlerinjektion mit AWS FIS beginnen, empfehlen wir Ihnen, sich mit den folgenden Prinzipien und Richtlinien vertraut zu machen.

Important

AWS FIS führt echte Aktionen an realen AWS Ressourcen in Ihrem System durch. Bevor Sie beginnen, AWS FIS zur Durchführung von Experimenten zu verwenden, empfehlen wir Ihnen daher dringend, zunächst eine Planungsphase und einen Test in einer Vorproduktions- oder Testumgebung abzuschließen.

Inhalt

- [Grundprinzipien und Richtlinien](#)
- [Richtlinien für die Planung von Experimenten](#)

Grundprinzipien und Richtlinien

Bevor Sie mit Experimenten mit AWS FIS beginnen, gehen Sie wie folgt vor:

1. Identifizieren Sie den Zieleinsatz für das Experiment — Identifizieren Sie zunächst den Zieleinsatz. Wenn dies Ihr erstes Experiment ist, empfehlen wir, in einer Vorproduktions- oder Testumgebung zu beginnen.
2. Überprüfen Sie die Anwendungsarchitektur — Sie müssen sicherstellen, dass Sie alle Anwendungskomponenten, Abhängigkeiten und Wiederherstellungsverfahren für jede Komponente identifiziert haben. Beginnen Sie mit der Überprüfung der Anwendungsarchitektur. Je nach Anwendung beziehen Sie sich auf das [AWS Well-Architected Framework](#).

3. Definieren Sie das stationäre Verhalten Ihres Systems — Definieren Sie das stationäre Verhalten Ihres Systems anhand wichtiger technischer und geschäftlicher Kennzahlen wie Latenz, CPU-Last, fehlgeschlagene Anmeldungen pro Minute, Anzahl der Wiederholungen oder Seitenladegeschwindigkeit.
4. Stellen Sie eine Hypothese auf — Stellen Sie eine Hypothese darüber auf, wie sich das Systemverhalten während des Experiments voraussichtlich ändern wird. Eine Hypothesendefinition folgt diesem Format:

Wenn durchgeführt *fault injection action* wird, *business or technical metric impact* sollte der nicht überschritten *value* werden.

Eine Hypothese für einen Authentifizierungsdienst könnte beispielsweise wie folgt lauten: „Wenn die Netzwerklatenz um 10% zunimmt, nimmt die Anzahl der Anmeldefehler um weniger als 1% zu.“ Nach Abschluss des Experiments bewerten Sie, ob die Ausfallsicherheit der Anwendung Ihren geschäftlichen und technischen Erwartungen entspricht.

Wir empfehlen außerdem, bei der Arbeit mit AWS FIS die folgenden Richtlinien zu beachten:

- Beginne immer, mit AWS FIS in einer Testumgebung zu experimentieren. Beginnen Sie niemals mit einer Produktionsumgebung. Wenn Sie mit Ihren Experimenten zur Fehlerinjektion vorankommen, können Sie auch in anderen kontrollierten Umgebungen außerhalb der Testumgebung experimentieren.
- Stärken Sie das Vertrauen Ihres Teams in die Resilienz Ihrer Anwendung, indem Sie mit kleinen, einfachen Experimenten beginnen, wie z. B. der Ausführung der Aktion `aws:ec2:stop-instances` auf einem Ziel.
- Die Fehlerinjektion kann zu echten Problemen führen. Gehen Sie vorsichtig vor und stellen Sie sicher, dass Ihre ersten Fehlerinjektionen auf Testinstanzen erfolgen, damit keine Kunden betroffen sind.
- Testen, testen und noch mehr testen. Die Fehlerinjektion soll in einer kontrollierten Umgebung mit gut geplanten Experimenten implementiert werden. Auf diese Weise können Sie Vertrauen in die Fähigkeiten Ihrer Anwendung und Ihrer Tools aufbauen, um turbulenten Bedingungen standzuhalten.
- Wir empfehlen dringend, dass Sie über ein hervorragendes Überwachungs- und Warnprogramm verfügen, bevor Sie beginnen. Ohne dieses Programm werden Sie nicht in der Lage sein, die Auswirkungen Ihrer Experimente zu verstehen oder zu messen, was für nachhaltige Verfahren zur Fehlerinjektion von entscheidender Bedeutung ist.

Richtlinien für die Planung von Experimenten

Mit AWS FIS führen Sie Experimente mit Ihren AWS Ressourcen durch, um Ihre Theorie zu testen, wie sich eine Anwendung oder ein System unter Fehlerbedingungen verhalten wird.

Im Folgenden finden Sie empfohlene Richtlinien für die Planung Ihrer AWS FIS-Experimente.

- **Ausfallverlauf überprüfen** — Sehen Sie sich die vorherigen Ausfälle und Ereignisse für Ihr System an. Dies kann Ihnen helfen, sich ein Bild vom allgemeinen Zustand und der Widerstandsfähigkeit Ihres Systems zu machen. Bevor Sie mit der Durchführung von Experimenten auf Ihrem System beginnen, sollten Sie sich mit bekannten Problemen und Schwächen in Ihrem System befassen.
- **Identifizieren Sie die Dienste mit den größten Auswirkungen** — Überprüfen Sie Ihre Dienste und identifizieren Sie diejenigen, die die größten Auswirkungen auf Ihre Endbenutzer oder Kunden haben, wenn sie ausfallen oder nicht richtig funktionieren.
- **Identifizieren Sie das Zielsystem** — Das Zielsystem ist das System, auf dem Sie Experimente durchführen werden. Wenn Sie mit FIS noch nicht vertraut sind oder noch nie zuvor AWS Fault-Injection-Experimente durchgeführt haben, empfehlen wir Ihnen, zunächst Experimente auf einem Vorproduktions- oder Testsystem durchzuführen.
- **Konsultieren Sie Ihr Team** — Fragen Sie, worüber es sich Sorgen macht. Sie können eine Hypothese aufstellen, um ihre Bedenken zu beweisen oder zu widerlegen. Sie können Ihr Team auch fragen, worüber es sich keine Sorgen macht. Diese Frage kann zwei weit verbreitete Irrtümer aufdecken: den Sunk-Cost-Irrtum und den Confirmation Bias-Irrtum. Die Aufstellung einer Hypothese auf der Grundlage der Antworten Ihres Teams kann dazu beitragen, mehr Informationen über den tatsächlichen Zustand Ihres Systems zu erhalten.
- **Überprüfen Sie Ihre Anwendungsarchitektur** — Führen Sie eine Überprüfung Ihres Systems oder Ihrer Anwendung durch und stellen Sie sicher, dass Sie alle Anwendungskomponenten, Abhängigkeiten und Wiederherstellungsverfahren für jede Komponente identifiziert haben.

Wir empfehlen Ihnen, das AWS Well-Architected Framework zu lesen. Das Framework kann Ihnen helfen, eine sichere, leistungsstarke, belastbare und effiziente Infrastruktur für Ihre Anwendungen und Workloads aufzubauen. Weitere Informationen finden Sie unter [AWS Well-Architected](#).

- **Identifizieren Sie die zutreffenden Kennzahlen** — Sie können die Auswirkungen eines Experiments auf Ihre AWS Ressourcen mithilfe von CloudWatch Amazon-Metriken überwachen. Sie können diese Kennzahlen verwenden, um den Ausgangswert oder den „Steady-State“ zu ermitteln, zu dem Ihre Anwendung optimal funktioniert. Anschließend können Sie diese Messwerte während oder nach dem Experiment überwachen, um die Auswirkungen zu ermitteln. Weitere Informationen finden Sie unter [Überwachen Sie die AWS FIS-Nutzungsmetriken mit Amazon CloudWatch](#).

- Definieren Sie einen akzeptablen Leistungsschwellenwert für Ihr System — Identifizieren Sie die Metrik, die einen akzeptablen, stabilen Zustand für Ihr System darstellt. Sie verwenden diese Metrik, um einen oder mehrere CloudWatch Alarme zu erstellen, die eine Stoppbedingung für Ihr Experiment darstellen. Wenn der Alarm ausgelöst wird, wird das Experiment automatisch gestoppt. Weitere Informationen finden Sie unter [Stoppbedingungen für AWS FIS](#).

AWS Komponenten der FIS-Versuchsvorlage

Sie verwenden die folgenden Komponenten, um Versuchsvorlagen zu erstellen:

Aktionen

Die [AWS FIS-Aktionen](#), die Sie ausführen möchten. Aktionen können in einer festgelegten Reihenfolge ausgeführt werden, die Sie angeben, oder sie können gleichzeitig ausgeführt werden. Weitere Informationen finden Sie unter [Aktionen](#).

Ziele

Die AWS Ressourcen, auf denen eine bestimmte Aktion ausgeführt wird. Weitere Informationen finden Sie unter [Ziele](#).

Bedingungen beenden

Die CloudWatch Alarmer, die einen Schwellenwert definieren, bei dem die Leistung Ihrer Anwendung nicht akzeptabel ist. Wenn während der Ausführung eines Experiments eine Stopp-Bedingung ausgelöst wird, stoppt AWS FIS das Experiment. Weitere Informationen finden Sie unter [Bedingungen beenden](#).

Rolle des Experiments

Eine IAM-Rolle, die AWS FIS die erforderlichen Berechtigungen erteilt, damit sie Experimente in Ihrem Namen durchführen kann. Weitere Informationen finden Sie unter [Die Rolle des Experiments](#).

Konfiguration des Versuchsberichts

Die Konfiguration zur Aktivierung von Experimentberichten. Weitere Informationen finden Sie unter [Konfigurationen von Experimentberichten für AWS FIS](#).

Optionen für Experimente

Optionen für die Experimentvorlage. Weitere Informationen finden Sie unter [Experimentieroptionen für AWS FIS](#).

Ihr Konto hat Kontingente für AWS FIS. Beispielsweise gibt es ein Kontingent für die Anzahl der Aktionen pro Versuchsvorlage. Weitere Informationen finden Sie unter [Kontingente und Einschränkungen](#).

Syntax der Vorlage

Das Folgende ist die Syntax für eine Experimentvorlage.

```
{
    "description": "string",
    "targets": {},
    "actions": {},
    "stopConditions": [],
    "roleArn": "arn:aws:iam::123456789012:role/AllowFISActions",
    "experimentReportConfiguration": {},
    "experimentOptions": {},
    "tags": {}
}
```

Beispiele finden Sie unter [Beispiel--Vorlagen](#).

Erste Schritte

Informationen zum Erstellen einer Versuchsvorlage mit dem AWS-Managementkonsole finden Sie unter [Erstellen Sie eine Versuchsvorlage](#).

Informationen zum Erstellen einer Versuchsvorlage mit dem AWS CLI finden Sie unter [Beispiele für AWS FIS-Experimentvorlagen](#).

Aktionen für AWS FIS

Um eine Versuchsvorlage zu erstellen, müssen Sie eine oder mehrere Aktionen definieren. Eine Liste der von AWS FIS bereitgestellten vordefinierten Aktionen finden Sie unter [Referenz zu Aktionen](#).

Sie können eine Aktion während eines Experiments nur einmal ausführen. Um dieselbe AWS FIS-Aktion mehrmals in demselben Experiment auszuführen, fügen Sie sie der Vorlage mehrmals unter unterschiedlichen Namen hinzu.

Inhalt

- [Syntax der Aktion](#)
- [Aktions-Identifikatoren](#)
- [Aktionsparameter](#)
- [Ziele der Aktion](#)

- [Dauer der Aktion](#)
- [Beispielaktionen](#)

Syntax der Aktion

Das Folgende ist die Syntax für eine Aktion.

```
{
  "actions": {
    "action_name": {
      "actionId": "aws:service:action-type",
      "description": "string",
      "parameters": {
        "name": "value"
      },
      "startAfter": ["action_name", ...],
      "targets": {
        "ResourceType": "target_name"
      }
    }
  }
}
```

Wenn Sie eine Aktion definieren, geben Sie Folgendes an:

action_name

Ein Name für die Aktion.

actionId

Die [Aktions-ID](#).

description

Eine optionale Beschreibung.

parameters

Beliebige [Aktionsparameter](#).

startAfter

Alle Aktionen, die abgeschlossen sein müssen, bevor diese Aktion gestartet werden kann. Andernfalls wird die Aktion zu Beginn des Experiments ausgeführt.

targets

Beliebige [Aktionsziele](#).

Beispiele finden Sie unter [the section called “Beispielaktionen”](#).

Aktions-Identifikatoren

Jede AWS FIS-Aktion hat eine Kennung mit dem folgenden Format:

```
aws:service-name:action-type
```

Mit der folgenden Aktion werden beispielsweise die Amazon EC2 EC2-Ziel-Instances gestoppt:

```
aws:ec2:stop-instances
```

Eine vollständige Liste der Aktionen finden Sie unter [AWS FIS Referenz zu Aktionen](#).

Aktionsparameter

Einige AWS FIS-Aktionen haben zusätzliche Parameter, die für die Aktion spezifisch sind. Diese Parameter werden verwendet, um Informationen an AWS FIS zu übergeben, wenn die Aktion ausgeführt wird.

AWS FIS unterstützt benutzerdefinierte Fehlertypen mithilfe der `aws:ssm:send-command` Aktion, bei der der SSM-Agent und ein SSM-Befehlsdokument verwendet werden, um den Fehlerzustand auf den Zielinstanzen zu erstellen. Die `aws:ssm:send-command` Aktion umfasst einen `documentArn` Parameter, der den Amazon-Ressourcennamen (ARN) eines SSM-Dokuments als Wert verwendet. Sie geben Werte für Parameter an, wenn Sie die Aktion zu Ihrer Experimentvorlage hinzufügen.

Weitere Informationen zum Angeben von Parametern für die `aws:ssm:send-command` Aktion finden Sie unter [Verwenden Sie die Aktion aws:ssm:send-command](#).

Wenn möglich, können Sie in die Aktionsparameter eine Rollback-Konfiguration (auch als Post-Aktion bezeichnet) eingeben. Eine Post-Aktion führt das Ziel in den Zustand zurück, in dem es sich vor Ausführung der Aktion befunden hat. Die Post-Aktion wird nach der in der Aktionsdauer angegebenen Zeit ausgeführt. Nicht alle Aktionen können Post-Aktionen unterstützen. Wenn die Aktion beispielsweise eine Amazon EC2 EC2-Instance beendet, können Sie die Instance nicht wiederherstellen, nachdem sie beendet wurde.

Ziele der Aktion

Eine Aktion wird auf den von Ihnen angegebenen Zielressourcen ausgeführt. Nachdem Sie ein Ziel definiert haben, können Sie seinen Namen angeben, wenn Sie eine Aktion definieren.

```
"targets": {  
  "ResourceType": "resource_name"  
}
```

AWS FIS-Aktionen unterstützen die folgenden Ressourcentypen für Aktionsziele:

- AutoScalingGroups— Amazon EC2 Auto Scaling Scaling-Gruppen
- Eimer — Amazon S3 S3-Eimer
- Cluster — Amazon EKS-Cluster
- Cluster — Amazon ECS-, Aurora DSQL- oder Amazon Aurora Aurora-DB-Cluster
- DBInstances— Amazon RDS-DB-Instances
- Funktionen — AWS Lambda Funktionen
- Instanzen — Amazon EC2 EC2-Instances
- KinesisStreams— Kinesis-Datenströme
- ManagedResources— Amazon EKS-Cluster, Amazon EC2 Application and Network Load Balancers und Amazon EC2 Auto Scaling Scaling-Gruppen, die für ARC-Zonenverschiebung aktiviert sind.
- MultiRegionClusters— Amazon MemoryDB-Cluster mit mehreren Regionen
- Knotengruppen — Amazon EKS-Knotengruppen
- Pods — Kubernetes-Pods auf Amazon EKS
- ReplicationGroups— Replikationsgruppen ElastiCache
- Rollen — IAM-Rollen
- SpotInstances— Amazon EC2-Spot-Instances
- Subnetze — VPC-Subnetze
- Tabellen — Amazon DynamoDB Starke und letztlich konsistente globale Tabellen mit mehreren Regionen
- Aufgaben — Amazon ECS-Aufgaben

- TransitGateways— Transit-Gateways
- VirtualInterfaces— Direct Connect Virtuelle Schnittstellen
- Volumen — Amazon EBS-Volumen
- VPCEndpoints— Amazon VPC-Endpunkte

Beispiele finden Sie unter [the section called “Beispielaktionen”](#).

Dauer der Aktion

Wenn eine Aktion einen Parameter enthält, mit dem Sie die Dauer der Aktion angeben können, wird die Aktion standardmäßig erst nach Ablauf der angegebenen Dauer als abgeschlossen betrachtet. Wenn Sie die `emptyTargetResolutionMode Experimentoption` auf `gesetzt habenskip` gesetzt haben, wird die Aktion sofort mit dem Status „übersprungen“ abgeschlossen, wenn keine Ziele gelöst wurden. Wenn Sie beispielsweise eine Dauer von 5 Minuten angeben, betrachtet AWS FIS die Aktion nach 5 Minuten als abgeschlossen. Anschließend wird die nächste Aktion gestartet, bis alle Aktionen abgeschlossen sind.

Die Dauer kann entweder der Zeitraum sein, für den eine Aktionsbedingung aufrechterhalten wird, oder der Zeitraum, für den Messwerte überwacht werden. Beispielsweise wird die Latenz für die angegebene Zeitdauer injiziert. Bei Aktionstypen, die sich fast augenblicklich ergeben, z. B. beim Beenden einer Instance, werden die Stoppbedingungen für den angegebenen Zeitraum überwacht.

Wenn eine Aktion innerhalb der Aktionsparameter eine Post-Aktion beinhaltet, wird die Post-Aktion nach Abschluss der Aktion ausgeführt. Die Zeit, die benötigt wird, um die Post-Aktion abzuschließen, kann zu einer Verzögerung zwischen der angegebenen Aktionsdauer und dem Beginn der nächsten Aktion (oder dem Ende des Experiments, wenn alle anderen Aktionen abgeschlossen sind) führen.

Beispielaktionen

Im Folgenden finden Sie Beispielaktionen.

Beispiele

- [Stoppen Sie EC2-Instances](#)
- [Spot-Instances unterbrechen](#)
- [Unterbrechen Sie den Netzwerkverkehr](#)
- [Kündigen Sie EKS-Mitarbeiter](#)

- [Starten Sie ARC Zonal Autoshift](#)

Beispiel: Stoppen Sie EC2-Instanzen

Die folgende Aktion stoppt die EC2-Instances, die mithilfe des genannten Ziels identifiziert wurden.

targetInstances Nach zwei Minuten werden die Ziel-Instances neu gestartet.

```
"actions": {
  "stopInstances": {
    "actionId": "aws:ec2:stop-instances",
    "parameters": {
      "startInstancesAfterDuration": "PT2M"
    },
    "targets": {
      "Instances": "targetInstances"
    }
  }
}
```

Beispiel: Spot-Instances unterbrechen

Die folgende Aktion stoppt die Spot-Instances, die mithilfe des genannten *targetSpotInstances* Ziels identifiziert wurden. Sie wartet zwei Minuten, bevor sie die Spot-Instance unterbricht.

```
"actions": {
  "interruptSpotInstances": {
    "actionId": "aws:ec2:send-spot-instance-interruptions",
    "parameters": {
      "durationBeforeInterruption": "PT2M"
    },
    "targets": {
      "SpotInstances": "targetSpotInstances"
    }
  }
}
```

Beispiel: Den Netzwerkverkehr unterbrechen

Die folgende Aktion verweigert den Verkehr zwischen den Zielsubnetzen und Subnetzen in anderen Availability Zones.

```

"actions": {
  "disruptAZConnectivity": {
    "actionId": "aws:network:disrupt-connectivity",
    "parameters": {
      "scope": "availability-zone",
      "duration": "PT5M"
    },
    "targets": {
      "Subnets": "targetSubnets"
    }
  }
}

```

Beispiel: Kündigen Sie EKS-Mitarbeiter

Die folgende Aktion beendet 50% der EC2-Instances im EKS-Cluster, die anhand des genannten Ziels identifiziert wurden. *targetNodeGroups*

```

"actions": {
  "terminateWorkers": {
    "actionId": "aws:eks:terminate-nodegroup-instances",
    "parameters": {
      "instanceTerminationPercentage": "50"
    },
    "targets": {
      "Nodegroups": "targetNodeGroups"
    }
  }
}

```

Beispiel: Starten Sie ARC Zonal Autoshift

Mit der folgenden Aktion wird ein ARC-Zonal-Autoshift gestartet, der verwaltete Ressourcen weg von for verschiebt. *az-in-parameters duration-in-parameteres* Der Ressourcentyp ManagedResources wird als Schlüssel für den Zielnamen in der AWS FIS-Experimentvorlage verwendet.

```

{
  "description": "aaa",

```

```

"targets": {
  "ManagedResources-Target-1": {
    "resourceType": "aws:arc:zonal-shift-managed-resource",
    "resourceArns": [
      "arn:aws:elasticloadbalancing:us-east-1:0124567890:loadbalancer/app/
application/11223312312516",
    ],
    "selectionMode": "ALL"
  }
},
"actions": {
  "arc": {
    "actionId": "aws:arc:start-zonal-autoshift",
    "parameters": {
      "availabilityZoneIdentifier": "us-east-1a",
      "duration": "PT1M"
    },
    "targets": {
      "ManagedResources": "ManagedResources-Target-1"
    }
  }
},
"stopConditions": [
  {
    "source": "none"
  }
],
"roleArn": "arn:aws:iam::718579638765:role/fis",
"tags": {},
"experimentOptions": {
  "accountTargeting": "single-account",
  "emptyTargetResolutionMode": "fail"
}
}

```

Ziele für AWS FIS

Ein Ziel ist eine oder mehrere AWS Ressourcen, für die der AWS AWS Fault Injection Service (FIS) während eines Experiments eine Aktion ausführt. Ziele können sich in demselben AWS-Konto wie das Experiment oder in einem anderen Konto befinden, wenn ein Experiment mit mehreren Konten verwendet wird. Weitere Informationen zum Targeting von Ressourcen in einem anderen Konto finden Sie unter [Arbeiten mit Experimenten mit mehreren Konten](#).

Sie definieren Ziele, wenn Sie [eine Versuchsvorlage erstellen](#). Sie können dasselbe Ziel für mehrere Aktionen in Ihrer Experimentvorlage verwenden.

AWS FIS identifiziert alle Ziele zu Beginn des Experiments, bevor eine der Aktionen im Aktionssatz gestartet wird. AWS FIS verwendet die Zielressourcen, die es für das gesamte Experiment auswählt. Wenn keine Ziele gefunden werden, schlägt das Experiment fehl.

Inhalt

- [Zielsyntax](#)
- [Ressourcentypen](#)
- [Identifizieren Sie die Zielressourcen](#)
 - [Ressourcenfilter](#)
 - [Parameter für Ressourcen](#)
- [Auswahlmodus](#)
- [Beispielziele](#)
- [Beispielfilter](#)

Zielsyntax

Das Folgende ist die Syntax für ein Ziel.

```
{
  "targets": {
    "target_name": {
      "resourceType": "resource-type",
      "resourceArns": [
        "resource-arn"
      ],
      "resourceTags": {
        "tag-key": "tag-value"
      },
      "parameters": {
        "parameter-name": "parameter-value"
      },
      "filters": [
        {
          "path": "path-string",
          "values": ["value-string"]
        }
      ]
    }
  }
}
```

```
        }
      ],
      "selectionMode": "value"
    }
  }
}
```

Wenn Sie ein Ziel definieren, geben Sie Folgendes an:

target_name

Ein Name für das Ziel.

resourceType

Der [Ressourcentyp](#).

resourceArns

Die Amazon-Ressourcennamen (ARN) bestimmter Ressourcen.

resourceTags

Die Tags, die auf bestimmte Ressourcen angewendet wurden.

parameters

Die [Parameter](#), die Ziele anhand bestimmter Attribute identifizieren.

filters

Der [Ressourcenfilter erfasst](#) den Geltungsbereich der identifizierten Zielressourcen anhand bestimmter Attribute.

selectionMode

Der [Auswahlmodus](#) für die identifizierten Ressourcen.

Beispiele finden Sie unter [the section called "Beispielziele"](#).

Ressourcentypen

Jede AWS FIS-Aktion wird für einen bestimmten AWS Ressourcentyp ausgeführt. Wenn Sie ein Ziel definieren, müssen Sie genau einen Ressourcentyp angeben. Wenn Sie ein Ziel für eine Aktion

angeben, muss es sich bei dem Ziel um den Ressourcentyp handeln, der von der Aktion unterstützt wird.

Die folgenden Ressourcentypen werden von AWS FIS unterstützt:

- `aws:arc: zonal-shift-managed-resource` — Eine AWS Ressource, die bei ARC Zonal Shift registriert ist
- `aws:directconnect:virtual-interface` — Eine virtuelle Schnittstelle Direct Connect
- `aws:dsql:cluster` — Ein Amazon Aurora DSQL-Cluster
- `aws:dynamodb:global-table` — Eine globale Amazon DynamoDB-Tabelle mit mehreren Regionen
- `aws:ec2:autoscaling-group` — Eine Amazon Auto Scaling Scaling-Gruppe EC2
- `aws:ec2:ebs-volume` — Ein Amazon EBS-Volume
- `aws:ec2:instance` — Eine Amazon-Instance EC2
- `aws:ec2:spot-instance` — Eine Amazon Spot-Instance EC2
- `aws:ec2:subnet` — Ein Amazon VPC-Subnetz
- `aws:ec2:transit-gateway` — Ein Transit-Gateway
- `aws:ec2:vpc-endpoint` — Ein Amazon VPC-Endpunkt
- `aws:ecs:cluster` — Ein Amazon ECS-Cluster
- `aws:ecs:task` — Eine Amazon ECS-Aufgabe
- `aws:eks:cluster` — Ein Amazon EKS-Cluster
- `aws:eks:nodegroup` — Eine Amazon EKS-Knotengruppe
- `aws:eks:pod` — Ein Kubernetes-Pod
- `aws:elasticache:replicationgroup` ElastiCache — Eine Replikationsgruppe
- `aws:iam:role` — Eine IAM-Rolle
- `aws:kinesis:stream` — Ein Amazon Kinesis Kinesis-Datenstream
- `aws:lambda:function` AWS Lambda — Eine Funktion
- `aws:memorydb: multi-region-cluster` — Ein Amazon MemoryDB-Cluster mit mehreren Regionen
- `aws:rds:cluster` — Ein Amazon Aurora Aurora-DB-Cluster
- `aws:rds:db` — Eine Amazon RDS-DB-Instance
- `aws:s3:bucket` — Ein Amazon S3 S3-Bucket

Identifizieren Sie die Zielressourcen

Wenn Sie in der AWS FIS-Konsole ein Ziel definieren, können Sie bestimmte AWS Ressourcen (eines bestimmten Ressourcentyps) als Ziel auswählen. Oder Sie können AWS FIS anhand der von Ihnen angegebenen Kriterien eine Gruppe von Ressourcen identifizieren lassen.

Um Ihre Zielressourcen zu identifizieren, können Sie Folgendes angeben:

- **Ressource IDs** — Die Ressource IDs bestimmter AWS Ressourcen. Alle Ressourcen IDs müssen denselben Ressourcentyp repräsentieren.
- **Ressourcen-Tags** — Die Tags, die auf bestimmte AWS Ressourcen angewendet wurden.
- **Ressourcenfilter** — Der Pfad und die Werte, die Ressourcen mit bestimmten Attributen repräsentieren. Weitere Informationen finden Sie unter [Ressourcenfilter](#).
- **Ressourcenparameter** — Die Parameter, die Ressourcen darstellen, die bestimmte Kriterien erfüllen. Weitere Informationen finden Sie unter [Parameter für Ressourcen](#).

Überlegungen

- Sie können nicht sowohl eine Ressourcen-ID als auch ein Ressourcen-Tag für dasselbe Ziel angeben.
- Sie können nicht sowohl eine Ressourcen-ID als auch einen Ressourcenfilter für dasselbe Ziel angeben.
- Wenn Sie ein Ressourcen-Tag mit einem leeren Tag-Wert angeben, entspricht dies nicht einem Platzhalter. Es entspricht Ressourcen, die ein Tag mit dem angegebenen Tag-Schlüssel und einem leeren Tag-Wert haben.
- Wenn Sie mehr als ein Tag angeben, müssen alle angegebenen Tags auf der Zielressource vorhanden sein, damit sie ausgewählt werden kann (AND).

Ressourcenfilter

Ressourcenfilter sind Abfragen, die Zielressourcen anhand bestimmter Attribute identifizieren. AWS FIS wendet die Abfrage auf die Ausgabe einer API-Aktion an, die die kanonische Beschreibung der AWS Ressource entsprechend dem von Ihnen angegebenen Ressourcentyp enthält. Ressourcen mit Attributen, die der Abfrage entsprechen, sind in der Zieldefinition enthalten.

Jeder Filter wird als Attributpfad und mögliche Werte ausgedrückt. Ein Pfad ist eine durch Punkte getrennte Folge von Elementen, die in der Ausgabe der Aktion „Beschreiben“ für eine Ressource

den Pfad zum Erreichen eines Attributs beschreiben. Jede Periode steht für die Erweiterung eines Elements. Jedes Element muss in Pascal-Groß- und Kleinschreibung ausgedrückt werden, auch wenn die Ausgabe der Aktion Describe für eine Ressource in Camel-Groß- und Kleinschreibung erfolgt. Beispielsweise sollten SieAvailabilityZone, nicht availablityZone als Attributelement verwenden.

```
"filters": [
  {
    "path": "Component.Component.Component",
    "values": [
      "string"
    ]
  }
],
```

Die folgende Logik gilt für alle Ressourcenfilter:

- Wenn mehrere Filter bereitgestellt werden, einschließlich Filter mit demselben Pfad, müssen alle Filter übereinstimmen, damit eine Ressource ausgewählt werden kann — AND
- Wenn mehrere Werte für einen einzelnen Filter angegeben werden, muss ein beliebiger Wert abgeglichen werden, damit eine Ressource ausgewählt werden kann — OR
- Wenn an der Pfadposition des Describe-API-Aufrufs mehrere Werte gefunden werden, muss ein beliebiger Wert abgeglichen werden, damit eine Ressource ausgewählt werden kann — OR
- Um anhand von key/value Tag-Paaren abzugleichen, sollten Sie die Zielressourcen stattdessen anhand von Tags auswählen (siehe oben).

Die folgende Tabelle enthält die API-Aktionen und AWS CLI -Befehle, mit denen Sie die kanonischen Beschreibungen für jeden Ressourcentyp abrufen können. AWS FIS führt diese Aktionen in Ihrem Namen aus, um die von Ihnen angegebenen Filter anzuwenden. In der entsprechenden Dokumentation werden die Ressourcen beschrieben, die standardmäßig in den Ergebnissen enthalten sind. Beispielsweise könnte die Dokumentation für DescribeInstances Staaten, in denen Instanzen kürzlich beendet wurden, in den Ergebnissen erscheinen.

Ressourcentyp	API-Aktion	AWS CLI Befehl
aws:arc:zonal-shift-managed-resource	ListManagedResources	list-managed-resources

Ressourcentyp	API-Aktion	AWS CLI Befehl
aws:directconnect:virtual-interface	DescribeVirtualInterfaces	describe-virtual-interfaces
aws:ec2:autoscaling-group	DescribeAutoScalingGroups	describe-auto-scaling-groups
aws:ec2:ebs-volume	DescribeVolumes	Volumen beschreiben
aws:ec2:instance	DescribeInstances	Instanzen beschreiben
aws:ec2:subnet	DescribeSubnets	describe-subnets
aws:ec2:transit-gateway	DescribeTransitGateways	describe-transit-gateways
aws:ec2:vpc-endpoint	DescribeVpcEndpoints	describe-vpc-endpoints
aws:ecs:cluster	DescribeClusters	describe-clusters
aws:ecs:task	DescribeTasks	Aufgaben beschreiben
aws:eks:cluster	DescribeClusters	describe-clusters
aws:eks:nodegroup	DescribeNodegroup	Knotengruppe beschreiben
aws:elasticache:replication-group	DescribeReplicationGroups	describe-replication-groups
aws:iam:role	ListRoles	Rollen auflisten
aws:kinesis:stream	DescribeStreamSummary	describe-stream-summary
aws:lambda:function	ListFunctions	Listenfunktionen
aws:memorydb:multi-region-clusters	DescribeMultiRegionClusters	describe-multi-region-clusters
aws:rds:cluster	Beschreiben DBClusters	describe-db-clusters
aws:rds:db	Beschreiben DBInstances	describe-db-instances
aws:s3:bucket	ListBuckets	list-buckets

Ressourcentyp	API-Aktion	AWS CLI Befehl
aws:dynamodb:global-table	DescribeTable	Tabelle beschreiben
aws:dsq:cluster	GetCluster	Cluster abrufen

Beispiele finden Sie unter [the section called “Beispielfilter”](#).

Parameter für Ressourcen

Ressourcenparameter identifizieren Zielressourcen nach bestimmten Kriterien.

Der folgende Ressourcentyp unterstützt Parameter.

aws:ec2:ebs-volume

- `availabilityZoneIdentifizier`— Der Code (z. B. `us-east-1a`) der Availability Zone, die die Zielvolumes enthält.

aws:ec2:subnet

- `availabilityZoneIdentifizier`— Der Code (z. B. `us-east-1a`) oder die AZ-ID (z. B. `use1-az1`) der Availability Zone, die die Zielsubnetze enthält.
- `vpc`— Die VPC, die die Zielsubnetze enthält. Unterstützt nicht mehr als eine VPC pro Konto.

aws:ecs:task

- `cluster`— Der Cluster, der die Zielaufgaben enthält.
- `service`— Der Dienst, der die Zielaufgaben enthält.

aws:eks:pod

- `availabilityZoneIdentifizier` – Optional. Die Availability Zone, die die Ziel-Pods enthält. Beispiel, `us-east-1d`. Wir bestimmen die Availability Zone eines Pods, indem wir seine Host-IP und die CIDR des Cluster-Subnetzes vergleichen.
- `clusterIdentifizier` – Erforderlich. Der Name oder ARN des EKS-Ziel-Clusters.
- `namespace` – Erforderlich. Der Kubernetes-Namespace der Ziel-Pods.
- `selectorType` – Erforderlich. Der Selektortyp. Die möglichen Werte sind `labelSelector`, `deploymentName` und `podName`.
- `selectorValue` – Erforderlich. Der Selektorwert. Dieser Wert hängt vom Wert von `selectorType` ab.

- `targetContainerName` – Optional. Der Name des Zielcontainers, wie in der Pod-Spezifikation definiert. Die Standardeinstellung ist der erste Container, der in jeder Ziel-Pod-Spezifikation definiert ist.

`aws:lambda:function`

- `functionQualifier` – Optional. Die Version oder der Alias der Zielfunktion. Wenn kein Qualifier angegeben ist, werden alle Aufrufe für das Targeting berücksichtigt. Wenn ein Alias mit mehreren Versionen angegeben wird, werden alle im Alias enthaltenen Versionen für das Targeting berücksichtigt, sofern sie mit einem ARN aufgerufen werden, der den Alias enthält. Wenn der spezielle Alias verwendet `$LATEST` wird, werden Aufrufe der Basisfunktion ARN und Aufrufe, die `$LATEST` im ARN enthalten sind, für die Fehlerinjektion berücksichtigt. Weitere Informationen zu Lambda-Versionen finden Sie im AWS Lambda Benutzerhandbuch unter [Lambda-Funktionsversionen verwalten](#).

`aws:rds:cluster`

- `writerAvailabilityZoneIdentifiers` – Optional. Die Availability Zones des Writers des DB-Clusters. Mögliche Werte sind: eine durch Kommas getrennte Liste von Availability Zone-Identifikatoren, `all`

`aws:rds:db`

- `availabilityZoneIdentifiers` – Optional. Die Availability Zones der DB-Instance, die betroffen sein soll. Mögliche Werte sind: eine durch Kommas getrennte Liste von Availability Zone-Identifikatoren, `all`

`aws:elasticache:replicationgroup`

- `availabilityZoneIdentifier` – Erforderlich. Der Code (z. B. `us-east-1a`) oder die AZ-ID (z. B. `use1-az1`) der Availability Zone, die die Zielknoten enthält.

Auswahlmodus

Sie können die identifizierten Ressourcen eingrenzen, indem Sie einen Auswahlmodus angeben. AWS FIS unterstützt die folgenden Auswahlmodi:

- **ALL**— Führt die Aktion auf allen Zielen aus.
- **COUNT(n)**— Führt die Aktion für die angegebene Anzahl von Zielen aus, die nach dem Zufallsprinzip aus den identifizierten Zielen ausgewählt wurden. Beispielsweise wählt **COUNT (1)** eines der identifizierten Ziele aus.

- **PERCENT(n)**— Führt die Aktion für den angegebenen Prozentsatz von Zielen aus, die nach dem Zufallsprinzip aus den identifizierten Zielen ausgewählt wurden. **PERCENT (25)** wählt beispielsweise 25% der identifizierten Ziele aus.

Wenn Sie über eine ungerade Anzahl von Ressourcen verfügen und 50% angeben, rundet AWS FIS ab. Wenn Sie beispielsweise fünf EC2 Amazon-Instances als Ziele hinzufügen und den Umfang auf 50% erhöhen, rundet AWS FIS auf zwei Instances ab. Sie können keinen Prozentsatz angeben, der weniger als eine Ressource ist. Wenn Sie beispielsweise vier EC2 Amazon-Instances hinzufügen und den Umfang auf 5% erhöhen, kann AWS FIS keine Instance auswählen.

Wenn Sie mehrere Ziele mit demselben Zielressourcentyp definieren, kann AWS FIS dieselbe Ressource mehrmals auswählen.

Unabhängig davon, welchen Auswahlmodus Sie verwenden, schlägt das Experiment fehl, wenn der von Ihnen angegebene Bereich keine Ressourcen identifiziert.

Beispielziele

Im Folgenden finden Sie Beispielziele.

Beispiele

- [Instanzen in der angegebenen VPC mit den angegebenen Tags](#)
- [Aufgaben mit den angegebenen Parametern](#)

Beispiel: Instances in der angegebenen VPC mit den angegebenen Tags

Die möglichen Ziele für dieses Beispiel sind EC2 Amazon-Instances in der angegebenen VPC mit dem Tagenv=prod. Der Auswahlmodus gibt an, dass AWS FIS eines dieser Ziele nach dem Zufallsprinzip auswählt.

```
{
  "targets": {
    "randomInstance": {
      "resourceType": "aws:ec2:instance",
      "resourceTags": {
        "env": "prod"
      },
    },
    "filters": [
```

```
{
  {
    "path": "VpcId",
    "values": [
      "vpc-aabbcc11223344556"
    ]
  },
  "selectionMode": "COUNT(1)"
}
```

Beispiel: Aufgaben mit den angegebenen Parametern

Die möglichen Ziele für dieses Beispiel sind Amazon ECS-Aufgaben mit dem angegebenen Cluster und Service. Der Auswahlmodus gibt an, dass AWS FIS eines dieser Ziele nach dem Zufallsprinzip auswählt.

```
{
  "targets": {
    "randomTask": {
      "resourceType": "aws:ecs:task",
      "parameters": {
        "cluster": "myCluster",
        "service": "myService"
      },
      "selectionMode": "COUNT(1)"
    }
  }
}
```

Beispielfilter

Im Folgenden finden Sie Beispielfilter.

Beispiele

- [EC2 -Instances](#)
- [DB-Cluster](#)

Beispiel: EC2 Instanzen

Wenn Sie einen Filter für eine Aktion angeben, die den Ressourcentyp `aws:ec2:instance` unterstützt, verwendet AWS FIS den EC2 `describe-instances` Amazon-Befehl und wendet den Filter an, um die Ziele zu identifizieren.

Der `describe-instances` Befehl gibt eine JSON-Ausgabe zurück, in der sich jede Instance einer Struktur unterordnet. Instances Im Folgenden finden Sie eine Teilausgabe, die Felder enthält, die mit gekennzeichnet sind *italics*. Wir werden Beispiele bereitstellen, die diese Felder verwenden, um einen Attributpfad aus der Struktur der JSON-Ausgabe anzugeben.

```
{
  "Reservations": [
    {
      "Groups": [],
      "Instances": [
        {
          "ImageId": "ami-0011111111111111",
          "InstanceId": "i-00aaaaaaaaaaaaaaaa",
          "InstanceType": "t2.micro",
          "KeyName": "virginia-kp",
          "LaunchTime": "2020-09-30T11:38:17.000Z",
          "Monitoring": {
            "State": "disabled"
          },
          "Placement": {
            "AvailabilityZone": "us-east-1a",
            "GroupName": "",
            "Tenancy": "default"
          },
          "PrivateDnsName": "ip-10-0-1-240.ec2.internal",
          "PrivateIpAddress": "10.0.1.240",
          "ProductCodes": [],
          "PublicDnsName": "ec2-203-0-113-17.compute-1.amazonaws.com",
          "PublicIpAddress": "203.0.113.17",
          "State": {
            "Code": 16,
            "Name": "running"
          },
          "StateTransitionReason": "",
          "SubnetId": "subnet-aabbcc11223344556",
          "VpcId": "vpc-00bbbbbbbbbbbbbbbb",
          ...
        }
      ]
    }
  ]
}
```

```

        "NetworkInterfaces": [
        {
            ...
            "Groups": [
                {
                    "GroupName": "sec-group-1",
                    "GroupId": "sg-a0011223344556677"
                },
                {
                    "GroupName": "sec-group-1",
                    "GroupId": "sg-b9988776655443322"
                }
            ],
            ...
        },
        ...
    ],
    ...
    {
        ...
    }
],
"OwnerId": "123456789012",
"ReservationId": "r-aaaaaabbbbbb111111"
},
...
]
}

```

Um mithilfe eines Ressourcenfilters Instanzen in einer bestimmten Availability Zone auszuwählen, geben Sie den Attributpfad für AvailabilityZone und den Code für die Availability Zone als Wert an. Beispiel:

```

"filters": [
    {
        "path": "Placement.AvailabilityZone",
        "values": [ "us-east-1a" ]
    }
],

```

Um Instances in einem bestimmten Subnetz mithilfe eines Ressourcenfilters auszuwählen, geben Sie den Attributpfad für SubnetId und die ID des Subnetzes als Wert an. Beispiel:

```
"filters": [  
  {  
    "path": "SubnetId",  
    "values": [ "subnet-aabbcc11223344556" ]  
  }  
],
```

Um Instances auszuwählen, die sich in einem bestimmten Instanzstatus befinden, geben Sie den Attributpfad für Name und einen der folgenden Statusnamen als Wert an: pending | | running | shutting-down | terminated | stopping. stopped Beispiel:

```
"filters": [  
  {  
    "path": "State.Name",  
    "values": [ "running" ]  
  }  
],
```

Um Instances auszuwählen, denen eine beliebige Anzahl von Sicherheitsgruppen angehängt ist, geben Sie einen einzelnen Filter mit dem Attributpfad für GroupId und mehrere Sicherheitsgruppen an IDs. Beispiel:

```
"filters": [  
  {  
    "path": "NetworkInterfaces.Groups.GroupId",  
    "values": [  
      "sg-a0011223344556677",  
      "sg-f1100110011001100"  
    ]  
  }  
],
```

Um Instances auszuwählen, denen alle Sicherheitsgruppen angehängt sind, geben Sie mehrere Filter mit dem Attributpfad für GroupId und eine einzelne Sicherheitsgruppen-ID für jeden Filter an. Beispiel:

```
"filters": [  
  {  
    "path": "NetworkInterfaces.Groups.GroupId",  
    "values": [  
      "sg-a0011223344556677",  
      "sg-f1100110011001100"  
    ]  
  },  
  {  
    "path": "NetworkInterfaces.Groups.GroupId",  
    "values": [  
      "sg-b0011223344556677",  
      "sg-g1100110011001100"  
    ]  
  }  
],
```



```

        "sg-a0011223344556677"
    ]
},
{
    "path": "NetworkInterfaces.Groups.GroupId",
    "values": [
        "sg-b9988776655443322"
    ]
}
],

```

Beispiel: Amazon RDS-Cluster (DB-Cluster)

Wenn Sie einen Filter für eine Aktion angeben, die den Ressourcentyp `aws:rds:cluster` unterstützt, führt AWS FIS den Amazon `describe-db-clusters` RDS-Befehl aus und wendet den Filter an, um die Ziele zu identifizieren.

Der `describe-db-clusters` Befehl gibt für jeden DB-Cluster eine JSON-Ausgabe zurück, die der folgenden ähnelt. Das Folgende ist eine Teilausgabe, die Felder enthält, die mit gekennzeichnet sind *italics*. Wir werden Beispiele bereitstellen, die diese Felder verwenden, um einen Attributpfad aus der Struktur der JSON-Ausgabe anzugeben.

```

[
  {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
      "us-east-2a",
      "us-east-2b",
      "us-east-2c"
    ],
    "BackupRetentionPeriod": 7,
    "DatabaseName": "",
    "DBClusterIdentifier": "database-1",
    "DBClusterParameterGroup": "default.aurora-postgresql11",
    "DBSubnetGroup": "default-vpc-01234567abc123456",
    "Status": "available",
    "EarliestRestorableTime": "2020-11-13T15:08:32.211Z",
    "Endpoint": "database-1.cluster-example.us-east-2.rds.amazonaws.com",
    "ReaderEndpoint": "database-1.cluster-ro-example.us-east-2.rds.amazonaws.com",
    "MultiAZ": false,
    "Engine": "aurora-postgresql",
    "EngineVersion": "11.7",
  }
]

```

```
    ...  
  }  
]
```

Um einen Ressourcenfilter anzuwenden, der nur die DB-Cluster zurückgibt, die eine bestimmte DB-Engine verwenden, geben Sie den Attributpfad als Engine und den Wert an, `aurora-postgresql` wie im folgenden Beispiel gezeigt.

```
"filters": [  
  {  
    "path": "Engine",  
    "values": [ "aurora-postgresql" ]  
  }  
],
```

Um einen Ressourcenfilter anzuwenden, der nur die DB-Cluster in einer bestimmten Availability Zone zurückgibt, geben Sie den Attributpfad und den Wert an, wie im folgenden Beispiel gezeigt.

```
"filters": [  
  {  
    "path": "AvailabilityZones",  
    "values": [ "us-east-2a" ]  
  }  
],
```

Stoppbedingungen für AWS FIS

AWS Der AWS Fault Injection Service (FIS) bietet Kontrollen und Leitplanken, mit denen Sie Experimente bei Workloads sicher durchführen können. AWS Eine Stoppbedingung ist ein Mechanismus, um ein Experiment zu beenden, wenn es einen Schwellenwert erreicht, den Sie als CloudWatch Amazon-Alarm definieren. Wenn während eines Experiments eine Stopp-Bedingung ausgelöst AWS wird, stoppt FIS das Experiment. Sie können ein gestopptes Experiment nicht fortsetzen.

Um eine Abbruchbedingung zu erstellen, definieren Sie zunächst den stationären Zustand für Ihre Anwendung oder Ihren Dienst. Der Steady-State liegt vor, wenn Ihre Anwendung optimal funktioniert, was anhand geschäftlicher oder technischer Kennzahlen definiert wird. Zum Beispiel Latenz, CPU-Last oder Anzahl der Wiederholungen. Sie können den stationären Zustand verwenden, um

einen CloudWatch Alarm auszulösen, mit dem Sie ein Experiment beenden können, wenn Ihre Anwendung oder Ihr Dienst einen Zustand erreicht, in dem die Leistung nicht akzeptabel ist. Weitere Informationen finden Sie unter [Verwenden von CloudWatch Amazon-Alarmen](#) im CloudWatch Amazon-Benutzerhandbuch.

Ihr Konto hat ein Kontingent für die Anzahl der Stoppbedingungen, die Sie in einer Versuchsvorlage angeben können. Weitere Informationen finden Sie unter [Kontingente und Einschränkungen für den AWS Fault Injection Service](#).

Syntax der Stoppbedingungen

Wenn Sie eine Versuchsvorlage erstellen, geben Sie eine oder mehrere Abbruchbedingungen an, indem Sie die CloudWatch Alarme angeben, die Sie erstellt haben.

```
{
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": "arn:aws:cloudwatch:region:123456789012:alarm:alarm-name"
    }
  ]
}
```

Das folgende Beispiel zeigt, dass in der Versuchsvorlage keine Stoppbedingung angegeben ist.

```
{
  "stopConditions": [
    {
      "source": "none"
    }
  ]
}
```

Weitere Informationen

Ein Tutorial, das zeigt, wie Sie einen CloudWatch Alarm erstellen und einer Versuchsvorlage eine Stoppbedingung hinzufügen, finden Sie unter [CPU-Belastung auf einer Instanz ausführen](#).

Weitere Informationen zu den CloudWatch Metriken, die für die von AWS FIS unterstützten Ressourcentypen verfügbar sind, finden Sie im Folgenden:

- [Überwachen Sie Ihre Instanzen mit CloudWatch](#)
- [Amazon CloudWatch ECS-Metriken](#)
- [Überwachung von Amazon RDS-Metriken mit CloudWatch](#)
- [Überwachung von Run Command-Metriken mithilfe von CloudWatch](#)

IAM-Rollen für AWS FIS-Experimente

AWS Identity and Access Management (IAM) ist ein AWS Dienst, der einem Administrator hilft, den Zugriff auf AWS Ressourcen sicher zu kontrollieren. Um AWS FIS verwenden zu können, müssen Sie eine IAM-Rolle erstellen, die AWS FIS die erforderlichen Berechtigungen erteilt, damit AWS FIS in Ihrem Namen Experimente durchführen kann. Sie geben diese Experimentrolle an, wenn Sie eine Experimentvorlage erstellen. Für ein Experiment mit einem einzigen Konto muss die IAM-Richtlinie für die Experimentrolle die Erlaubnis erteilen, die Ressourcen zu ändern, die Sie in Ihrer Experimentvorlage als Ziele angeben. Bei einem Experiment mit mehreren Konten muss die Experimentrolle der Orchestrator-Rolle die Berechtigung erteilen, die IAM-Rolle für jedes Zielkonto zu übernehmen. Weitere Informationen finden Sie unter [Berechtigungen für Experimente mit mehreren Konten](#).

Wir empfehlen, dass Sie die standardmäßige Sicherheitspraxis der Gewährung der geringsten Rechte befolgen. Sie können dies tun, indem Sie in Ihren Richtlinien bestimmte Ressourcen ARNs oder Tags angeben.

Damit Sie schnell mit AWS FIS loslegen können, stellen wir AWS verwaltete Richtlinien zur Verfügung, die Sie bei der Erstellung einer Testrolle angeben können. Alternativ können Sie diese Richtlinien auch als Modell verwenden, wenn Sie Ihre eigenen Inline-Richtliniendokumente erstellen.

Inhalt

- [Voraussetzungen](#)
- [Option 1: Erstellen Sie eine Experimentrolle und fügen Sie eine AWS verwaltete Richtlinie hinzu](#)
- [Option 2: Erstellen Sie eine Experimentrolle und fügen Sie ein integriertes Richtliniendokument hinzu](#)

Voraussetzungen

Bevor Sie beginnen, installieren Sie die Vertrauensrichtlinie AWS CLI und erstellen Sie die erforderliche Vertrauensrichtlinie.

Installieren Sie das AWS CLI

Bevor Sie beginnen, installieren und konfigurieren Sie die AWS CLI. Wenn Sie das konfigurieren AWS CLI, werden Sie zur Eingabe von AWS Anmeldeinformationen aufgefordert. In den Beispielen in diesem Tutorial wird davon ausgegangen, dass Sie eine Standardregion konfiguriert haben. Andernfalls fügen Sie für jeden Befehl die `--region`-Option hinzu. Informationen finden Sie unter [Installieren und Aktualisieren der AWS CLI](#) und [Konfigurieren der AWS CLI](#).

Erstellen Sie eine Vertrauensbeziehungsrichtlinie

Eine experimentelle Rolle muss über eine Vertrauensbeziehung verfügen, die es dem AWS FIS-Dienst ermöglicht, die Rolle zu übernehmen. Erstellen Sie eine Textdatei mit dem Namen `fis-role-trust-policy.json` und fügen Sie die folgende Vertrauensbeziehungsrichtlinie hinzu.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "fis.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Wir empfehlen Ihnen, die `aws:SourceAccount`- und `aws:SourceArn`-Bedingungsschlüssel zu verwenden, um sich vor dem [Problem des verwirrten Stellvertreters](#) zu schützen. Das Quellkonto ist der Besitzer des Experiments und der Quell-ARN ist der ARN des Experiments. Sie sollten beispielsweise den folgenden Bedingungsblock zu Ihrer Vertrauensrichtlinie hinzufügen.

```
"Condition": {
  "StringEquals": {
    "aws:SourceAccount": "account_id"
  },
  "ArnLike": {
    "aws:SourceArn": "arn:aws:fis:region:account_id:experiment/*"
  }
}
```

```
}
}
```

Fügen Sie Berechtigungen hinzu, um die Rollen eines Zielkontos zu übernehmen (nur bei Experimenten mit mehreren Konten)

Für Experimente mit mehreren Konten benötigen Sie Berechtigungen, die es dem Orchestrator-Konto ermöglichen, Zielkontenrollen zu übernehmen. Sie können das folgende Beispiel ändern und es als integriertes Richtliniendokument hinzufügen, um Zielkontenrollen zu übernehmen:

```
{
  "Effect": "Allow",
  "Action": "sts:AssumeRole",
  "Resource": [
    "arn:aws:iam::target_account_id:role/role_name"
  ]
}
```

Option 1: Erstellen Sie eine Experimentrolle und fügen Sie eine AWS verwaltete Richtlinie hinzu

Verwenden Sie eine der AWS verwalteten Richtlinien von AWS FIS, um schnell loszulegen.

Um eine Experimentierrolle zu erstellen und eine AWS verwaltete Richtlinie anzuhängen

1. Stellen Sie sicher, dass es in Ihrem Experiment eine verwaltete Richtlinie für die AWS FIS-Aktionen gibt. Andernfalls müssen Sie stattdessen Ihr eigenes Inline-Richtliniendokument erstellen. Weitere Informationen finden Sie unter [the section called “AWS verwaltete Richtlinien”](#).
2. Verwenden Sie den folgenden Befehl [create-role](#), um eine Rolle zu erstellen und die Vertrauensrichtlinie hinzuzufügen, die Sie in den Voraussetzungen erstellt haben.

```
aws iam create-role --role-name my-fis-role --assume-role-policy-document
file://fis-role-trust-policy.json
```

3. Verwenden Sie den folgenden [attach-role-policy](#) Befehl, um die AWS verwaltete Richtlinie anzuhängen.

```
aws iam attach-role-policy --role-name my-fis-role --policy-arn fis-policy-arn
```

Wo *fis-policy-arn* ist einer der folgenden:

- `arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorEC2Access`
- `arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorECSAccess`
- `arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorEKSAccess`
- `arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorNetworkAccess`
- `arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorRDSAccess`
- `arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorSSMAccess`

Option 2: Erstellen Sie eine Experimentrolle und fügen Sie ein integriertes Richtliniendokument hinzu

Verwenden Sie diese Option für Aktionen, für die es keine verwaltete Richtlinie gibt, oder um nur die Berechtigungen einzubeziehen, die für Ihr spezielles Experiment erforderlich sind.

Um ein Experiment zu erstellen und ein integriertes Richtliniendokument hinzuzufügen

1. Verwenden Sie den folgenden Befehl [create-role](#), um eine Rolle zu erstellen und die Vertrauensrichtlinie hinzuzufügen, die Sie in den Voraussetzungen erstellt haben.

```
aws iam create-role --role-name my-fis-role --assume-role-policy-document  
file://fis-role-trust-policy.json
```

2. Erstellen Sie eine Textdatei mit dem Namen `fis-role-permissions-policy.json` und fügen Sie eine Berechtigungsrichtlinie hinzu. Ein Beispiel, das Sie als Ausgangspunkt verwenden können, finden Sie im Folgenden.

- Aktionen zur Fehlerinjektion — Gehen Sie von der folgenden Richtlinie aus.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowFISExperimentRoleFaultInjectionActions",  
      "Effect": "Allow",  
      "Action": [  

```

```

        "fis:InjectApiInternalError",
        "fis:InjectApiThrottleError",
        "fis:InjectApiUnavailableError"
    ],
    "Resource": "arn:*:fis:*:experiment/*"
}
]
}

```

- Amazon EBS-Aktionen — Gehen Sie von der folgenden Richtlinie aus.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:PauseVolumeIO"
      ],
      "Resource": "arn:aws:ec2:*:volume/*"
    }
  ]
}

```

- Amazon EC2 EC2-Aktionen — Beginnen Sie mit der [AWSFaultInjectionSimulatorEC2Zugriffsrichtlinie](#).
- Amazon ECS-Aktionen — Beginnen Sie mit der [AWSFaultInjectionSimulatorECSAccess](#) Richtlinie.
- Amazon EKS-Aktionen — Beginnen Sie mit der [AWSFaultInjectionSimulatorEKSAccess](#) Richtlinie.
- Netzwerkaktionen — Beginnen Sie mit der [AWSFaultInjectionSimulatorNetworkAccess](#) Richtlinie.

- Amazon RDS-Aktionen — Beginnen Sie mit der [AWSFaultInjectionSimulatorRDSAccess](#)Richtlinie.
 - Systems Manager Manager-Aktionen — Beginnen Sie mit der [AWSFaultInjectionSimulatorSSMAccess](#)Richtlinie.
3. Verwenden Sie den folgenden [put-role-policy](#)Befehl, um die Berechtigungsrichtlinie hinzuzufügen, die Sie im vorherigen Schritt erstellt haben.

```
aws iam put-role-policy --role-name my-fis-role --policy-name my-fis-policy --  
policy-document file://fis-role-permissions-policy.json
```

Konfigurationen von Experimentberichten für AWS FIS

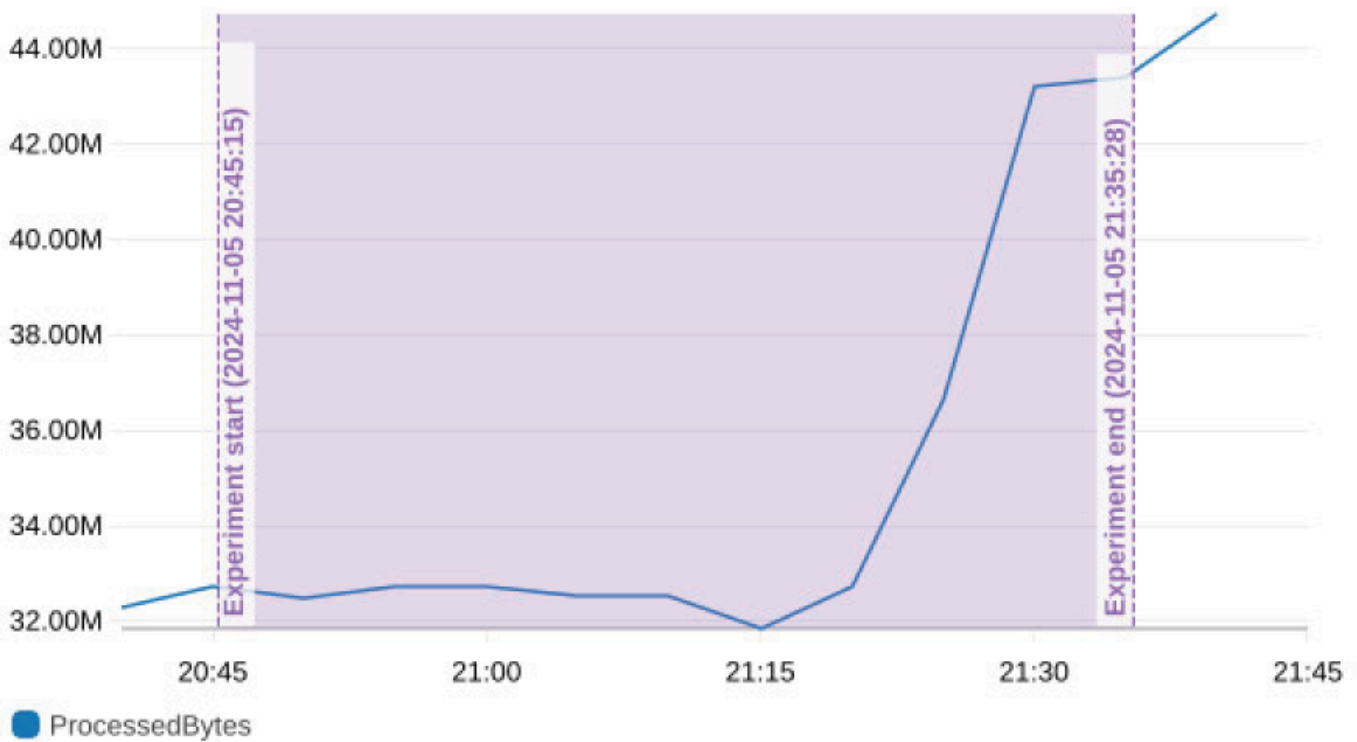
Sie können den AWS Fault Injection Service (FIS) aktivieren, um Berichte für Experimente zu generieren, was es einfacher macht, Nachweise für Resilienztests vorzulegen. Der Versuchsbericht ist ein PDF-Dokument, das die Versuchsaktionen zusammenfasst und optional die Reaktion der Anwendung aus einem von Ihnen CloudWatch angegebenen Dashboard erfasst. Um ein Beispiel für einen Versuchsbericht zu sehen, laden Sie die ZIP-Datei [hier](#) herunter.

Um den Inhalt des für das Experiment generierten Berichts zu aktivieren und zu konfigurieren, definieren Sie die Konfiguration des Experimentberichts für die Versuchsvorlage. Wenn Sie ein CloudWatch Dashboard angeben, enthält AWS FIS ein Snapshot-Diagramm aller Widgets im angegebenen Dashboard, das mit der Start- und Endzeit des Experiments über einen von Ihnen angegebenen Zeitraum annotiert ist, wie im Beispiel unten gezeigt.

Dieses Beispiel zeigt die Auswirkungen eines Experiments zum Verlust von Paketen in einer Availability Zone (AZ). Wenn Paketverlust in AZ use1-az6 eingeführt wird, verlagert sich der Datenverkehr weg von use1-az6 und hin zu use1-az4, sodass die Anzahl der vom Load Balancer in dieser AZ verarbeiteten Byte abnimmt.

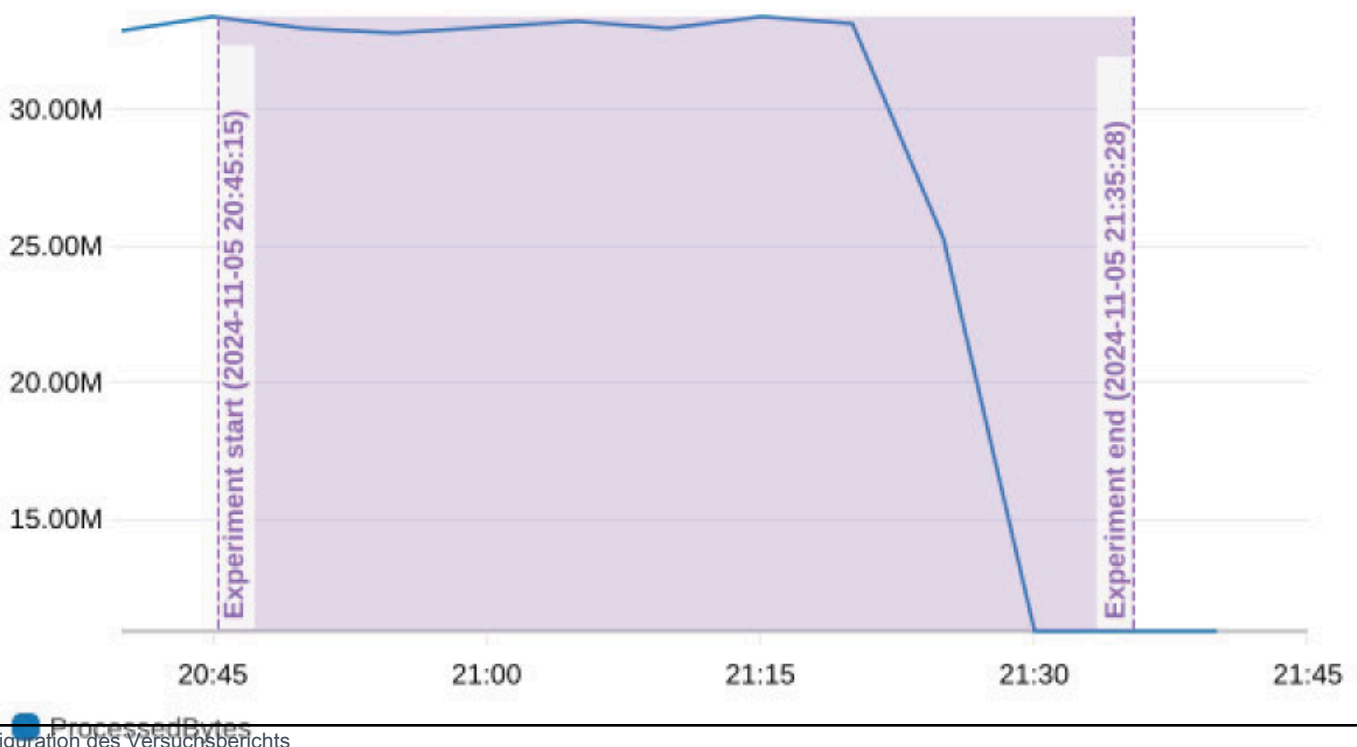
NLB ProcessedBytes use1-az4

Bytes



NLB ProcessedBytes use1-az6

Bytes



Wenn das Experiment beendet ist, kann der Bericht von der AWS FIS-Konsole heruntergeladen werden und wird auch in einem Amazon S3 S3-Bucket gespeichert. Wenn Sie ein CloudWatch Dashboard in Ihre Berichtskonfiguration aufnehmen, werden auch Bilder von jedem Widget geliefert. Berichte werden nicht für Experimente generiert, die Teil der Zielvorschau sind cancelled oder ausgeführt werden (wenn ActionsMode auf skip-all gesetzt ist). Sobald das Experiment das Datenaufbewahrungslimit für das Experiment überschreitet, ist der Bericht nur im Amazon S3 S3-Bucket verfügbar. AWS FIS-Gebühren fallen für jeden übermittelten Bericht an, mit Ausnahme von Berichten, die aufgrund interner Fehler fehlschlagen. Weitere Informationen finden Sie unter [Preise für den AWS Fault Injection Service](#) und [Kontingente und Einschränkungen für den AWS Fault Injection Service](#). Aufnahme- und Speichergebühren für Amazon S3 sowie CloudWatch API-Gebühren für GetMetricWidgetImage und GetDashboardAnfragen können anfallen. Weitere Informationen finden Sie unter [Amazon S3 — Preise](#) und [CloudWatch Preise](#).

Inhalt

- [Syntax der Konfiguration des Versuchsberichts](#)
- [Berechtigungen für Versuchsberichte](#)
- [Bewährte Methoden für Versuchsberichte](#)

Syntax der Konfiguration des Versuchsberichts

Im Folgenden finden Sie die Syntax für die Konfiguration des Versuchsberichts, einen optionalen Abschnitt der Versuchsvorlage.

```
{
  "experimentReportConfiguration": {
    "outputs": {
      "s3Configuration": {
        "bucketName": "my-bucket-name",
        "prefix": "report-storage-prefix"
      }
    },
    "dataSources": {
      "cloudWatchDashboards": [
        {
          "dashboardIdentifier": "arn:aws:cloudwatch::123456789012:dashboard/MyDashboard"
        }
      ]
    }
  },
}
```

```
    "preExperimentDuration": "PT20M",  
    "postExperimentDuration": "PT20M"  
  }  
}
```

Mithilfe von können Sie das `experimentReportConfiguration` Ausgabeziel, die Eingabedaten und die Zeitfenster für die Daten, die in den Experimentbericht aufgenommen werden sollen, anpassen, was Ihnen helfen kann, die Auswirkungen und Ergebnisse Ihrer AWS FIS-Experimente besser zu verstehen. Wenn Sie die Konfiguration des Versuchsberichts definieren, geben Sie Folgendes an:

outputs

Abschnitt `vonexperimentReportConfiguration`, der angibt, wohin der Versuchsbericht geliefert werden soll. In spezifizieren Sie `dasoutputs`, `s3Configuration` indem Sie Folgendes angeben:

- `bucketName`- Der Name des Amazon S3 S3-Buckets, in dem der Bericht gespeichert wird. Der Bucket muss sich in derselben Region wie das Experiment befinden.
- `prefix(Optional)` — Ein Präfix innerhalb des Amazon S3 S3-Buckets, in dem der Bericht gespeichert wird. Dieses Feld wird dringend empfohlen, damit Sie den Zugriff nur auf das Präfix beschränken können.

Datenquellen

Optionaler Abschnitt von, in dem `experimentReportConfiguration` die zusätzlichen Datenquellen angegeben werden, die in den Experimentbericht aufgenommen werden.

- `cloudWatchDashboards`- Eine Reihe von CloudWatch Dashboards, die in den Bericht aufgenommen werden. Beschränkt auf ein CloudWatch Dashboard.
- `dashboardIdentifier`- Der ARN des CloudWatch Armaturenbretts. Snapshot-Diagramme aller Widgets mit dem Typ `metric` in diesem Dashboard werden in den Bericht aufgenommen, mit Ausnahme von regionsübergreifenden Metriken.

preExperimentDuration

Optionaler Abschnitt von, in dem `experimentReportConfiguration` die Dauer bis zu 30 Minuten vor dem Experiment für die CloudWatch Dashboard-Metriken definiert wird, die in den Bericht aufgenommen werden sollen. Dabei sollte es sich um einen Zeitraum handeln, der den stabilen Zustand Ihrer Anwendung darstellt. Eine Dauer von 5 Minuten vor dem Experiment bedeutet beispielsweise, dass die Schnappschussdiagramme 5 Minuten vor

Beginn des Experiments Metriken enthalten. Das Format für die Dauer ist ISO 8601 und die Standardeinstellung ist 20 Minuten.

postExperimentDuration

Optionaler Abschnitt von, in dem `experimentReportConfiguration` die Dauer bis zu 2 Stunden nach dem Experiment für die CloudWatch Dashboard-Metriken definiert wird, die in den Bericht aufgenommen werden sollen. Dabei sollte es sich um eine Dauer handeln, die dem Dauerzustand oder der Erholungsphase Ihrer Anwendung entspricht. Wenn Sie beispielsweise eine Dauer von 5 Minuten nach dem Experiment angeben, enthalten die Snapshot-Diagramme Metriken bis 5 Minuten nach Ende des Experiments. Das Format für die Dauer ist ISO 8601 und die Standardeinstellung ist 20 Minuten.

Berechtigungen für Versuchsberichte

Damit AWS FIS den Versuchsbericht generieren und speichern kann, müssen Sie in Ihrer IAM-Rolle für das AWS FIS-Experiment die folgenden Operationen zulassen:

- `cloudwatch:GetDashboard`
- `cloudwatch:GetMetricWidgetImage`
- `s3:GetObject`
- `s3:PutObject`

Wir empfehlen Ihnen, bewährte AWS Sicherheitsmethoden zu befolgen und die Rolle des Experiments auf den Bucket und das Präfix zu beschränken. Im Folgenden finden Sie ein Beispiel für eine Richtlinienanweisung, die den Zugriff auf die Versuchsrolle einschränkt.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
```

```

        "Resource": "arn:aws:s3::my-experiment-report-bucket/my-prefix/
    *",
        "Effect": "Allow"
    },
    {
        "Action": [
            "cloudwatch:GetDashboard"
        ],
        "Resource": "arn:aws:cloudwatch::012345678912:dashboard/my-
experiment-report-dashboard",
        "Effect": "Allow"
    },
    {
        "Action": [
            "cloudwatch:GetMetricWidgetImage"
        ],
        "Resource": "*",
        "Effect": "Allow"
    }
]
}

```

Zusätzliche Berechtigungen für Berichte, die an Amazon S3 S3-Buckets gesendet werden und mit vom Kunden verwalteten Schlüsseln (CMK) verschlüsselt sind

Wenn der Amazon S3 S3-Bucket, den Sie im angeben, mit CMK verschlüsselt S3Configuration ist, müssen Sie der FIS-Experimentierrolle in Ihrer KMS-Schlüsselrichtlinie die folgenden zusätzlichen Berechtigungen gewähren:

- kms:GenerateDataKey
- kms:Decrypt

Im Folgenden finden Sie ein Beispiel für eine KMS-Schlüsselrichtlinie, die es der FIS-Experimentierrolle ermöglicht, Berichte in verschlüsselte Buckets zu schreiben:

```

{
    "Sid": "Allow FIS experiment report",
    "Effect": "Allow",
    "Principal":
    {

```

```
"AWS": [
    "arn:aws:iam::012345678912:role/FISExperimentRole",
],
"Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
],
"Resource": "*"
}
```

Bewährte Methoden für Versuchsberichte

Im Folgenden finden Sie bewährte Methoden für die Verwendung der AWS FIS-Konfiguration für Versuchsberichte:

- Bevor Sie ein Experiment starten, generieren Sie eine Zielvorschau, um zu überprüfen, ob Ihre Experimentvorlage erwartungsgemäß konfiguriert ist. Die Zielvorschau gibt Ihnen Informationen über die erwarteten Ziele Ihres Experiments. Weitere Informationen hierzu finden Sie unter [Generieren Sie eine Zielvorschau aus einer Experimentvorlage](#).
- Der Bericht sollte nicht zur Behebung fehlgeschlagener Experimente verwendet werden. Verwenden Sie stattdessen Versuchsprotokolle, um Versuchsfehler zu beheben. Wir empfehlen, dass Sie sich nur bei Experimenten, die Sie zuvor ausgeführt und erfolgreich abgeschlossen haben, auf den Bericht verlassen.
- Schränken Sie den Zugriff der IAM-Rolle für das Experiment ein und erhalten Sie Objektzugriff auf den S3-Ziel-Bucket und das Präfix. Wir empfehlen, das Bucket-/Präfix nur für AWS FIS-Experimentberichte zu verwenden und anderen AWS Diensten keinen Zugriff auf diesen Bucket und dieses Präfix zu gewähren.
- Verwenden Sie Amazon S3 Object Lock, um zu verhindern, dass der Bericht für einen bestimmten Zeitraum oder auf unbestimmte Zeit gelöscht oder überschrieben wird. Weitere Informationen finden Sie unter [Sperren von Objekten mit Object Lock](#).
- Wenn sich Ihr CloudWatch Dashboard in einem separaten Konto in derselben Region befindet, können Sie die CloudWatch kontoübergreifende Observability verwenden, um Ihr AWS FIS-Orchestrator-Konto als Überwachungskonto und das separate Konto als Quellkonto über die CloudWatch Konsole oder die Observability Access Manager-Befehle in der AND-API zu aktivieren. AWS CLI [Weitere Informationen finden Sie unter Kontoübergreifende Observabilität. CloudWatch](#)

Experimentieroptionen für AWS FIS

Experimentoptionen sind optionale Einstellungen für ein Experiment. Sie können bestimmte Experimentoptionen in der Experimentvorlage definieren. Zusätzliche Experimentoptionen werden festgelegt, wenn Sie mit dem Experiment beginnen.

Im Folgenden finden Sie die Syntax für die Experimentoptionen, die Sie in der Experimentvorlage definieren.

```
{
  "experimentOptions": {
    "accountTargeting": "single-account | multi-account",
    "emptyTargetResolutionMode": "fail | skip"
  }
}
```

Wenn Sie bei der Erstellung der Experimentvorlage keine Experimentoptionen angeben, wird die Standardeinstellung für jede Option verwendet.

Im Folgenden finden Sie die Syntax für die Experimentoptionen, die Sie zu Beginn des Experiments festlegen.

```
{
  "experimentOptions": {
    "actionsMode": "run-all | skip-all"
  }
}
```

Wenn Sie zu Beginn des Experiments keine Experimentoptionen angeben, run-all wird die Standardeinstellung verwendet.

Inhalt

- [Ausrichtung auf Konten](#)
- [Leerer Zielauflösungsmodus](#)
- [Modus „Aktionen“](#)

Ausrichtung auf Konten

Wenn Sie mehrere AWS Konten mit Ressourcen haben, auf die Sie in einem Experiment abzielen möchten, können Sie mithilfe der Option `Konten-Targeting-Experiment` ein Experiment mit mehreren Konten definieren. Sie führen Experimente mit mehreren Konten von einem Orchestrator-Konto aus durch, das sich auf Ressourcen in mehreren Zielkonten auswirkt. Das Orchestrator-Konto ist Eigentümer der AWS FIS Versuchsvorlage und des Experiments. Ein Zielkonto ist ein einzelnes AWS-Konto mit Ressourcen, die durch ein AWS FIS Experiment beeinträchtigt werden können. Weitere Informationen finden Sie unter [Arbeiten mit Multi-Account-Experimenten für AWS FIS](#).

Sie verwenden das Konto-Targeting, um den Standort Ihrer Zielressourcen anzugeben. Sie können zwei Werte für das Konto-Targeting angeben:

- Einzelkonto — Standard. Das Experiment zielt nur auf Ressourcen in dem AWS Konto ab, auf dem das AWS FIS Experiment ausgeführt wird.
- mehrere Konten — Das Experiment kann auf Ressourcen in mehreren AWS-Konten abzielen.

Konfigurationen der Zielkonten

Um ein Experiment mit mehreren Konten durchzuführen, müssen Sie eine oder mehrere Zielkontenkonfigurationen definieren. Eine Zielkontokonfiguration spezifiziert die `accountId`, `roleArn` und eine Beschreibung für jedes Konto mit Ressourcen, auf die das Experiment abzielt. Das Konto IDs der Zielkontokonfigurationen für eine Versuchsvorlage muss eindeutig sein.

Wenn Sie eine Versuchsvorlage mit mehreren Konten erstellen, gibt die Experimentvorlage ein schreibgeschütztes Feld `zurücktargetAccountConfigurationsCount`, d. h. die Anzahl aller Zielkontokonfigurationen für die Versuchsvorlage.

Im Folgenden finden Sie die Syntax für die Konfiguration eines Zielkontos.

```
{
  accountId: "123456789012",
  roleArn: "arn:aws:iam::123456789012:role/AllowFISActions",
  description: "fis-ec2-test"
}
```

Wenn Sie eine Zielkontokonfiguration erstellen, geben Sie Folgendes an:

accountId

12-stellige AWS-Konto-ID des Zielkontos.

roleArn

Eine IAM-Rolle, die AWS FIS Berechtigungen zum Ausführen von Aktionen im Zielkonto gewährt.

description

Eine optionale Beschreibung.

Weitere Informationen zum Arbeiten mit Zielkontokonfigurationen finden Sie unter [Arbeiten mit Multi-Account-Experimenten für AWS FIS](#).

Leerer Zielauflösungsmodus

In diesem Modus können Sie festlegen, dass Experimente auch dann abgeschlossen werden, wenn eine Zielressource nicht aufgelöst wurde.

- scheitern — Standardeinstellung. Wenn keine Ressourcen für das Ziel gefunden wurden, wird das Experiment sofort mit dem Status beendet `failed`.
- überspringen — Wenn keine Ressourcen für das Ziel gefunden wurden, wird das Experiment fortgesetzt und alle Aktionen ohne gelöste Ziele werden übersprungen. Aktionen mit Zielen, die mit eindeutigen Kennungen definiert wurden, wie z. B. ARNs, können nicht übersprungen werden. Wenn ein Ziel, das mit einer eindeutigen Kennung definiert wurde, nicht gefunden wird, wird das Experiment sofort mit dem Status beendet `failed`.

Modus „Aktionen“

Der Aktionsmodus ist ein optionaler Parameter, den Sie angeben können, wenn Sie ein Experiment starten. Sie können den Aktionsmodus auf `skip-all` einstellen, um eine Zielvorschau zu generieren, bevor Fehler in Ihre Zielressourcen injiziert werden. Mit der Zielvorschau können Sie Folgendes überprüfen:

- Dass Sie Ihre Experimentvorlage so konfiguriert haben, dass sie auf die Ressourcen abzielt, die Sie erwarten. Die tatsächlichen Ressourcen, auf die Sie zu Beginn dieses Experiments abzielen, können sich von der Vorschau unterscheiden, da Ressourcen möglicherweise entfernt, aktualisiert oder nach dem Zufallsprinzip ausgewählt werden.

- Dass Ihre Protokollierungskonfigurationen korrekt eingerichtet sind.
- Dass Sie für Experimente mit mehreren Konten eine IAM-Rolle für jede Ihrer Zielkontokonfigurationen korrekt eingerichtet haben.

 Note

In diesem `skip-all` Modus können Sie nicht überprüfen, ob Sie über die erforderlichen Berechtigungen verfügen, um das AWS FIS Experiment durchzuführen und Maßnahmen für Ihre Ressourcen zu ergreifen.

Der Parameter für den Aktionsmodus akzeptiert die folgenden Werte:

- `run-all`— (Standard) Das Experiment führt Aktionen für Zielressourcen durch.
- `skip-all`- Bei dem Experiment werden alle Aktionen für Zielressourcen übersprungen.

Weitere Informationen darüber, wie Sie den Parameter für den Aktionsmodus festlegen, wenn Sie ein Experiment starten, finden Sie unter [Generieren Sie eine Zielvorschau aus einer Experimentvorlage](#).

AWS FIS Referenz zu Aktionen

Eine Aktion ist die Fault-Injection-Aktivität, die Sie mit AWS Fault Injection Service (AWS FIS) auf einem Ziel ausführen. AWS FIS stellt AWS dienstübergreifend vorkonfigurierte Aktionen für bestimmte Zieltypen bereit. Sie fügen Aktionen zu einer Experimentvorlage hinzu, die Sie dann zum Ausführen von Experimenten verwenden.

In dieser Referenz werden die häufigsten Aktionen in beschrieben AWS FIS, einschließlich Informationen zu den Aktionsparametern und den erforderlichen IAM-Berechtigungen. Sie können die unterstützten AWS FIS Aktionen auch mithilfe der AWS FIS Konsole oder mit dem Befehl [list-actions](#) über () AWS Command Line Interface auflisten. Sobald Sie den Namen einer bestimmten Aktion haben, können Sie mit dem [Befehl get-action](#) detaillierte Informationen zu der Aktion anzeigen. Weitere Informationen zur Verwendung von AWS FIS Befehlen mit dem AWS CLI finden Sie im [AWS Command Line Interface Benutzerhandbuch](#) und unter [fis](#) in der AWS CLI Befehlsreferenz.

Weitere Informationen zur Funktionsweise von AWS FIS Aktionen finden Sie unter [Aktionen für AWS FIS](#) und [So funktioniert der AWS Fault Injection Service mit IAM](#).

Aktionen

- [Aktionen zur Fehlerinjektion](#)
- [Aktion zur Wiederherstellung](#)
- [Warte auf Aktion](#)
- [CloudWatch Amazon-Aktionen](#)
- [Amazon DynamoDB DynamoDB-Aktionen](#)
- [Amazon Aurora DSQL-Aktionen](#)
- [Amazon EBS-Aktionen](#)
- [Amazon EC2 EC2-Aktionen](#)
- [Amazon ECS-Aktionen](#)
- [Amazon EKS-Aktionen](#)
- [ElastiCache Amazon-Aktionen](#)
- [Amazon Kinesis Data Streams Streams-Aktionen](#)
- [AWS Lambda Aktionen](#)
- [Amazon MemoryDB-Aktion](#)

- [Netzwerkaktionen](#)
- [Amazon RDS-Aktionen](#)
- [Amazon-S3-Aktionen](#)
- [Systems Manager Manager-Aktionen](#)
- [AWS Direct Connect Aktionen](#)
- [Verwenden Sie Systems Manager SSM-Dokumente mit AWS FIS](#)
- [Verwenden Sie die AWS FIS-Aktionen aws:ecs:task](#)
- [Verwenden Sie die AWS FIS-Aktionen aws:eks:pod](#)
- [Verwenden Sie die Aktionen AWS FIS aws:lambda:function](#)

Aktionen zur Fehlerinjektion

AWS FIS unterstützt die folgenden Fehlerinjektionsaktionen.

Aktionen

- [aws:fis:inject-api-internal-error](#)
- [aws:fis:inject-api-throttle-error](#)
- [aws:fis:inject-api-unavailable-error](#)

aws:fis:inject-api-internal-error

Fügt interne Fehler in Anfragen ein, die von der IAM-Zielrolle gestellt wurden. Die spezifische Antwort hängt von jedem Dienst und jeder API ab. Weitere Informationen finden Sie in der SDK- und API-Dokumentation Ihres Dienstes.

Ressourcentyp

- `aws:iam:role`

Parameters

- `duration`— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

- `service`— Der AWS Ziel-API-Namespace. Der unterstützte Wert ist `ec2` und `kinesis`.
- `percentage`— Der Prozentsatz (1—100) der Aufrufe, in die der Fehler eingeschleust werden soll.
- `operations`— Die Operationen, in die der Fehler eingefügt werden soll, getrennt durch Kommas. Eine Liste der API-Aktionen für den `ec2` Namespace finden Sie unter [Amazon EC2 API Reference](#) und [Amazon Kinesis Data Streams](#) API Reference.

Berechtigungen

- `fis:InjectApiInternalError`

`aws:fis:inject-api-throttle-error`

Fügt Drosselungsfehler in Anfragen ein, die von der IAM-Zielrolle gestellt wurden. Die spezifische Antwort hängt von jedem Dienst und jeder API ab. Weitere Informationen finden Sie in der SDK- und API-Dokumentation Ihres Dienstes.

Ressourcentyp

- `aws:iam:role`

Parameters

- `duration`— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. `PT1M` steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- `service`— Der AWS Ziel-API-Namespace. Der unterstützte Wert ist `ec2` und `kinesis`.
- `percentage`— Der Prozentsatz (1—100) der Aufrufe, in die der Fehler eingeschleust werden soll.
- `operations`— Die Operationen, in die der Fehler eingefügt werden soll, getrennt durch Kommas. Eine Liste der API-Aktionen für den `ec2` Namespace finden Sie unter [Amazon EC2 API Reference](#) und [Amazon Kinesis Data Streams](#) API Reference.

Berechtigungen

- `fis:InjectApiThrottleError`

aws:fis:inject-api-unavailable-error

Fügt Unverfügbare-Fehler in Anfragen ein, die von der IAM-Zielrolle gestellt wurden. Die spezifische Antwort hängt von jedem Dienst und jeder API ab. Weitere Informationen finden Sie in der SDK- und API-Dokumentation Ihres Dienstes.

Ressourcentyp

- `aws:iam:role`

Parameters

- `duration`— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- `service`— Der AWS Ziel-API-Namespace. Der unterstützte Wert ist `ec2` und `kinesis`.
- `percentage`— Der Prozentsatz (1—100) der Aufrufe, in die der Fehler eingeschleust werden soll.
- `operations`— Die Operationen, in die der Fehler eingefügt werden soll, getrennt durch Kommas. Eine Liste der API-Aktionen für den `ec2` Namespace finden Sie unter [Amazon EC2 API Reference](#) und [Amazon Kinesis Data Streams](#) API Reference.

Berechtigungen

- `fis:InjectApiUnavailableError`

Aktion zur Wiederherstellung

Wiederherstellungsmaßnahmen werden durchgeführt, um Risiken zu minimieren oder Anwendungen nach einer Beeinträchtigung zu schützen.

AWS FIS unterstützt die folgenden Wiederherstellungsaktionen.

aws:arc:start-zonal-autoshift

Leitet den Datenverkehr für unterstützte Ressourcen automatisch von einer potenziell beeinträchtigten Availability Zone (AZ) weg und leitet sie an fehlerfreie Ressourcen AZs in derselben AWS-Region weiter. Dies ermöglicht die automatische Zonenverlagerung über FIS. Zonal Autoshift ist

eine Funktion in Amazon Application Recovery Controller (ARC), die es ermöglicht, den Verkehr für eine Ressource in Ihrem Namen von einer AZ weg AWS zu verlagern, wenn AWS festgestellt wird, dass eine Beeinträchtigung vorliegt, die sich möglicherweise auf Kunden in der AZ auswirken könnte.

Wenn Sie die `aws:arc:start-zonal-autoshift` Aktion ausführen, AWS FIS verwaltet die Zonenverschiebung mithilfe von `StartZonalShift`, `UpdateZonalShift`, `CancelZonalShift` APIs wobei das `expiresIn` Feld für diese Anfragen aus Sicherheitsgründen auf 1 Minute gesetzt ist. Auf diese Weise kann AWS FIS die Zonenverschiebung bei unerwarteten Ereignissen wie Netzwerkausfällen oder Systemproblemen schnell rückgängig gemacht werden. In der ARC-Konsole wird im Feld Ablaufzeit der Wert AWS FIS-managed angezeigt, und der tatsächliche erwartete Ablauf wird durch die Dauer bestimmt, die in der Aktion Zonal Shift angegeben wurde.

Ressourcentyp

- `aws:arc:zonal-shift-managed-resource`

Bei Ressourcen, die mit Zonal Shift verwaltet werden, handelt es sich um Ressourcentypen wie Amazon EKS-Cluster, Amazon EC2 Application and Network Load Balancers und Amazon EC2 Auto Scaling Scaling-Gruppen, die für ARC Zonal Autoshift aktiviert werden können. Weitere Informationen finden Sie unter [Unterstützte Ressourcen und Aktivierung von zonalen Autoshift-Ressourcen](#) im ARC Developer Guide.

Parameters

- `duration`— Die Zeitspanne, für die der Verkehr verlagert wird. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- `availabilityZonelfield`— Der Verkehr bewegt sich von dieser AZ weg. Dies kann ein AZ-Name (`us-east-1a`) oder eine AZ-ID (`use1-az1`) sein.
- `managedResourceTypes`— Die durch Kommas getrennten Ressourcentypen, von denen der Traffic verlagert wird. Mögliche Optionen sind ASG (Auto Scaling Group), ALB (Application Load Balancer), NLB (Network Load Balancer) und EKS (Amazon EKS).
- `zonalAutoshiftStatus`— Der `zonalAutoshiftStatus` Status der Ressourcen, auf die Sie abzielen möchten. Mögliche Optionen sind `ENABLED`, `DISABLED`, und `ANY`. Der Standardwert ist `ENABLED`.

Berechtigungen

- `arc-zonal-shift:StartZonalShift`
- `arc-zonal-shift:GetManagedResource`
- `arc-zonal-shift:UpdateZonalShift`
- `arc-zonal-shift:CancelZonalShift`
- `arc-zonal-shift:ListManagedResources`
- automatische Skalierung: `DescribeTags`
- Etikett: `GetResources`

Warte auf Aktion

AWS FIS unterstützt die folgende Warteaktion.

`aws:fis:wait`

Führt die AWS FIS Warteaktion aus.

Parameters

- `duration`— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. `PT1M` steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- Keine

CloudWatch Amazon-Aktionen

AWS FIS unterstützt die folgende CloudWatch Amazon-Aktion.

`aws:cloudwatch:assert-alarm-state`

Überprüft, ob sich die angegebenen Alarmer in einem der angegebenen Alarmzustände befinden.

Ressourcentyp

- Keine

Parameters

- `alarmArns`— Der ARNs der Alarmer, durch Kommas getrennt. Sie können bis zu fünf Alarmer angeben.
- `alarmStates`— Die Alarmstatus, getrennt durch Kommas. Die möglichen Alarmzustände sind `OKALARM`, und `INSUFFICIENT_DATA`.

Berechtigungen

- `cloudwatch:DescribeAlarms`

Amazon DynamoDB DynamoDB-Aktionen

AWS FIS unterstützt die folgende Amazon DynamoDB DynamoDB-Aktion.

`aws:dynamodb:global-table-pause-replication`

Unterbricht die globale Tabellenreplikation von Amazon DynamoDB für mehrere Regionen in eine beliebige Replikattabelle. Tabellen können bis zu 5 Minuten nach Beginn der Aktion weiter repliziert werden.

Globale Tabellen mit mehreren Regionen (Strongly Consistent, MRSC)

Die folgenden Anweisungen werden dynamisch an die Richtlinie für die globale DynamoDB-MRSC-Zieltabelle angehängt:

```
{
  "Statement": [
    {
      "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxxxx",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "dynamodb:UpdateTable"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable",
    "Condition": {
      "DateLessThan": {
        "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
      },
      "ArnEquals": {
        "aws:PrincipalArn": "arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
      }
    }
  },
  {
    "Sid":
"DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxxxxxxxxForApplicationAutoScaling",
    "Effect": "Deny",
    "Principal": {
      "AWS": "*"
    },
    "Action": [
      "dynamodb:DescribeTable",
      "dynamodb:UpdateTable"
    ],
    "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable",
    "Condition": {
      "DateLessThan": {
        "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
      },
      "ArnEquals": {
        "aws:PrincipalArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable"
      }
    }
  }
]
}

```

Wenn einer Zieltabelle keine Ressourcenrichtlinien angehängt sind, wird eine Ressourcenrichtlinie für die Dauer des Experiments erstellt und nach Abschluss des Experiments automatisch gelöscht. Andernfalls wird die Fehleranweisung in eine bestehende Richtlinie eingefügt, ohne dass zusätzliche Änderungen an den vorhandenen Richtlinienanweisungen vorgenommen werden. Die Fehlererklärung wird dann am Ende des Experiments aus der Richtlinie entfernt.

Die globalen MRSC-Zieltabellen von Amazon DynamoDB unterliegen einem zusätzlichen Kontingent. Dieses Kontingent sieht vor, dass keine einzelne Tabelle innerhalb eines fortlaufenden Zeitfensters von 7 Tagen länger als 5.040 Minuten beeinträchtigt werden darf.

Globaltabellen für mehrere Regionen und letztlich konsistente (MREC)

Die folgende Anweisung wird dynamisch an die Richtlinie für die globale DynamoDB-MREC-Zieltabelle angehängt:

```
{
  "Statement": [
    {
      "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxxxx",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb>DeleteItem",
        "dynamodb:DescribeTable",
        "dynamodb:UpdateTable",
        "dynamodb:Scan",
        "dynamodb:DescribeTimeToLive",
        "dynamodb:UpdateTimeToLive"
      ],
      "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable",
      "Condition": {
        "DateLessThan": {
          "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
        },
        "ArnEquals": {
          "aws:PrincipalArn": "arn:aws:iam::123456789012:role/aws-service-role/replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
        }
      }
    }
  ]
}
```

Die folgende Anweisung wird dynamisch an die Stream-Richtlinie für die globale DynamoDB-MREC-Zieltabelle angehängt:

```
{
  "Statement": [
    {
      "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxxxx",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "dynamodb:GetRecords",
        "dynamodb:DescribeStream",
        "dynamodb:GetShardIterator"
      ],
      "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable/stream/2023-08-31T09:50:24.025",
      "Condition": {
        "DateLessThan": {
          "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
        },
        "ArnEquals": {
          "aws:PrincipalArn": "arn:aws:iam::123456789012:role/aws-service-role/replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
        }
      }
    }
  ]
}
```

Wenn einer Zieltabelle oder einem Zielstream keine Ressourcenrichtlinien angehängt sind, wird eine Ressourcenrichtlinie für die Dauer des Experiments erstellt und nach Abschluss des Experiments automatisch gelöscht. Andernfalls wird die Fehleranweisung in eine bestehende Richtlinie eingefügt, ohne dass zusätzliche Änderungen an den vorhandenen Richtlinienanweisungen vorgenommen werden. Die Fehlererklärung wird dann am Ende des Experiments aus der Richtlinie entfernt.

Ressourcentyp

- `aws:dynamodb:global-table`

Parameters

- **duration**— In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `dynamodb:PutResourcePolicy`
- `dynamodb>DeleteResourcePolicy`
- `dynamodb:GetResourcePolicy`
- `dynamodb:DescribeTable`
- `tag:GetResources`
- `dynamodb:InjectError` *

* Die Berechtigung ist nur erforderlich, wenn Sie auf globale MRSC-Tabellen abzielen

Amazon Aurora DSQL-Aktionen

AWS FIS unterstützt die folgenden Amazon Aurora DSQL-Aktionen.

`aws:dsql:cluster-connection-failure`

Erzeugt kontrollierte Verbindungsfehler in einem Aurora DSQL-Cluster für eine bestimmte Dauer, um die Ausfallsicherheit von Anwendungen zu testen.

Ressourcentyp

- `aws:dsql:cluster`

Parameters

- **duration**— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- **percentage**— Der Prozentsatz (1—100) der Aufrufe, bei denen der Fehler ausgelöst werden soll.

Berechtigungen

- `dsql:InjectError`
- `dsql:GetCluster`
- `tag:GetResources`

Informationen zum Starten des Experiments mit Aurora DSQL finden Sie unter [Fault Injection Testing](#) im Aurora DSQL-Benutzerhandbuch.

Amazon EBS-Aktionen

AWS FIS unterstützt die folgende Amazon EBS-Aktion.

Aktionen

- [aws:ebs:pause-volume-io](#)
- [aws:ebs:volume-io-latency](#)

aws:ebs:pause-volume-io

Unterbricht den I/O Betrieb auf EBS-Zielvolumes. Die Ziel-Volumes müssen sich in derselben Availability Zone befinden und an Instances angehängt sein, die auf dem Nitro-System basieren. Die Volumes können nicht an Instances auf einem Outpost angehängt werden.

Informationen zum Starten des Experiments mit der Amazon EC2 EC2-Konsole finden Sie unter [Fehlertests auf Amazon EBS](#) im Amazon EC2 EC2-Benutzerhandbuch.

Ressourcentyp

- `aws:ec2:ebs-volume`

Parameters

- `duration`— Die Dauer, von einer Sekunde bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute, PT5 S für fünf Sekunden und PT6 H für sechs Stunden. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein. Wenn die Dauer klein ist, z. B. PT5 S, I/O wird sie für die

angegebene Dauer angehalten. Aufgrund der Zeit, die für die Initialisierung des Experiments benötigt wird, kann es jedoch länger dauern, bis das Experiment abgeschlossen ist.

Berechtigungen

- `ec2:DescribeVolumes`
- `ec2:PauseVolumeIO`
- `tag:GetResources`

aws:ebs:volume-io-latency

Sorgt für Latenz bei I/O Vorgängen auf EBS-Zielvolumes. Die Ziel-Volumes müssen sich in derselben Availability Zone befinden und an [Nitro-basierte](#) Instances angehängt sein. Die Volumes können nicht an Instances in einem Outpost angehängt werden.

Informationen zum Starten des Experiments mit der Amazon EC2 EC2-Konsole finden Sie unter [Fehlertests auf Amazon EBS](#) im Amazon EBS-Benutzerhandbuch.

Ressourcentyp

- `aws:ec2:ebs-volume`

Parameters

- `readIOPercentage`— Der Prozentsatz der I/O Lesevorgänge, bei denen Latenz auftritt, liegt zwischen 0,1% und 100%. Dies ist der Prozentsatz aller I/O Lesevorgänge auf dem Volume, die während des Experiments beeinträchtigt werden. Der Standardwert ist 100.
- `readIOLatencyMilliseconds`— Die Latenz, die bei I/O Lesevorgängen entsteht, in Millisekunden, von 1 ms (IO2-Volumes) oder 10 ms (Nicht-IO2-Volumes) bis 60 Sekunden. Dies ist der Latenzwert, der für den angegebenen Prozentsatz der Lesevorgänge während des Experiments beobachtet wird. I/O Der Standardwert ist 100.
- `writeIOPercentage`— Der Prozentsatz der I/O Schreibvorgänge, bei denen Latenz ausgelöst wird, liegt zwischen 0,1% und 100%. Dies ist der Prozentsatz aller I/O Schreibvorgänge auf dem Volume, die während des Experiments beeinträchtigt werden. Der Standardwert ist 100.
- `writeIOLatencyMilliseconds`— Die Latenz, die bei I/O Schreibvorgängen entsteht, in Millisekunden, von 1 ms (IO2-Volumes) oder 10 ms (Nicht-IO2-Volumes) bis 60 Sekunden. Dies ist der

Latenzwert, der für den angegebenen Prozentsatz der Lesevorgänge während des Experiments beobachtet wird. I/O Der Standardwert ist 100.

- `duration`— Die Dauer, für die die Latenz injiziert wird, von 1 Sekunde bis 12 Stunden.

Berechtigungen

- `ec2:DescribeVolumes`
- `ec2:InjectVolumeIOLatency`
- `tag:GetResources`

Amazon EC2 EC2-Aktionen

AWS FIS unterstützt die folgenden Amazon EC2 EC2-Aktionen.

Aktionen

- [aws:ec2:api-insufficient-instance-capacity-error](#)
- [aws:ec2:asg-insufficient-instance-capacity-error](#)
- [aws:ec2:reboot-instances](#)
- [aws:ec2:send-spot-instance-interruptions](#)
- [aws:ec2:stop-instances](#)
- [aws:ec2:terminate-instances](#)

AWS FIS unterstützt auch Fault-Injection-Aktionen über den AWS Systems Manager SSM-Agenten. Systems Manager verwendet ein SSM-Dokument, das Aktionen definiert, die auf EC2-Instances ausgeführt werden sollen. Sie können Ihr eigenes Dokument verwenden, um benutzerdefinierte Fehler einzufügen, oder Sie können vorkonfigurierte SSM-Dokumente verwenden. Weitere Informationen finden Sie unter [the section called “Aktionen für SSM-Dokumente”](#).

aws:ec2:api-insufficient-instance-capacity-error

Fügt `InsufficientInstanceCapacity` Fehlerantworten auf Anfragen ein, die von den IAM-Zielrollen gestellt wurden. Unterstützte Operationen sind `RunInstances`, `CreateCapacityReservation` `StartInstances`, `CreateFleet` Aufrufe. Anfragen, die Kapazitätsanfragen in mehreren Availability Zones beinhalten, werden nicht unterstützt. Diese Aktion unterstützt nicht die Definition von Zielen mithilfe von Ressourcen-Tags, Filtern oder Parametern.

Für den Auto Scaling LaunchInstances Scaling-Vorgang werden `InsufficientInstanceCapacity` Fehler im `errors` Antwortfeld zurückgegeben, aber die gewünschte Kapazität der Auto Scaling Scaling-Gruppe wird trotzdem aktualisiert, sodass der asynchrone Skalierungsprozess möglicherweise Instances starten kann. Für umfassendere Tests auf unzureichende Kapazitätsverwaltung mit sollten Sie erwägen LaunchInstances, diese Aktion zusammen mit [the section called “aws:ec2:asg-insufficient-instance-capacity-error”](#) zu verwenden.

Ressourcentyp

- `aws:iam:role`

Parameters

- `duration`— In der AWS FIS API ist der Wert eine Zeichenfolge im Format ISO 8601. `PT1M` steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- `availabilityZoneldentifiers`— Die durch Kommas getrennte Liste der Availability Zones. Unterstützt Zonen IDs (z. B. `"use1-az1, use1-az2"`) und Zonennamen (z. B. `"us-east-1a"`).
- `percentage`— Der Prozentsatz (1—100) der Aufrufe, in die der Fehler eingeschleust werden soll.

Berechtigungen

- `ec2:InjectApiError` wobei der `ec2:FisActionId` Wert des Bedingungsschlüssels auf `aws:ec2:api-insufficient-instance-capacity-error` und der `ec2:FisTargetArns` Bedingungsschlüssel auf die IAM-Zielrollen gesetzt ist.

Eine Beispielrichtlinie finden Sie unter [Beispiel: Verwenden Sie Bedingungsschlüssel für `ec2:InjectApiError`](#).

`aws:ec2:asg-insufficient-instance-capacity-error`

Fügt `InsufficientInstanceCapacity` Fehlerantworten auf Anfragen der Auto Scaling Scaling-Zielgruppen ein. Diese Aktion unterstützt nur Auto Scaling Scaling-Gruppen, die Startvorlagen verwenden. Weitere Informationen zu Fehlern bei unzureichender Instance-Kapazität finden Sie im [Amazon EC2 EC2-Benutzerhandbuch](#).

Ressourcentyp

- `aws:ec2:autoscaling-group`

Parameters

- `duration`— In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- `availabilityZonendefiners`— Die durch Kommas getrennte Liste der Availability Zones. Unterstützt Zonen IDs (z. B. "use1-az1, use1-az2") und Zonennamen (z. B. "us-east-1a").
- `percentage` – Optional. Der Prozentsatz (1—100) der Startanfragen der Auto Scaling Scaling-Zielgruppe zur Injection des Fehlers. Der Standardwert ist 100.

Berechtigungen

- `ec2:InjectApiError` mit Bedingungsschlüssel `ec2:FisActionId` Wert auf gesetzt `aws:ec2:asg-insufficient-instance-capacity-error` und `ec2:FisTargetArns` Bedingungsschlüssel auf Auto Scaling Scaling-Zielgruppen gesetzt.
- `autoscaling:DescribeAutoScalingGroups`

Eine Beispielrichtlinie finden Sie unter [Beispiel: Verwenden Sie Bedingungsschlüssel für `ec2:InjectApiError`](#).

`aws:ec2:reboot-instances`

Führt die Amazon EC2 EC2-API-Aktion [RebootInstances](#) auf den Ziel-EC2-Instances aus.

Ressourcentyp

- `aws:ec2:instance`

Parameters

- Keine

Berechtigungen

- `ec2:RebootInstances`
- `ec2:DescribeInstances`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEC2Access](#)

aws:ec2:send-spot-instance-interruptions

Unterbricht die Ziel-Spot-Instances. Sendet zwei Minuten vor [deren Unterbrechung eine Benachrichtigung](#) über eine Unterbrechung der Spot-Instances an die Ziel-Spot-Instances. Die Unterbrechungszeit wird durch den angegebenen `durationBeforeInterruptionParameter` bestimmt. Zwei Minuten nach der Unterbrechungszeit werden die Spot-Instances je nach ihrem Unterbrechungsverhalten beendet oder gestoppt. Eine Spot Instance, die von AWS FIS angehalten wurde, bleibt angehalten, bis Sie sie neu starten.

Unmittelbar nach dem Initiieren der Aktion erhält die Ziel-Instance eine Empfehlung zur [Neuverteilung der EC2-Instance](#). Wenn Sie dies angegeben haben `durationBeforeInterruption`, könnte es zu einer Verzögerung zwischen der Empfehlung zur Neuverteilung und der Benachrichtigung über die Unterbrechung kommen.

Weitere Informationen finden Sie unter [the section called “Testen Sie Spot-Instance-Unterbrechungen”](#). Alternativ können Sie das Experiment mithilfe der Amazon EC2 EC2-Konsole starten. Weitere Informationen finden Sie unter [Initiieren einer Spot-Instance-Unterbrechung](#) im Amazon EC2 EC2-Benutzerhandbuch.

Ressourcentyp

- `aws:ec2:spot-instance`

Parameters

- `durationBeforeInterruption`— Die Wartezeit, bevor die Instance unterbrochen wird, zwischen 2 und 15 Minuten. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT2M steht beispielsweise für zwei Minuten. In der AWS FIS Konsole geben Sie die Anzahl der Minuten ein.

Berechtigungen

- `ec2:SendSpotInstanceInterruptions`
- `ec2:DescribeInstances`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEC2Access](#)

aws:ec2:stop-instances

Führt die Amazon EC2 EC2-API-Aktion [StopInstances](#) auf den Ziel-EC2-Instances aus.

Ressourcentyp

- `aws:ec2:instance`

Parameters

- `startInstancesAfterDuration` – Optional. Die Wartezeit vor dem Starten der Instance, zwischen einer Minute und 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein. Wenn die Instance über ein verschlüsseltes EBS-Volume verfügt, müssen Sie dem KMS-Schlüssel, der zur Verschlüsselung des Volumes verwendet wird, die AWS FIS Erlaubnis erteilen oder die Experimentierrolle zur KMS-Schlüsselrichtlinie hinzufügen.
- `completeIfInstancesTerminated` – Optional. Wenn der Wert wahr ist und auch wahr `startInstancesAfterDuration` ist, schlägt diese Aktion nicht fehl, wenn EC2-Zielinstanzen durch eine separate Anfrage außerhalb von FIS beendet wurden und nicht neu gestartet werden können. Auto Scaling Scaling-Gruppen können beispielsweise gestoppte EC2-Instances unter ihrer Kontrolle beenden, bevor diese Aktion abgeschlossen ist. Der Standardwert lautet „false“.

Berechtigungen

- `ec2:StopInstances`
- `ec2:StartInstances`
- `ec2:DescribeInstances` – Optional. Erforderlich bei `completeIfInstancesTerminated`, um den Instanzstatus am Ende der Aktion zu überprüfen.

- `kms:CreateGrant` – Optional. Erforderlich mit `startInstancesAfterDuration`, um Instanzen mit verschlüsselten Volumes neu zu starten.

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEC2Access](#)

aws:ec2:terminate-instances

Führt die Amazon EC2 EC2-API-Aktion [TerminateInstances](#) auf den Ziel-EC2-Instances aus.

Ressourcentyp

- `aws:ec2:instance`

Parameters

- Keine

Berechtigungen

- `ec2:TerminateInstances`
- `ec2:DescribeInstances`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEC2Access](#)

Amazon ECS-Aktionen

AWS FIS unterstützt die folgenden Amazon ECS-Aktionen.

Aktionen

- [aws:ecs:drain-container-instances](#)
- [aws:ecs:stop-task](#)
- [aws:ecs:task-cpu-stress](#)

- [aws:ecs:task-io-stress](#)
- [aws:ecs:task-kill-process](#)
- [aws:ecs:task-network-blackhole-port](#)
- [aws:ecs:task-network-latency](#)
- [aws:ecs:task-network-packet-loss](#)

aws:ecs:drain-container-instances

Führt die Amazon ECS-API-Aktion aus [UpdateContainerInstancesState](#), um den angegebenen Prozentsatz der zugrunde liegenden Amazon EC2 EC2-Instances auf den Zielclustern zu entladen.

Ressourcentyp

- aws:ecs:cluster

Parameters

- **drainagePercentage**— Der Prozentsatz (1-100).
- **duration**— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `ecs:DescribeClusters`
- `ecs:UpdateContainerInstancesState`
- `ecs:ListContainerInstances`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorECSAccess](#)

aws:ecs:stop-task

Führt die Amazon ECS-API-Aktion aus [StopTask](#), um die Zielaufgabe zu beenden.

Ressourcentyp

- aws:ecs:task

Parameters

- Keine

Berechtigungen

- ecs:DescribeTasks
- ecs:ListTasks
- ecs:StopTask
- tag:GetResources

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorECSAccess](#)

aws:ecs:task-cpu-stress

Führt die CPU-Belastung für die Zielaufgaben aus. Verwendet das [AWSFIS-RunSSM-Dokument -CPU-Stress](#). Die Aufgaben müssen von verwaltet werden. AWS Systems Manager Weitere Informationen finden Sie unter [ECS-Aufgabenaktionen](#).

Ressourcentyp

- aws:ecs:task

Parameters

- duration— Die Dauer des Stresstests im Format ISO 8601.

- `percent` – Optional. Der Ziellastprozentsatz, von 0 (keine Last) bis 100 (Volllast). Der Standardwert ist 100.
- `workers` – Optional. Die Anzahl der zu verwendenden Stressoren. Die Standardeinstellung ist 0, wodurch alle Stressoren verwendet werden.
- `installDependencies` – Optional. Wenn dieser Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf dem Sidecar-Container für den SSM-Agenten, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeit ist `stress-ng`.

Berechtigungen

- `ecs:DescribeTasks`
- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-io-stress

Führt zu I/O Stress bei den Zielaufgaben. Verwendet das [AWSFIS-RunSSM-Dokument -IO-Stress](#). Die Aufgaben müssen verwaltet werden. AWS Systems Manager Weitere Informationen finden Sie unter [ECS-Aufgabenaktionen](#).

Ressourcentyp

- `aws:ecs:task`

Parameters

- `duration`— Die Dauer des Stresstests im Format ISO 8601.
- `percent` – Optional. Der Prozentsatz des freien Speicherplatzes im Dateisystem, der während des Stresstests verwendet werden soll. Die Standardeinstellung ist 80%.
- `workers` – Optional. Die Anzahl der Worker. Worker führen eine Mischung aus sequentiellen, zufälligen und speicherbezogenen read/write Vorgängen, erzwungener Synchronisation und Cache-Löschen aus. Mehrere untergeordnete Prozesse führen unterschiedliche I/O Operationen an derselben Datei aus. Der Standardwert ist 1.

- `installDependencies` – Optional. Wenn dieser Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf dem Sidecar-Container für den SSM-Agenten, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeit ist `stress-ng`

Berechtigungen

- `ecs:DescribeTasks`
- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-kill-process

Stoppt den angegebenen Prozess in den Aufgaben mithilfe des `killall` Befehls. Verwendet das [AWSFIS-RunSSM-Dokument -Kill-Process](#). Die Aufgabendefinition muss auf `pidMode task` eingestellt sein. Die Aufgaben müssen von verwaltet werden AWS Systems Manager. Weitere Informationen finden Sie unter [ECS-Aufgabenaktionen](#).

Ressourcentyp

- `aws:ecs:task`

Parameters

- `processName`— Der Name des Prozesses, der gestoppt werden soll.
- `signal` – Optional. Das Signal, das zusammen mit dem Befehl gesendet werden soll. Die möglichen Werte sind `SIGTERM` (die der Empfänger ignorieren kann) und `SIGKILL` (die nicht ignoriert werden können). Der Standardwert ist `SIGTERM`.
- `installDependencies` – Optional. Wenn dieser Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf dem Sidecar-Container für den SSM-Agenten, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeit ist `killall`

Berechtigungen

- `ecs:DescribeTasks`
- `ssm:SendCommand`

- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-network-blackhole-port

Löscht eingehenden oder ausgehenden Datenverkehr für das angegebene Protokoll und den angegebenen Port mithilfe der [Amazon ECS Fault Injection-Endpunkte](#). Verwendet das SSM-Dokument [AWSFIS-Run-Network-Blackhole-Port-ECS](#). Die Aufgabendefinition muss auf `bridge` eingestellt sein. `pidMode task` Die Aufgaben müssen von verwaltet werden AWS Systems Manager. Das können Sie `bridge` in `networkMode` der Aufgabendefinition nicht festlegen. Weitere Informationen finden Sie unter [ECS-Aufgabenaktionen](#).

Wenn auf `useEcsFaultInjectionEndpoints` `false` gesetzt, verwendet der Fehler das `iptables` Tool und das SSM-Dokument [AWSFIS-Run-Network-Blackhole-Port](#).

Ressourcentyp

- `aws:ecs:task`

Parameters

- `duration`— Die Dauer des Tests im Format ISO 8601.
- `port`— Die Portnummer.
- `trafficType`— Die Art des Datenverkehrs. Die möglichen Werte sind `ingress` und `egress`.
- `protocol` – Optional. Das Protokoll. Die möglichen Werte sind `tcp` und `udp`. Der Standardwert ist `tcp`.
- `installDependencies` – Optional. Wenn dieser Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf dem Sidecar-Container für den SSM-Agenten, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeiten sind `datd`, `undcurl-minimal`, `dig` und `jq`.
- `useEcsFaultInjectionEndpoints` – Optional. Wenn auf `true` gesetzt, APIs wird Amazon ECS Fault Injection verwendet. Der Standardwert lautet „`false`“.

Berechtigungen

- `ecs:DescribeTasks`

- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-network-latency

Fügt der Netzwerkschnittstelle Latenz und Jitter für ausgehenden Datenverkehr zu bestimmten Quellen hinzu, wobei die [Amazon ECS Fault Injection-Endpunkte](#) verwendet werden. Verwendet das SSM-Dokument [AWSFIS-Run-Network-Latency-ECS](#). Die Aufgabendefinition muss auf `task` eingestellt sein. Die Aufgaben müssen von verwaltet werden AWS Systems Manager. Das können Sie `bridge` in `networkMode` der Aufgabendefinition nicht festlegen. Weitere Informationen finden Sie unter [ECS-Aufgabenaktionen](#).

Wenn auf `useEcsFaultInjectionEndpoints` `false` ist, verwendet der Fehler das `tc` Tool und das SSM-Dokument [AWSFIS-Run-Network-Latency-SOURCES](#).

Verwenden Sie den `flowsPercent` Parameter, um die Latenz für einen bestimmten Prozentsatz der Verbindungen zu erhöhen. Um den `flowsPercent` Parameter verwenden zu können, muss die Version des ECS-Agenten `1.100.0` oder höher sein.

Um AZ-Namen oder AZ IDs im `sources` Parameter zu verwenden, müssen sich alle Ziele der Aktion auf derselben VPC befinden.

Ressourcentyp

- `aws:ecs:task`

Parameters

- `duration`— Die Dauer des Tests im ISO 8601-Format.
- `delayMilliseconds` – Optional. Die Verzögerung in Millisekunden. Die Standardeinstellung ist 200.
- `jitterMilliseconds` – Optional. Der Jitter in Millisekunden. Der Standardwert ist 10.
- `flowsPercent` – Optional. Der Prozentsatz der Netzwerkflüsse, die von der Aktion betroffen sein werden. Der Standardwert für ist 100 %.
- `sources` – Optional. Die Quellen, durch Kommas getrennt, ohne Leerzeichen. Die möglichen Werte sind: eine IPv4 Adresse, ein IPv4 CIDR-Block, ein Domainname, ein AZ-Name (`us-east-1a`), eine

AZ-ID (use1-az1), ALL, und. DYNAMODB S3 Wenn Sie DYNAMODB oder angebenS3, gilt dies nur für den regionalen Endpunkt in der aktuellen Region. Die Standardeinstellung ist ALL, was dem gesamten IPv4 Verkehr entspricht.

- `installDependencies` – Optional. Wenn dieser Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf dem Sidecar-Container für den SSM-Agenten, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeiten sind `datd`, `curl-minimal`, `dig` und `jq` `ls`
- `useEcsFaultInjectionEndpoints` – Optional. Wenn auf `true` gesetzt, APIs wird Amazon ECS Fault Injection verwendet. Der Standardwert lautet „`false`“.

Berechtigungen

- `ecs:DescribeTasks`
- `ecs:DescribeContainerInstances`
- `ec2:DescribeInstances`
- `ec2:DescribeSubnets`
- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-network-packet-loss

Fügt der Netzwerkschnittstelle Paketverlust für ausgehenden Datenverkehr zu bestimmten Quellen hinzu, wobei die [Amazon ECS Fault Injection-Endpunkte](#) verwendet werden. Verwendet das SSM-Dokument [AWSFIS-Run-Network-Packet-Loss-ECS](#). Die Aufgabendefinition muss auf `ingress` eingestellt sein. `pidMode task` Die Aufgaben müssen von verwaltet werden AWS Systems Manager. Das können Sie `bridge` in `networkMode` der Aufgabendefinition nicht festlegen. Weitere Informationen finden Sie unter [ECS-Aufgabenaktionen](#).

Wenn auf `egress` gesetzt `useEcsFaultInjectionEndpoints` ist `false`, verwendet der Fehler das `tc` Tool und das SSM-Dokument [AWSFIS-Run-Network-Packet-Loss-Sources](#).

Verwenden Sie den `flowsPercent` Parameter, um Paketverlust bei einem bestimmten Prozentsatz der Verbindungen zu verursachen. Um den `flowsPercent` Parameter verwenden zu können, muss die Version des ECS-Agenten `1.100.0` oder höher sein.

Um AZ-Namen oder AZ IDs im `sources` Parameter zu verwenden, müssen sich alle Ziele der Aktion auf derselben VPC befinden.

Ressourcentyp

- `aws:ecs:task`

Parameters

- `duration`— Die Dauer des Tests im ISO 8601-Format.
- `lossPercent` – Optional. Der Prozentsatz des Paketverlusts. Die Standardeinstellung ist 7%.
- `flowsPercent` – Optional. Der Prozentsatz der Netzwerkflüsse, die von der Aktion betroffen sein werden. Der Standardwert für ist 100 %.
- `sources` – Optional. Die Quellen, durch Kommas getrennt, ohne Leerzeichen. Die möglichen Werte sind: eine IPv4 Adresse, ein IPv4 CIDR-Block, ein Domainname, ein AZ-Name (us-east-1a), eine AZ-ID (use1-az1), ALL, und. DYNAMODB S3 Wenn Sie DYNAMODB oder angebenS3, gilt dies nur für den regionalen Endpunkt in der aktuellen Region. Die Standardeinstellung ist ALL, was dem gesamten IPv4 Verkehr entspricht.
- `installDependencies` – Optional. Wenn dieser Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf dem Sidecar-Container für den SSM-Agenten, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeiten sind `datd`, `curl-minimal`, `dig` und `jq` Isof
- `useEcsFaultInjectionEndpoints` – Optional. Wenn auf `true` gesetzt, APIs wird Amazon ECS Fault Injection verwendet. Der Standardwert lautet „false“.

Berechtigungen

- `ecs:DescribeTasks`
- `ecs:DescribeContainerInstances`
- `ec2:DescribeInstances`
- `ec2:DescribeSubnets`
- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

Amazon EKS-Aktionen

AWS FIS unterstützt die folgenden Amazon EKS-Aktionen.

Aktionen

- [aws:eks:inject-kubernetes-custom-resource](#)
- [aws:eks:pod-cpu-stress](#)
- [aws:eks:pod-delete](#)
- [aws:eks:pod-io-stress](#)
- [aws:eks:pod-memory-stress](#)
- [aws:eks:pod-network-blackhole-port](#)
- [aws:eks:pod-network-latency](#)
- [aws:eks:pod-network-packet-loss](#)
- [aws:eks:terminate-nodegroup-instances](#)

aws:eks:inject-kubernetes-custom-resource

Führt ein ChaosMesh oder Litmus-Experiment auf einem einzelnen Zielcluster aus. Sie müssen ChaosMesh oder Litmus auf dem Zielcluster installieren.

Wenn Sie eine Versuchsvorlage erstellen und ein Ziel vom Typ definieren `aws:eks:cluster`, müssen Sie diese Aktion auf einen einzelnen Amazon-Ressourcennamen (ARN) ausrichten. Diese Aktion unterstützt nicht die Definition von Zielen mithilfe von Ressourcen-Tags, Filtern oder Parametern.

Bei der Installation ChaosMesh müssen Sie die entsprechende Container-Laufzeit angeben. Ab Amazon EKS Version 1.23 wurde die Standardraufzeit von Docker auf geändert. containerd Ab Version 1.24 wurde Docker entfernt.

Ressourcentyp

- `aws:eks:cluster`

Parameters

- `kubernetesApiVersion`— Die API-Version der benutzerdefinierten [Kubernetes-Ressource](#). Die möglichen Werte sind `chaos-mesh.org/v1alpha1` | `litmuschaos.io/v1alpha1`
- `kubernetesKind`— Der benutzerdefinierte Kubernetes-Ressourcentyp. Der Wert hängt von der API-Version ab.
 - `chaos-mesh.org/v1alpha1`— Die möglichen Werte sind `AWSChaos` | `DNSChaos` | `GCPChaos` | `HTTPChaos` | `IOChaos` | `JVMChaos` | `KernelChaos` | `NetworkChaos` | `PhysicalMachineChaos` | `PodChaos` | `PodHttpChaos` | `PodIOChaos` | `PodNetworkChaos` | `Schedule` | `StressChaos` | `TimeChaos` |
- `litmuschaos.io/v1alpha1`— Der mögliche Wert ist `ChaosEngine`.
- `kubernetesNamespace`— Der [Kubernetes-Namespace](#).
- `kubernetesSpec`— Der spec Abschnitt der benutzerdefinierten Kubernetes-Ressource im JSON-Format.
- `maxDuration`— Die maximale Zeit, die für den Abschluss der Automatisierungsausführung zulässig ist, zwischen einer Minute und 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

Für diese Aktion sind keine AWS Identitäts- und Zugriffsverwaltungsberechtigungen (IAM) erforderlich. Die für die Verwendung dieser Aktion erforderlichen Berechtigungen werden von Kubernetes mithilfe der RBAC-Autorisierung gesteuert. Weitere Informationen finden Sie unter [Verwenden der RBAC-Autorisierung](#) in der offiziellen Kubernetes-Dokumentation. Weitere Informationen zu Chaos Mesh finden Sie in der [offiziellen](#) Chaos Mesh-Dokumentation. Weitere Informationen zu Litmus findest du in der [offiziellen Litmus-Dokumentation](#).

aws:eks:pod-cpu-stress

Führt die CPU-Belastung auf den Ziel-Pods aus. Weitere Informationen finden Sie unter [EKS-Pod-Aktionen](#).

Ressourcentyp

- `aws:eks:pod`

Parameters

- **duration**— Die Dauer des Stresstests im Format ISO 8601.
- **percent** – Optional. Der Ziellastprozentsatz, von 0 (keine Last) bis 100 (Volllast). Der Standardwert ist 100.
- **workers** – Optional. Die Anzahl der zu verwendenden Stressoren. Die Standardeinstellung ist 0, wodurch alle Stressoren verwendet werden.
- **kubernetesServiceAccount**— Das Kubernetes-Dienstkonto. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [the section called “Das Kubernetes-Servicekonto konfigurieren”](#).
- **fisPodContainerImage** – Optional. Das Container-Image, das zur Erstellung des Fault Injector-Pods verwendet wurde. Standardmäßig werden die von AWS FIS bereitgestellten Bilder verwendet. Weitere Informationen finden Sie unter [the section called “Pod-Container-Bilder”](#).
- **maxErrorsPercent** – Optional. Der Prozentsatz der Ziele, die ausfallen können, bevor die Fehlerinjektion fehlschlägt. Der Standardwert ist 0.
- **fisPodLabels** – Optional. Die Kubernetes-Labels, die an den von FIS erstellten Fault Orchestration Pod angehängt sind.
- **fisPodAnnotations** – Optional. Die Kubernetes-Anmerkungen, die an den von FIS erstellten Fehlerorchestrierungs-Pod angehängt sind.
- **fisPodSecurityPolicy** – Optional. Die Richtlinie der [Kubernetes-Sicherheitsstandards](#), die für den von FIS erstellten Fehlerorchestrierungs-Pod und die kurzlebigen Container verwendet werden soll. Mögliche Werte sind, und. `privileged` `baseline` `restricted` Diese Maßnahme ist mit allen politischen Ebenen vereinbar.

Berechtigungen

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEKSAccess](#)

aws:eks:pod-delete

Löscht die Ziel-Pods. Weitere Informationen finden Sie unter [EKS-Pod-Aktionen](#).

Ressourcentyp

- aws:eks:pod

Parameters

- `gracePeriodSeconds` – Optional. Die Wartezeit in Sekunden, bis der Pod ordnungsgemäß beendet wird. Wenn der Wert 0 ist, führen wir die Aktion sofort aus. Wenn der Wert Null ist, verwenden wir die Standard-Kulanzzeit für den Pod.
- `kubernetesServiceAccount`— Das Kubernetes-Dienstkonto. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [the section called “Das Kubernetes-Servicekonto konfigurieren”](#).
- `fisPodContainerImage` – Optional. Das Container-Image, das zur Erstellung des Fault Injector-Pods verwendet wurde. Standardmäßig werden die von AWS FIS bereitgestellten Bilder verwendet. Weitere Informationen finden Sie unter [the section called “Pod-Container-Bilder”](#).
- `maxErrorsPercent` – Optional. Der Prozentsatz der Ziele, die ausfallen können, bevor die Fehlerinjektion fehlschlägt. Der Standardwert ist 0.
- `fisPodLabels` – Optional. Die Kubernetes-Labels, die an den von FIS erstellten Fault Orchestration Pod angehängt sind.
- `fisPodAnnotations` – Optional. Die Kubernetes-Anmerkungen, die an den von FIS erstellten Fehlerorchestrierungs-Pod angehängt sind.
- `fisPodSecurityPolicy` – Optional. Die Richtlinie der [Kubernetes-Sicherheitsstandards](#), die für den von FIS erstellten Fehlerorchestrierungs-Pod und die kurzlebigen Container verwendet werden soll. Mögliche Werte sind, und. `privileged` `baseline` `restricted` Diese Maßnahme ist mit allen politischen Ebenen vereinbar.

Berechtigungen

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEKSAccess](#)

aws:eks:pod-io-stress

I/O stress die Ziel-Pods. Weitere Informationen finden Sie unter [EKS-Pod-Aktionen](#).

Ressourcentyp

- aws:eks:pod

Parameters

- duration— Die Dauer des Stresstests im Format ISO 8601.
- workers – Optional. Die Anzahl der Worker. Worker führen eine Mischung aus sequentiellen, zufälligen und speicherbezogenen read/write Vorgängen, erzwungener Synchronisation und Löschen des Caches durch. Mehrere untergeordnete Prozesse führen unterschiedliche I/O Operationen an derselben Datei aus. Der Standardwert ist 1.
- percent – Optional. Der Prozentsatz des freien Speicherplatzes im Dateisystem, der während des Stresstests verwendet werden soll. Die Standardeinstellung ist 80%.
- kubernetesServiceAccount— Das Kubernetes-Dienstkonto. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [the section called “Das Kubernetes-Servicekonto konfigurieren”](#).
- fisPodContainerImage – Optional. Das Container-Image, das zur Erstellung des Fault Injector-Pods verwendet wurde. Standardmäßig werden die von AWS FIS bereitgestellten Bilder verwendet. Weitere Informationen finden Sie unter [the section called “Pod-Container-Bilder”](#).
- maxErrorsPercent – Optional. Der Prozentsatz der Ziele, die ausfallen können, bevor die Fehlerinjektion fehlschlägt. Der Standardwert ist 0.
- fisPodLabels – Optional. Die Kubernetes-Labels, die an den von FIS erstellten Fault Orchestration Pod angehängt sind.
- fisPodAnnotations – Optional. Die Kubernetes-Anmerkungen, die an den von FIS erstellten Fehlerorchestrierungs-Pod angehängt sind.
- fisPodSecurityPolicy – Optional. Die Richtlinie der [Kubernetes-Sicherheitsstandards](#), die für den von FIS erstellten Fehlerorchestrierungs-Pod und die kurzlebigen Container verwendet werden soll.

Mögliche Werte sind, und. `privileged baseline restricted` Diese Maßnahme ist mit allen politischen Ebenen vereinbar.

Berechtigungen

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEKSAccess](#)

`aws:eks:pod-memory-stress`

Führt zu einer Speicherbelastung der Ziel-Pods. Weitere Informationen finden Sie unter [EKS-Pod-Aktionen](#).

Ressourcentyp

- `aws:eks:pod`

Parameters

- `duration`— Die Dauer des Stresstests im Format ISO 8601.
- `workers` – Optional. Die Anzahl der zu verwendenden Stressoren. Der Standardwert ist 1.
- `percent` – Optional. Der Prozentsatz des virtuellen Speichers, der während des Stresstests verwendet werden soll. Die Standardeinstellung ist 80%.
- `kubernetesServiceAccount`— Das Kubernetes-Dienstkonto. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [the section called “Das Kubernetes-Servicekonto konfigurieren”](#).
- `fisPodContainerImage` – Optional. Das Container-Image, das zur Erstellung des Fault Injector-Pods verwendet wurde. Standardmäßig werden die von AWS FIS bereitgestellten Bilder verwendet. Weitere Informationen finden Sie unter [the section called “Pod-Container-Bilder”](#).
- `maxErrorsPercent` – Optional. Der Prozentsatz der Ziele, die ausfallen können, bevor die Fehlerinjektion fehlschlägt. Der Standardwert ist 0.

- `fisPodLabels` – Optional. Die Kubernetes-Labels, die an den von FIS erstellten Fault Orchestration Pod angehängt sind.
- `fisPodAnnotations` – Optional. Die Kubernetes-Anmerkungen, die an den von FIS erstellten Fehlerorchestrierungs-Pod angehängt sind.
- `fisPodSecurityPolicy` – Optional. Die Richtlinie der [Kubernetes-Sicherheitsstandards](#), die für den von FIS erstellten Fehlerorchestrierungs-Pod und die kurzlebigen Container verwendet werden soll. Mögliche Werte sind, und. `privileged` `baseline` `restricted` Diese Maßnahme ist mit allen politischen Ebenen vereinbar.

Berechtigungen

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEKSAccess](#)

`aws:eks:pod-network-blackhole-port`

Löscht eingehenden oder ausgehenden Datenverkehr für das angegebene Protokoll und den angegebenen Port. Nur kompatibel mit der Richtlinie für [Kubernetes Security Standards](#). `privileged` Weitere Informationen finden Sie unter [EKS-Pod-Aktionen](#).

Ressourcentyp

- `aws:eks:pod`

Parameters

- `duration`— Die Dauer des Tests im ISO 8601-Format.
- `protocol`— Das Protokoll. Die möglichen Werte sind `tcp` und `udp`.
- `trafficType`— Die Art des Verkehrs. Die möglichen Werte sind `ingress` und `egress`.
- `port`— Die Portnummer.

- `kubernetesServiceAccount`— Das Kubernetes-Dienstkonto. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [the section called “Das Kubernetes-Servicekonto konfigurieren”](#).
- `fisPodContainerImage` – Optional. Das Container-Image, das zur Erstellung des Fault Injector-Pods verwendet wurde. Standardmäßig werden die von AWS FIS bereitgestellten Bilder verwendet. Weitere Informationen finden Sie unter [the section called “Pod-Container-Bilder”](#).
- `maxErrorsPercent` – Optional. Der Prozentsatz der Ziele, die ausfallen können, bevor die Fehlerinjektion fehlschlägt. Der Standardwert ist 0.
- `fisPodLabels` – Optional. Die Kubernetes-Labels, die an den von FIS erstellten Fault Orchestration Pod angehängt sind.
- `fisPodAnnotations` – Optional. Die Kubernetes-Anmerkungen, die an den von FIS erstellten Fehlerorchestrierungs-Pod angehängt sind.

Berechtigungen

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEKSAccess](#)

`aws:eks:pod-network-latency`

Fügt der Netzwerkschnittstelle Latenz und Jitter hinzu, indem das `tc` Tool für den Datenverkehr zu oder von bestimmten Quellen verwendet wird. Nur kompatibel mit der Richtlinie für [Kubernetes Security Standards](#). `privileged` Weitere Informationen finden Sie unter [EKS-Pod-Aktionen](#).

Verwenden Sie den `flowsPercent` Parameter, um die Latenz für einen bestimmten Prozentsatz der Verbindungen zu erhöhen.

Ressourcentyp

- `aws:eks:pod`

Parameters

- **duration**— Die Dauer des Tests im ISO 8601-Format.
- **interface** – Optional. Die Netzwerkschnittstellen, durch Kommas getrennt. Die Werte ALL und DEFAULT werden unterstützt. Die Standardeinstellung ist DEFAULT, die auf die primäre Netzwerkschnittstelle für das Betriebssystem abzielt.
- **delayMilliseconds** – Optional. Die Verzögerung in Millisekunden. Die Standardeinstellung ist 200.
- **jitterMilliseconds** – Optional. Der Jitter in Millisekunden. Der Standardwert ist 10.
- **flowsPercent** – Optional. Der Prozentsatz der Netzwerkflüsse, die von der Aktion betroffen sein werden. Der Standardwert für ist 100 %.
- **sources** – Optional. Die Quellen, durch Kommas getrennt, ohne Leerzeichen. Die möglichen Werte sind: eine IPv4 Adresse, ein IPv4 CIDR-Block, ein Domainname, ein AZ-Name (us-east-1a), eine AZ-ID (use1-az1), ALL, und. DYNAMODB S3 Wenn Sie DYNAMODB oder angebenS3, gilt dies nur für den regionalen Endpunkt in der aktuellen Region. Bei Domainnamen werden 10 DNS-Auflösungsversuche unternommen, um IP-Adressen zu sammeln. Aufgrund des DNS-Lastenausgleichs und der Rotation beeinträchtigt diese Aktion möglicherweise nicht alle möglichen IP-Adressen, zu denen die Domain aufgelöst werden könnte. Die Standardeinstellung ist ALL, was dem gesamten IPv4 Datenverkehr entspricht.
- **kubernetesServiceAccount**— Das Kubernetes-Dienstkonto. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [the section called “Das Kubernetes-Servicekonto konfigurieren”](#).
- **fisPodContainerImage** – Optional. Das Container-Image, das zur Erstellung des Fault Injector-Pods verwendet wurde. Standardmäßig werden die von AWS FIS bereitgestellten Bilder verwendet. Weitere Informationen finden Sie unter [the section called “Pod-Container-Bilder”](#).
- **maxErrorsPercent** – Optional. Der Prozentsatz der Ziele, die ausfallen können, bevor die Fehlerinjektion fehlschlägt. Der Standardwert ist 0.
- **fisPodLabels** – Optional. Die Kubernetes-Labels, die an den von FIS erstellten Fault Orchestration Pod angehängt sind.
- **fisPodAnnotations** – Optional. Die Kubernetes-Anmerkungen, die an den von FIS erstellten Fehlerorchestrierungs-Pod angehängt sind.

Berechtigungen

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`

- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEKSAccess](#)

aws:eks:pod-network-packet-loss

Fügt der Netzwerkschnittstelle mithilfe des tc Tools Paketverlust hinzu. Nur kompatibel mit der Richtlinie für [Kubernetes Security Standards](#) `privileged`. Weitere Informationen finden Sie unter [EKS-Pod-Aktionen](#).

Verwenden Sie den `flowsPercent` Parameter, um Paketverlust bei einem bestimmten Prozentsatz der Verbindungen zu verursachen.

Ressourcentyp

- `aws:eks:pod`

Parameters

- `duration`— Die Dauer des Tests im ISO 8601-Format.
- `interface` – Optional. Die Netzwerkschnittstellen, durch Kommas getrennt. Die Werte `ALL` und `DEFAULT` werden unterstützt. Die Standardeinstellung ist `DEFAULT`, die auf die primäre Netzwerkschnittstelle für das Betriebssystem abzielt.
- `lossPercent` – Optional. Der Prozentsatz des Paketverlusts. Die Standardeinstellung ist 7%.
- `flowsPercent` – Optional. Der Prozentsatz der Netzwerkflüsse, die von der Aktion betroffen sein werden. Der Standardwert für ist 100 %.
- `sources` – Optional. Die Quellen, durch Kommas getrennt, ohne Leerzeichen. Die möglichen Werte sind: eine IPv4 Adresse, ein IPv4 CIDR-Block, ein Domainname, ein AZ-Name (`us-east-1a`), eine AZ-ID (`use1-az1`), `ALL`, und `DYNAMODB S3`. Wenn Sie `DYNAMODB` oder angeben `S3`, gilt dies nur für den regionalen Endpunkt in der aktuellen Region. Bei Domainnamen werden 10 DNS-Auflösungsversuche unternommen, um IP-Adressen zu sammeln. Aufgrund des DNS-Lastenausgleichs und der Rotation beeinträchtigt diese Aktion möglicherweise nicht alle möglichen IP-Adressen, zu denen die Domain aufgelöst werden könnte. Die Standardeinstellung ist `ALL`, was dem gesamten IPv4 Datenverkehr entspricht.

- `kubernetesServiceAccount`— Das Kubernetes-Dienstkonto. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [the section called “Das Kubernetes-Servicekonto konfigurieren”](#).
- `fisPodContainerImage` – Optional. Das Container-Image, das zur Erstellung des Fault Injector-Pods verwendet wurde. Standardmäßig werden die von AWS FIS bereitgestellten Bilder verwendet. Weitere Informationen finden Sie unter [the section called “Pod-Container-Bilder”](#).
- `maxErrorsPercent` – Optional. Der Prozentsatz der Ziele, die ausfallen können, bevor die Fehlerinjektion fehlschlägt. Der Standardwert ist 0.
- `fisPodLabels` – Optional. Die Kubernetes-Labels, die an den von FIS erstellten Fault Orchestration Pod angehängt sind.
- `fisPodAnnotations` – Optional. Die Kubernetes-Anmerkungen, die an den von FIS erstellten Fehlerorchestrierungs-Pod angehängt sind.

Berechtigungen

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEKSAccess](#)

`aws:eks:terminate-nodegroup-instances`

Führt die Amazon EC2 EC2-API-Aktion für [TerminateInstances](#) die Zielknotengruppe aus. Nur kompatibel mit von Amazon EKS verwalteten Knotengruppen. Selbstverwaltete Knotengruppen werden nicht unterstützt. Weitere Informationen finden Sie unter [EKS Manage Compute](#).

Ressourcentyp

- `aws:eks:nodegroup`

Parameters

- `instanceTerminationPercentage`— Der Prozentsatz (1—100) der zu beendenden Instances.

Berechtigungen

- `ec2:DescribeInstances`
- `ec2:TerminateInstances`
- `eks:DescribeNodegroup`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEKSAccess](#)

ElastiCache Amazon-Aktionen

AWS FIS unterstützt die folgende ElastiCache Aktion.

`aws:elasticache:replicationgroup-interrupt-az-power`

Unterbricht die Stromversorgung von Knoten in der angegebenen Availability Zone für ElastiCache Zielreplikationsgruppen, für die Multi-AZ aktiviert ist. Es kann jeweils nur eine Availability Zone pro Replikationsgruppe betroffen sein. Wenn ein primärer Knoten als Ziel ausgewählt wird, wird die entsprechende Read Replica mit der geringsten Replikationsverzögerung zum primären Knoten heraufgestuft. Read Replica-Ersetzungen in der angegebenen Availability Zone werden für die Dauer dieser Aktion blockiert, was bedeutet, dass Zielreplikationsgruppen mit reduzierter Kapazität arbeiten. Das Ziel für diese Aktion unterstützt sowohl Redis- als auch Valkey-Engines. Die Aktion unterstützt die Bereitstellungsoption „serverlos“ nicht.

Ressourcentyp

- `aws:elasticache:replicationgroup`

Parameters

- `duration`— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `elasticache:InterruptClusterAzPower`
- `elasticache:DescribeReplicationGroups`
- `tag:GetResources`

Note

Die AZ-Power-Aktion „ElastiCache Interrupt“ unterstützt jetzt alle Typen von Replikationsgruppen, einschließlich Valkey und Redis. Um diese Funktionalität besser darzustellen, wurde die Aktion umbenannt. Wenn Sie derzeit die Aktion verwenden `aws:elasticache:interrupt-cluster-az-power`, empfehlen wir Ihnen, auf die neue Aktion zu migrieren `aws:elasticache:replicationgroup-interrupt-az-power`, um die neuesten Funktionen nutzen zu können.

Amazon Kinesis Data Streams Streams-Aktionen

Amazon Kinesis Data Streams unterstützt die folgenden Kinesis-Aktionen.

Aktionen

- [aws:kinesis:stream-provisioned-throughput-exception](#)
- [aws:kinesis:stream-expired-iterator-exception](#)

aws:kinesis:stream-provisioned-throughput-exception

Fügt `ProvisionedThroughputExceededException` Fehlerantworten auf Anfragen für die Ziel-Kinesis Data Streams ein. Zu den unterstützten Vorgängen gehören: `GetRecords`, `GetShardIteratorPutRecord`, und `PutRecords`

Ressourcentyp

- `aws:kinesis:stream`

Parameters

- **Dauer** — Die Dauer, die zwischen einer Minute und 12 Stunden liegt. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- **Prozentsatz** — Der Prozentsatz (1—100) der Aufrufe, bei denen der Fehler ausgelöst werden soll.

Berechtigungen

- `kinesis:InjectApiError`

aws:kinesis:stream-expired-iterator-exception

Fügt `ExpiredIteratorException` Fehlerantworten für `GetRecords` Aufrufe ein, die auf bestimmte Kinesis Data Streams abzielen.

Ressourcentyp

- `aws:kinesis:stream`

Parameters

- **Dauer** — Die Dauer, die zwischen einer Minute und 12 Stunden liegt. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- **Prozentsatz** — Der Prozentsatz (1—100) der Aufrufe, bei denen der Fehler ausgelöst werden soll.

Berechtigungen

- `kinesis:InjectApiError`

AWS Lambda Aktionen

AWS Lambda unterstützt die folgenden Lambda-Aktionen

Aktionen

- [aws:lambda:invocation-add-delay](#)

- [aws:lambda:invocation-error](#)
- [aws:lambda:invocation-http-integration-response](#)

aws:lambda:invocation-add-delay

Verzögert den Start einer Funktion um eine von Ihnen angegebene Anzahl von Millisekunden. Die Wirkung dieser Aktion ähnelt Lambda-Kaltstarts, aber die zusätzliche Zeit wird als Teil der abgerechneten Dauer aufgewendet und auf alle Ausführungsumgebungen angewendet, anstatt sich nur auf neue Ausführungsumgebungen auszuwirken. Dies bedeutet, dass sowohl ein Lambda-Kaltstart als auch diese Verzögerung auftreten können. Wenn Sie einen Latenzwert festlegen, der höher ist als das in der Lambda-Funktion konfigurierte Timeout, ermöglicht diese Aktion auch den Zugriff auf ein Timeout-Ereignis mit hoher Genauigkeit.

Ressourcentyp

- `aws:lambda:function`

Parameters

- **Dauer** — Die Dauer der Aktion. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- **InvocationPercentage** — Optional. Der Prozentsatz (1—100) der Funktionsaufrufen, in die der Fehler eingefügt werden soll. Der Standardwert ist 100.
- **startupDelayMilliseconds** – Optional. Die Zeit in Millisekunden (0-900.000), die zwischen dem Aufruf und der Ausführung des Funktionscodes gewartet werden soll. Der Standardwert ist 1000.

Berechtigungen

- `s3:PutObject`
- `s3>DeleteObject`
- `lambda:GetFunction`
- `tag:GetResources`

aws:lambda:invocation-error

Markiert Lambda-Funktionsaufrufen als fehlgeschlagen. Diese Aktion ist nützlich, um Mechanismen zur Fehlerbehandlung zu testen, z. B. Alarme und Wiederholungskonfigurationen. Während Sie diese Aktion verwenden, wählen Sie aus, ob der Funktionscode ausgeführt werden soll, bevor ein Fehler zurückgegeben wird.

Ressourcentyp

- `aws:lambda:function`

Parameters

- **Dauer** — Die Dauer der Aktion. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- **InvocationPercentage** — Optional. Der Prozentsatz (1—100) der Funktionsaufrufen, in die der Fehler eingefügt werden soll. Der Standardwert ist 100.
- **preventExecution** — Wenn der Wert wahr ist, gibt die Aktion den Fehler zurück, ohne die Funktion auszuführen.

Berechtigungen

- `s3:PutObject`
- `s3>DeleteObject`
- `lambda:GetFunction`
- `tag:GetResources`

aws:lambda:invocation-http-integration-response

Ändert das Verhalten der Funktion. Sie wählen einen Inhaltstyp und einen HTTP-Antwortcode aus, um Integrationen mit ALB, API-GW und VPC Lattice zu unterstützen. Um die selektive Auswirkung auf Upstream- oder Downstream-Integrationen zu ermöglichen, können Sie wählen, ob Sie die geänderte Antwort direkt zurückgeben oder ob Sie die Funktion ausführen und die Antwort ersetzen möchten, nachdem die Ausführung der Funktion abgeschlossen ist.

Ressourcentyp

- `aws:lambda:function`

Parameters

- `contentTypeHeader`— Zeichenkettenwert des HTTP-Inhaltstyp-Headers, der von der Lambda-Funktion zurückgegeben werden soll.
- `Dauer` — Die Dauer der Aktion. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- `InvocationPercentage` — Optional. Der Prozentsatz (1—100) der Funktionsaufrufen, in die der Fehler eingefügt werden soll. Der Standardwert ist 100.
- `preventExecution` — Wenn der Wert wahr ist, gibt die Aktion die Antwort zurück, ohne die Funktion auszuführen.
- `statusCode` — Wert des HTTP-Statuscodes (000-999), der von der Lambda-Funktion zurückgegeben werden soll.

Berechtigungen

- `s3:PutObject`
- `s3>DeleteObject`
- `lambda:GetFunction`
- `tag:GetResources`

Amazon MemoryDB-Aktion

AWS FIS unterstützt die folgende MemoryDB-Aktion.

`aws:memorydb:multi-region-cluster-pause-replication`

Unterbricht die Replizierung zwischen einem regionalen Cluster und allen anderen regionalen Clustern innerhalb des Clusters mit mehreren Regionen. Der angestrebte regionale Cluster ist der Cluster in der Region, in der das FIS-Experiment läuft. Solange die Replizierung angehalten ist, kann der Multiregion-Cluster nicht aktualisiert werden. Sobald die Aktion abgeschlossen ist, kann

es einige Minuten dauern, bis der Cluster mit mehreren Regionen wieder verfügbar ist. Weitere Informationen zu Amazon MemoryDB Multi-Region finden Sie im Amazon [MemoryDB Multi-Region Developer Guide](#). [Informationen zur regionalen Verfügbarkeit finden Sie unter Voraussetzungen und Einschränkungen von MemoryDB Multi-Region.](#)

Ressourcentyp

- `aws:memorydb:multi-region-cluster`

Parameters

- `duration`— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `memorydb:DescribeMultiRegionClusters`
- `memorydb:PauseMultiRegionClusterReplication`
- `tag:GetResources`

Netzwerkaktionen

AWS FIS unterstützt die folgenden Netzwerkaktionen.

Aktionen

- [aws:network:disrupt-connectivity](#)
- [aws:network:route-table-disrupt-cross-region-connectivity](#)
- [aws:network:transit-gateway-disrupt-cross-region-connectivity](#)
- [aws:network:disrupt-vpc-endpoint](#)

aws:network:disrupt-connectivity

Verweigert den angegebenen Datenverkehr zu den Zielsubnetzen, indem vorübergehend die ursprüngliche Netzwerkzugriffskontrollliste (Network Access Control List, Netzwerk-ACL) geklont wird, die dem Zielsubnetz zugeordnet ist. FIS fügt der geklonten Netzwerk-ACL, die über das Tag

managedByFIS=true verfügt, Ablehnungsregeln hinzu und ordnet sie für die Dauer der Aktion dem Subnetz zu. Nach Abschluss der Aktion löscht FIS die geklonte Netzwerk-ACL und stellt die ursprüngliche Netzwerk-ACL-Zuordnung wieder her.

Ressourcentyp

- aws:ec2:subnet

Parameters

- scope— Die Art des Datenverkehrs, der verweigert werden soll. Wenn der Bereich nicht angegeben ist `all`, beträgt die maximale Anzahl von Einträgen im Netzwerk ACLs 20. Die möglichen Werte sind:
 - `all`— Verweigert den gesamten Datenverkehr, der in das Subnetz eingeht und es verlässt. Beachten Sie, dass diese Option subnetzinternen Verkehr zulässt, einschließlich Verkehr zu und von Netzwerkschnittstellen im Subnetz.
 - `availability-zone`— Verweigert internen VPC-Verkehr zu und von Subnetzen in anderen Availability Zones. Die maximale Anzahl von Subnetzen, die in einer VPC als Ziel ausgewählt werden können, ist 30.
 - `dynamodb`— Verweigert den Verkehr zum und vom regionalen Endpunkt für DynamoDB in der aktuellen Region.
 - `prefix-list`— Verweigert den Verkehr zur und von der angegebenen Präfixliste.
 - `s3`— Verweigert den Verkehr zum und vom regionalen Endpunkt für Amazon S3 in der aktuellen Region.
 - `s3express`— Verweigert den Verkehr zum und vom zonalen Endpunkt für Amazon S3 Express One Zone in der AZ der Zielsubnetze. Die Zielsubnetze müssen sich dort befinden, AZs wo S3 Express One Zone derzeit verfügbar ist. Weitere Informationen finden Sie unter [S3 Express One Zone Availability Zones and Regions](#).
 - `vpc`— Verweigert den Verkehr, der in die VPC eingeht und sie verlässt.
- duration— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- prefixListIdentifier— Bei einem Gültigkeitsbereich handelt es `prefix-list` sich um den Bezeichner der vom Kunden verwalteten Präfixliste. Sie können einen Namen, eine ID oder einen ARN angeben. Die Präfixliste kann maximal 10 Einträge enthalten.

Berechtigungen

- `ec2:CreateNetworkAcl`— Erzeugt die Netzwerk-ACL mit dem Tag `managedByFis=True`.
- `ec2:CreateNetworkAclEntry`— Die Netzwerk-ACL muss das Tag `ManagedByFis=True` haben.
- `ec2:CreateTags`
- `ec2>DeleteNetworkAcl`— Die Netzwerk-ACL muss das Tag `ManagedByFis=True` haben.
- `ec2:DescribeManagedPrefixLists`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `ec2:GetManagedPrefixListEntries`
- `ec2:ReplaceNetworkAclAssociation`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorNetworkAccess](#)

aws:network:route-table-disrupt-cross-region-connectivity

Blockiert den Datenverkehr, der seinen Ursprung in den Zielsubnetzen hat und für die angegebene Region bestimmt ist. Erstellt Routentabellen, die alle Routen für die zu isolierende Region enthalten. Damit FIS diese Routing-Tabellen erstellen kann, erhöhen Sie das Amazon VPC-Kontingent für `routes per route table` auf 250 (oder 350, wenn der `region` Parameter `us-east-1` ist) zuzüglich der Anzahl der Routen in Ihren vorhandenen Routentabellen.

Ressourcentyp

- `aws:ec2:subnet`

Parameters

- `region`— Der Code der zu isolierenden Region (zum Beispiel `eu-west-1`).
- `duration`— Die Dauer der Aktion. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. `PT1M` steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `ec2:AssociateRouteTable`
- `ec2:CreateManagedPrefixList †`
- `ec2:CreateNetworkInterface †`
- `ec2:CreateRoute †`
- `ec2:CreateRouteTable †`
- `ec2:CreateTags †`
- `ec2>DeleteManagedPrefixList †`
- `ec2>DeleteNetworkInterface †`
- `ec2>DeleteRouteTable †`
- `ec2:DescribeManagedPrefixLists`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcPeeringConnections`
- `ec2:DescribeVpcs`
- `ec2:DisassociateRouteTable`
- `ec2:GetManagedPrefixListEntries`
- `ec2:ModifyManagedPrefixList †`
- `ec2:ModifyVpcEndpoint`
- `ec2:ReplaceRouteTableAssociation`

† Mit dem Tag abgegrenzt. `managedByFIS=true` Sie müssen dieses Tag nicht verwalten. AWS FIS fügt dieses Tag während des Experiments hinzu und entfernt es.

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorNetworkAccess](#)

aws:network:transit-gateway-disrupt-cross-region-connectivity

Blockiert den Datenverkehr vom Ziel-Transit-Gateway-Peering-Anhang, der für die angegebene Region bestimmt ist.

Ressourcentyp

- `aws:ec2:transit-gateway`

Parameters

- `region`— Der Code der zu isolierenden Region (zum Beispiel eu-west-1).
- `duration`— Die Dauer der Aktion. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `ec2:AssociateTransitGatewayRouteTable`
- `ec2:DescribeTransitGatewayAttachments`
- `ec2:DescribeTransitGatewayPeeringAttachments`
- `ec2:DescribeTransitGateways`
- `ec2:DisassociateTransitGatewayRouteTable`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorNetworkAccess](#)

aws:network:disrupt-vpc-endpoint

Blockiert eingehenden und ausgehenden Datenverkehr der VPC-Endpunkte der Zielschnittstelle. FIS erstellt eine verwaltete Sicherheitsgruppe mit leeren Regeln und ersetzt vorübergehend die Sicherheitsgruppen der Ziel-VPC-Endpunkte durch diese verwaltete Sicherheitsgruppe. Wenn während der Ausführung der Aktion Änderungen an den Zielressourcen vorgenommen werden, schlägt die Aktion fehl und die Ressourcen werden nicht auf ihren Zustand vor dem Experiment

zurückgesetzt. Wenn eine von FIS verwaltete Sicherheitsgruppe während der Ausführung der Aktion geändert wird, wird sie außerdem nicht von FIS gelöscht.

Ressourcentyp

- `aws:ec2:vpc-endpoint`

Parameters

- `duration`— Die Dauer der Aktion. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `ec2:DescribeVpcEndpoints`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyVpcEndpoint`
- `ec2:CreateSecurityGroup`
- `ec2>DeleteSecurityGroup`
- `ec2:RevokeSecurityGroupEgress`
- `ec2:CreateTags`
- `vpce:AllowMultiRegion` *

* Die Genehmigung ist nur erforderlich, wenn Sie auf regionsübergreifende VPC-Endpunkte abzielen

Amazon RDS-Aktionen

AWS FIS unterstützt die folgenden Amazon RDS-Aktionen.

Aktionen

- [`aws:rds:failover-db-cluster`](#)
- [`aws:rds:reboot-db-instances`](#)

aws:rds:failover-db-cluster

Führt die Amazon RDS-API-Aktion [Failover DBCluster](#) auf dem Aurora-DB-Zielcluster aus. RDS-Cluster und DocumentDB-Cluster werden unterstützt.

Ressourcentyp

- `aws:rds:cluster`

Parameters

- Keine

Berechtigungen

- `rds:FailoverDBCluster`
- `rds:DescribeDBClusters`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorRDSAccess](#)

aws:rds:reboot-db-instances

Führt die Amazon RDS-API-Aktion [Reboot DBInstance](#) auf der Ziel-DB-Instance aus. RDS-Cluster und DocumentDB-Cluster werden unterstützt.

Ressourcentyp

- `aws:rds:db`

Parameters

- `forceFailover` – Optional. Wenn der Wert wahr ist und wenn es sich bei den Instances um Multi-AZ handelt, wird ein Failover von einer Availability Zone zur anderen erzwungen. Der Standardwert lautet „false“.

Berechtigungen

- `rds:RebootDBInstance`
- `rds:DescribeDBInstances`
- `tag:GetResources`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorRDSAccess](#)

Amazon-S3-Aktionen

AWS FIS unterstützt die folgende Amazon S3 S3-Aktion.

Aktionen

- [aws:s3:bucket-pause-replication](#)

aws:s3:bucket-pause-replication

Unterbricht die Replikation von den Ziel-Quell-Buckets zu den Ziel-Buckets. Ziel-Buckets können sich in anderen AWS-Regionen oder in derselben Region wie der Quell-Bucket befinden. Bestehende Objekte können bis zu einer Stunde nach Beginn der Aktion weiter repliziert werden. Diese Aktion unterstützt nur das Targeting nach Tags. Weitere Informationen zu Amazon S3 Replication finden Sie im [Amazon S3 S3-Benutzerhandbuch](#).

Ressourcentyp

- `aws:s3:bucket`

Parameters

- `duration`— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.
- `region`— Die AWS-Region, in der sich Ziel-Buckets befinden.
- `destinationBuckets` – Optional. Durch Kommas getrennte Liste von Ziel-S3-Buckets.

- prefixes – Optional. Durch Kommas getrennte Liste von S3-Objektschlüsselpräfixen aus Replikationsregelfiltern. Die Replikationsregeln von Ziel-Buckets mit einem Filter, der auf dem/den Präfix (en) basiert, werden angehalten.

Berechtigungen

- S3:PutReplicationConfiguration wobei der Bedingungsschlüssel auf gesetzt ist S3:IsReplicationPauseRequest True
- S3:GetReplicationConfiguration wobei der Bedingungsschlüssel S3:IsReplicationPauseRequest auf eingestellt ist True
- S3:PauseReplication
- S3:ListAllMyBuckets
- tag:GetResources

Eine Beispielrichtlinie finden Sie unter [Beispiel: Verwenden Sie Bedingungsschlüssel für aws:s3:bucket-pause-replication](#).

Systems Manager Manager-Aktionen

AWS FIS unterstützt die folgenden Systems Manager Manager-Aktionen.

Aktionen

- [aws:ssm:send-command](#)
- [aws:ssm:start-automation-execution](#)

aws:ssm:send-command

Führt die Systems Manager Manager-API-Aktion [SendCommand](#) auf den EC2-Zielinstanzen aus. Das Systems Manager-Dokument (SSM-Dokument) definiert die Aktionen, die Systems Manager auf Ihren Instances ausführt. Weitere Informationen finden Sie unter [Verwenden Sie die Aktion aws:ssm:send-command](#).

Ressourcentyp

- aws:ec2:instance

Parameters

- **documentArn**— Der Amazon-Ressourcenname (ARN) des Dokuments. In der Konsole wird dieser Parameter für Sie vervollständigt, wenn Sie unter Aktionstyp einen Wert auswählen, der einem der [vorkonfigurierten AWS FIS SSM-Dokumente](#) entspricht.
- **documentVersion** – Optional. Die Version des Dokuments. Wenn leer, wird die Standardversion ausgeführt.
- **documentParameters**— Befristet. Die erforderlichen und optionalen Parameter, die das Dokument akzeptiert. Das Format ist ein JSON-Objekt mit Schlüsseln, die Zeichenfolgen sind, und Werten, die entweder Zeichenketten oder Zeichenkettenarrays sind.
- **duration**— Die Dauer, von einer Minute bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorEC2Access](#)

aws:ssm:start-automation-execution

Führt die Systems Manager Manager-API-Aktion aus [StartAutomationExecution](#).

Ressourcentyp

- Keine

Parameters

- **documentArn**— Der Amazon-Ressourcenname (ARN) des Automatisierungsdokuments.

- `documentVersion` – Optional. Die Version des Dokuments. Wenn leer, wird die Standardversion ausgeführt.
- `documentParameters`— Befristet. Die erforderlichen und optionalen Parameter, die das Dokument akzeptiert. Das Format ist ein JSON-Objekt mit Schlüsseln, die Zeichenfolgen sind, und Werten, die entweder Zeichenketten oder Zeichenkettenarrays sind.
- `maxDuration`— Die maximale Zeit, die für den Abschluss der Automatisierungsausführung zulässig ist, zwischen einer Minute und 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT1M steht beispielsweise für eine Minute. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `ssm:StopAutomationExecution`
- `iam:PassRole` – Optional. Erforderlich, wenn das Automatisierungsdokument eine Rolle annimmt.

AWS verwaltete Richtlinie

- [AWSFaultInjectionSimulatorSSMAccess](#)

AWS Direct Connect Aktionen

AWS FIS unterstützt die folgende AWS Direct Connect Aktion.

Aktionen

- [aws:directconnect:virtual-interface-disconnect](#)

aws:directconnect:virtual-interface-disconnect

Testet die Belastbarkeit der AWS Direct Connect Verbindung, indem die Border Gateway Protocol (BGP) -Sitzungen zwischen den lokalen Netzwerken und Peers, die den virtuellen Zielschnittstellen () zugeordnet sind, vorübergehend unterbrochen werden. VIFs Vor Beginn des Experiments überprüft FIS, ob sich alle VIFs Zielpersonen des Experiments im Status „verfügbar“ befinden und dass jede VIF über alle BGP-Peers mit dem Status „verfügbar“ und dem BGP-Status „aktiv“ verfügt. Während

des Experiments werden die BGP-Peering-Sitzungen für die ausgewählten virtuellen Schnittstellen in den Status „Inaktiv“ versetzt. Ausführliche Informationen zu Direct Connect-Failover-Tests finden Sie in der [AWS Direct Connect Dokumentation](#).

Ressourcentyp

- `aws:directconnect:virtual-interface`

Parameters

- `duration`— Die Dauer, von 10 Minuten bis 12 Stunden. In der AWS FIS API ist der Wert eine Zeichenfolge im ISO 8601-Format. PT10M steht beispielsweise für zehn Minuten. In der AWS FIS-Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein.

Berechtigungen

- `directconnect:DescribeVirtualInterfaces`
- `directconnect:StartBgpFailoverTest`
- `directconnect:ListVirtualInterfaceTestHistory`
- `directconnect:StopBgpFailoverTest`
- `tag:GetResources`

Verwenden Sie Systems Manager SSM-Dokumente mit AWS FIS

AWS FIS unterstützt benutzerdefinierte Fehlertypen über den AWS Systems Manager SSM-Agenten und die AWS FIS-Aktion. [aws:ssm:send-command](#) Vorkonfigurierte Systems Manager Manager-SSM-Dokumente (SSM-Dokumente), mit denen allgemeine Fehlerinjektionsaktionen erstellt werden können, sind als öffentliche AWS Dokumente verfügbar, die mit dem AWSFIS Präfix - beginnen.

SSM Agent ist Amazon-Software, die auf Amazon EC2-Instances, lokalen Servern oder virtuellen Maschinen () installiert und konfiguriert werden kann. VMs Dadurch kann Systems Manager diese Ressourcen verwalten. Der Agent verarbeitet Anfragen von Systems Manager und führt sie dann wie in der Anfrage angegeben aus. Sie können Ihr eigenes SSM-Dokument hinzufügen, um benutzerdefinierte Fehler einzufügen, oder auf eines der öffentlichen Dokumente verweisen, die sich im Besitz von Amazon befinden.

Voraussetzungen

Bei Aktionen, bei denen der SSM-Agent die Aktion auf dem Ziel ausführen muss, müssen Sie Folgendes sicherstellen:

- Der Agent ist auf dem Ziel installiert. SSM Agent ist standardmäßig auf einigen Amazon Machine Images (AMIs) installiert. Andernfalls können Sie den SSM-Agent auf Ihren Instances installieren. Weitere Informationen finden Sie unter [Manuelles Installieren des SSM-Agenten für EC2-Instances](#) im AWS Systems Manager Benutzerhandbuch.
- Systems Manager ist berechtigt, Aktionen auf Ihren Instances durchzuführen. Sie gewähren Zugriff mithilfe eines IAM-Instanzprofils. Weitere Informationen finden Sie im Benutzerhandbuch unter [Erstellen eines IAM-Instanzprofils für Systems Manager](#) und [Anhängen eines IAM-Instanzprofils an eine EC2-Instance](#). AWS Systems Manager

Verwenden Sie die Aktion `aws:ssm:send-command`

Ein SSM-Dokument definiert die Aktionen, die Systems Manager auf Ihren verwalteten Instances durchführt. Systems Manager enthält eine Reihe vorkonfigurierter Dokumente, Sie können aber auch eigene Dokumente erstellen. Weitere Informationen zum Erstellen Ihres eigenen SSM-Dokuments finden Sie unter [Erstellen von Systems Manager Manager-Dokumenten](#) im AWS Systems Manager Benutzerhandbuch. Weitere Informationen zu SSM-Dokumenten im Allgemeinen finden Sie unter [AWS Systems Manager Dokumente](#) im AWS Systems Manager Benutzerhandbuch.

AWS FIS stellt vorkonfigurierte SSM-Dokumente zur Verfügung. [Sie können die vorkonfigurierten SSM-Dokumente unter Dokumente in der Konsole einsehen: Dokumente. AWS Systems Manager](#) <https://console.aws.amazon.com/systems-manager/> In der FIS-Konsole können Sie auch aus einer Auswahl vorkonfigurierter Dokumente wählen. AWS Weitere Informationen finden Sie unter [Vorkonfigurierte AWS FIS SSM-Dokumente](#).

Um ein SSM-Dokument in Ihren AWS FIS-Experimenten zu verwenden, können Sie die Aktion verwenden. [aws:ssm:send-command](#) Diese Aktion ruft das angegebene SSM-Dokument ab und führt es auf Ihren Zielinstanzen aus.

Wenn Sie die `aws:ssm:send-command` Aktion in Ihrer Experimentvorlage verwenden, müssen Sie zusätzliche Parameter für die Aktion angeben, darunter die folgenden:

- `documentArn` – Erforderlich. Der Amazon-Ressourcenname (ARN) des SSM-Dokuments.

- `documentParameters`— Befriedigend. Die erforderlichen und optionalen Parameter, die das SSM-Dokument akzeptiert. Das Format ist ein JSON-Objekt mit Schlüsseln, die Zeichenketten sind, und Werten, die entweder Zeichenketten oder Zeichenkettenarrays sind.
- `documentVersion` – Optional. Die Version des SSM-Dokuments, das ausgeführt werden soll.

Sie können die Informationen für ein SSM-Dokument (einschließlich der Parameter für das Dokument) mithilfe der Systems Manager Manager-Konsole oder der Befehlszeile anzeigen.

Um Informationen zu einem SSM-Dokument mithilfe der Konsole anzuzeigen

1. Öffnen Sie die AWS Systems Manager Konsole unter. <https://console.aws.amazon.com/systems-manager/>
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie das Dokument und dann die Registerkarte Details aus.

Um Informationen zu einem SSM-Dokument über die Befehlszeile anzuzeigen

Verwenden Sie den SSM-Befehl [describe-document](#).

Erfahren Sie mehr über den Aktionsstatus

Der SSM-Aktionsstatus wird durch den [SSM-Befehlsstatus](#) bestimmt.

Vorkonfigurierte AWS FIS SSM-Dokumente

Sie können vorkonfigurierte AWS FIS SSM-Dokumente mit der Aktion in Ihren Experimentvorlagen verwenden. `aws:ssm:send-command`

Voraussetzungen

- Die von AWS FIS bereitgestellten vorkonfigurierten SSM-Dokumente werden nur auf den folgenden Betriebssystemen unterstützt:
 - Amazon Linux 2023, Amazon Linux 2
 - Ubuntu
 - RHEL 8, 9
 - CentOS 9

- Die von AWS FIS bereitgestellten vorkonfigurierten SSM-Dokumente werden nur auf EC2-Instances unterstützt. Sie werden auf anderen Arten von verwalteten Knoten, wie z. B. lokalen Servern, nicht unterstützt.

Um diese SSM-Dokumente in Experimenten mit ECS-Aufgaben zu verwenden, verwenden Sie die entsprechenden Dokumente. [the section called “Amazon ECS-Aktionen”](#) Die `aws:ecs:task-cpu-stress` Aktion verwendet beispielsweise das `AWSFIS-Run-CPU-Stress` Dokument.

Dokumente

- [AWSFIS-Run-CPU-Stress](#)
- [AWSFIS-Run-Disk-Fill](#)
- [AWSFIS-Run-IO-Stress](#)
- [AWSFIS-Run-Kill-Process](#)
- [AWSFIS-Run-Memory-Stress](#)
- [AWSFIS-Run-Network-Blackhole-Port](#)
- [AWSFIS-Run-Network-Latency](#)
- [AWSFIS-Run-Network-Latency-Sources](#)
- [AWSFIS-Run-Network-Packet-Loss](#)
- [AWSFIS-Run-Network-Packet-Loss-Sources](#)

Unterschied zwischen der Dauer der Aktion und `DurationSeconds` den in AWS FIS SSM-Dokumenten

Einige SSM-Dokumente beschränken ihre eigene Ausführungszeit, zum Beispiel wird der `DurationSeconds` Parameter von einigen der vorkonfigurierten AWS FIS SSM-Dokumente verwendet. Daher müssen Sie in der FIS-Aktionsdefinition zwei unabhängige Dauern angeben: AWS

- **Action duration:** Bei Experimenten mit einer einzigen Aktion entspricht die Aktionsdauer der Versuchsdauer. Bei mehreren Aktionen hängt die Versuchsdauer von der Dauer der einzelnen Aktionen und der Reihenfolge ab, in der sie ausgeführt werden. AWS FIS überwacht jede Aktion, bis ihre Aktionsdauer abgelaufen ist.
- **DokumentparameterDurationSeconds:** Die in Sekunden angegebene Dauer, für die das SSM-Dokument ausgeführt wird.

Sie können unterschiedliche Werte für die beiden Arten von Dauer wählen:

- **Action duration exceeds DurationSeconds:** Die Ausführung des SSM-Dokuments wird abgeschlossen, bevor die Aktion abgeschlossen ist. AWS FIS wartet, bis die Aktionsdauer abgelaufen ist, bevor nachfolgende Aktionen gestartet werden.
- **Action duration is shorter than DurationSeconds:** Das SSM-Dokument setzt die Ausführung fort, nachdem die Aktion abgeschlossen ist. Wenn die Ausführung des SSM-Dokuments noch läuft und die Aktionsdauer abgelaufen ist, wird der Aktionsstatus auf Abgeschlossen gesetzt. AWS FIS überwacht die Ausführung nur, bis die Aktionsdauer abgelaufen ist.

Beachten Sie, dass einige SSM-Dokumente eine variable Dauer haben. AWS FIS SSM-Dokumente bieten beispielsweise die Option, erforderliche Komponenten zu installieren, wodurch die Gesamtdauer der Ausführung über den angegebenen Parameter hinaus verlängert werden kann. **DurationSeconds** Wenn Sie also die Aktionsdauer **DurationSeconds** auf denselben Wert festlegen, ist es möglich, dass das SSM-Skript länger als die Aktionsdauer ausgeführt wird.

AWSFIS-Run-CPU-Stress

Führt mithilfe des `stress-ng` Tools die CPU-Belastung auf einer Instanz aus. Verwendet das [AWSFIS-RunSSM-Dokument -CPU-Stress](#).

Aktionstyp (nur Konsole)

`aws:ssm:send-command/AWSFIS-Run-CPU-Stress`

ARN

`arn:aws:ssm:region::document/AWSFIS-Run-CPU-Stress`

Parameter des Dokuments

- **DurationSeconds** – Erforderlich. Die Dauer des CPU-Stresstests in Sekunden.
- **CPU** – Optional. Die Anzahl der zu verwendenden CPU-Stressoren. Die Standardeinstellung ist 0, wodurch alle CPU-Stressoren verwendet werden.
- **LoadPercent** – Optional. Der Prozentsatz der CPU-Ziellast, von 0 (keine Last) bis 100 (Volllast). Der Standardwert ist 100.
- **InstallDependencies** – Optional. Wenn der Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeit ist `stress-ng`.

Das Folgende ist ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"DurationSeconds":"60", "InstallDependencies":"True"}
```

AWSFIS-Run-Disk-Fill

Weist Festplattenspeicher auf dem Root-Volume einer Instanz zu, um zu simulieren, dass die Festplatte voll ist. Verwendet das [AWSFIS-RunSSM-Dokument -Disk-Fill](#).

Wenn das Experiment, bei dem dieser Fehler ausgelöst wurde, entweder manuell oder durch eine Stopp-Bedingung beendet wird, versucht AWS FIS, ein Rollback durchzuführen, indem es das laufende SSM-Dokument abbricht. Wenn die Festplatte jedoch zu 100% voll ist, entweder aufgrund des Fehlers oder des Fehlers und der Anwendungsaktivität, kann Systems Manager den Abbruchvorgang möglicherweise nicht abschließen. Wenn Sie das Experiment möglicherweise beenden müssen, stellen Sie daher sicher, dass die Festplatte nicht zu 100% voll ist.

Aktionstyp (nur Konsole)

aws:ssm:send-command/AWSFIS-Run-Disk-Fill

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Disk-Fill

Parameter des Dokuments

- **DurationSeconds** – Erforderlich. Die Dauer des Festplattenfülltests in Sekunden.
- **Percent** – Optional. Der Prozentsatz der Festplatte, der während des Festplattenfülltests zugewiesen werden soll. Die Standardeinstellung ist 95%.
- **InstallDependencies** – Optional. Wenn der Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeiten sind `datd`, `kmod` und `fallocate`.

Das Folgende ist ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"DurationSeconds":"60", "InstallDependencies":"True"}
```


AWSFIS-Run-IO-Stress

Führt mithilfe des stress-ng Tools I/O-Stress auf einer Instanz aus. Verwendet das [AWSFIS-RunSSM-Dokument -IO-Stress](#).

Aktionstyp (nur Konsole)

aws:ssm:send-command/AWSFIS-Run-IO-Stress

ARN

arn:aws:ssm:region::document/AWSFIS-Run-IO-Stress

Parameter des Dokuments

- **DurationSeconds** – Erforderlich. Die Dauer des IO-Stresstests in Sekunden.
- **Workers** – Optional. Die Anzahl der Worker, die eine Mischung aus sequentiellen, zufälligen und speicherbezogenen read/write Vorgängen, erzwungener Synchronisation und Cache-Löschen ausführen. Mehrere untergeordnete Prozesse führen unterschiedliche I/O Operationen an derselben Datei aus. Der Standardwert ist 1.
- **Percent** – Optional. Der Prozentsatz des freien Speicherplatzes auf dem Dateisystem, der während des IO-Stresstests verwendet werden soll. Die Standardeinstellung ist 80%.
- **InstallDependencies** – Optional. Wenn der Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeit ist `stress-ng`.

Das Folgende ist ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"Workers":"1", "Percent":"80", "DurationSeconds":"60", "InstallDependencies":"True"}
```

AWSFIS-Run-Kill-Process

Stoppt den angegebenen Prozess in der Instanz mithilfe des killall Befehls. Verwendet das [AWSFIS-RunSSM-Dokument -Kill-Process](#).

Aktionstyp (nur Konsole)

aws:ssm:send-command/AWSFIS-Run-Kill-Process

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Kill-Process

Parameter des Dokuments

- **ProcessName** – Erforderlich. Der Name des Prozesses, der gestoppt werden soll.
- **Signal** – Optional. Das Signal, das zusammen mit dem Befehl gesendet werden soll. Die möglichen Werte sind `SIGTERM` (die der Empfänger ignorieren kann) und `SIGKILL` (die nicht ignoriert werden können). Der Standardwert ist `SIGTERM`.
- **InstallDependencies** – Optional. Wenn der Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeit ist `killall`.

Das Folgende ist ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"ProcessName":"myapplication", "Signal":"SIGTERM"}
```

AWSFIS-Run-Memory-Stress

Führt eine Speicherbelastung auf einer Instanz aus, die das `stress-ng` Tool verwendet. Verwendet das [AWSFIS-RunSSM-Dokument -Memory-Stress](#).

Aktionstyp (nur Konsole)

aws:ssm:send-command/AWSFIS-Run-Memory-Stress

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Memory-Stress

Parameter des Dokuments

- **DurationSeconds** – Erforderlich. Die Dauer des Speicherstresstests in Sekunden.
- **Workers** – Optional. Die Anzahl der Stressfaktoren für den virtuellen Speicher. Der Standardwert ist `1`.
- **Percent** – Erforderlich. Der Prozentsatz des virtuellen Speichers, der während des Speicherbelastungstests verwendet werden soll.

- **InstallDependencies** – Optional. Wenn der Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeit ist `stress-ng`.

Das Folgende ist ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"Percent":"80", "DurationSeconds":"60", "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Blackhole-Port

Löscht eingehenden oder ausgehenden Datenverkehr für das Protokoll und den Port mithilfe des `iptables` Tools. Verwendet das [AWSFIS-RunSSM-Dokument -Network-Blackhole-Port](#).

Aktionstyp (nur Konsole)

`aws:ssm:send-command/AWSFIS-Run-Network-Blackhole-Port`

ARN

`arn:aws:ssm:region::document/AWSFIS-Run-Network-Blackhole-Port`

Parameter des Dokuments

- **Protocol** – Erforderlich. Das Protokoll. Die möglichen Werte sind `tcp` und `udp`.
- **Port** – Erforderlich. Die Port-Nummer.
- **TrafficType** – Optional. Der Typ des Datenverkehrs. Die möglichen Werte sind `ingress` und `egress`. Der Standardwert ist `ingress`.
- **DurationSeconds** – Erforderlich. Die Dauer des Netzwerk-Blackhole-Tests in Sekunden.
- **InstallDependencies** – Optional. Wenn der Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeiten sind `atddig`, `lsof`, und `iptables`.

Im Folgenden finden Sie ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"Protocol":"tcp", "Port":"8080", "TrafficType":"egress", "DurationSeconds":"60",  
  "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Latency

Fügt der Netzwerkschnittstelle mithilfe des tc Tools Latenz hinzu. Verwendet das [AWSFIS-RunSSM-Dokument -Network-Latency](#).

Aktionstyp (nur Konsole)

aws:ssm:send-command/AWSFIS-Run-Network-Latency

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Network-Latency

Parameter des Dokuments

- Interface – Optional. Die Netzwerkschnittstelle. Der Standardwert ist `eth0`.
- DelayMilliseconds – Optional. Die Verzögerung in Millisekunden. Die Standardeinstellung ist 200.
- DurationSeconds – Erforderlich. Die Dauer des Netzwerklatenztests in Sekunden.
- InstallDependencies – Optional. Wenn der Wert lautet `True`, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen, sofern sie nicht bereits installiert sind. Der Standardwert ist `True`. Die Abhängigkeiten sind `atddig`, und `tc`.

Im Folgenden finden Sie ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"DelayMilliseconds":"200", "Interface":"eth0", "DurationSeconds":"60",  
  "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Latency-Sources

Fügt der Netzwerkschnittstelle Latenz und Jitter hinzu, indem das tc Tool für den Datenverkehr zu oder von bestimmten Quellen verwendet wird. Verwendet das SSM-Dokument [AWSFIS-Run-Network-Latency-Sources](#).

Verwenden Sie den `FlowsPercent` Parameter, um die Latenz für einen bestimmten Prozentsatz der Verbindungen zu erhöhen.

Aktionstyp (nur Konsole)

aws:ssm:send-command/AWSFIS-Run-Network-Latency-Sources

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Network-Latency-Sources

Parameter des Dokuments

- **Interface** – Optional. Die Netzwerkschnittstellen, durch Kommas getrennt. Die Werte ALL und DEFAULT werden unterstützt. Die Standardeinstellung ist DEFAULT, die auf die primäre Netzwerkschnittstelle für das Betriebssystem abzielt.
- **DelayMilliseconds** – Optional. Die Verzögerung in Millisekunden. Die Standardeinstellung ist 200.
- **JitterMilliseconds** – Optional. Der Jitter in Millisekunden. Der Standardwert ist 10.
- **FlowsPercent** – Optional. Der Prozentsatz der Netzwerkflüsse, die von der Aktion betroffen sein werden. Der Standardwert für ist 100 %.
- **Sources** – Erforderlich. Die Quellen, durch Kommas getrennt, ohne Leerzeichen. Die möglichen Werte sind: eine IPv4 Adresse, ein IPv4 CIDR-Block, ein Domainname, ein AZ-Name (us-east-1a), eine AZ-ID (use1-az1), ALL, und. DYNAMODB S3 Wenn Sie DYNAMODB oder angeben S3, gilt dies nur für den regionalen Endpunkt in der aktuellen Region.
- **TrafficType** – Optional. Der Typ des Datenverkehrs. Die möglichen Werte sind ingress und egress. Der Standardwert ist ingress.
- **DurationSeconds** – Erforderlich. Die Dauer des Netzwerklatenztests in Sekunden.
- **InstallDependencies** – Optional. Wenn der Wert lautet True, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen, sofern sie nicht bereits installiert sind. Der Standardwert ist True. Die Abhängigkeiten lauten atddig,jq,lsf, und tc.

Wenn Sie dieses Dokument verwenden, sind für die Rolle „Experiment“ die folgenden Berechtigungen erforderlich:

- `ec2:DescribeInstances`
- `ec2:DescribeSubnets`

Das Folgende ist ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"DelayMilliseconds":"200", "JitterMilliseconds":"15",  
  "Sources":"S3,www.example.com,72.21.198.67", "Interface":"eth0",  
  "TrafficType":"egress", "DurationSeconds":"60", "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Packet-Loss

Fügt der Netzwerkschnittstelle mithilfe des tc Tools Paketverlust hinzu. Verwendet das [AWSFIS-RunSSM-Dokument -Network-Packet-Loss](#).

Aktionstyp (nur Konsole)

aws:ssm:send-command/AWSFIS-Run-Network-Packet-Loss

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Network-Packet-Loss

Parameter des Dokuments

- Interface – Optional. Die Netzwerkschnittstelle. Der Standardwert ist eth0.
- LossPercent – Optional. Der Prozentsatz des Paketverlusts. Die Standardeinstellung ist 7%.
- DurationSeconds – Erforderlich. Die Dauer des Tests zum Verlust von Netzwerkpaketen in Sekunden.
- InstallDependencies – Optional. Wenn der Wert lautet True, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen. Der Standardwert ist True. Die Abhängigkeiten sind atdlsf,dig, undtc.

Im Folgenden finden Sie ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"LossPercent":"15", "Interface":"eth0", "DurationSeconds":"60",  
  "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Packet-Loss-Sources

Fügt der Netzwerkschnittstelle Paketverlust hinzu, indem das tc Tool für den Datenverkehr zu oder von bestimmten Quellen verwendet wird. Verwendet das SSM-Dokument [AWSFIS-Run-Network-Packet-Loss-Sources](#).

Verwenden Sie den FlowsPercent Parameter, um Paketverlust bei einem bestimmten Prozentsatz der Verbindungen zu verursachen.

Aktionstyp (nur Konsole)

aws:ssm:send-command/AWSFIS-Run-Network-Packet-Loss-Sources

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Network-Packet-Loss-Sources

Parameter des Dokuments

- **Interface** – Optional. Die Netzwerkschnittstellen, durch Kommas getrennt. Die Werte ALL und DEFAULT werden unterstützt. Die Standardeinstellung ist DEFAULT, die auf die primäre Netzwerkschnittstelle für das Betriebssystem abzielt.
- **LossPercent** – Optional. Der Prozentsatz des Paketverlusts. Die Standardeinstellung ist 7%.
- **FlowsPercent** – Optional. Der Prozentsatz der Netzwerkflüsse, die von der Aktion betroffen sein werden. Der Standardwert für ist 100 %.
- **Sources** – Erforderlich. Die Quellen, durch Kommas getrennt, ohne Leerzeichen. Die möglichen Werte sind: eine IPv4 Adresse, ein IPv4 CIDR-Block, ein Domainname, ein AZ-Name (us-east-1a), eine AZ-ID (use1-az1), ALL, und. DYNAMODB S3 Wenn Sie DYNAMODB oder angeben S3, gilt dies nur für den regionalen Endpunkt in der aktuellen Region.
- **TrafficType** – Optional. Der Typ des Datenverkehrs. Die möglichen Werte sind ingress und egress. Der Standardwert ist ingress.
- **DurationSeconds** – Erforderlich. Die Dauer des Tests zum Verlust von Netzwerkpaketen in Sekunden.
- **InstallDependencies** – Optional. Wenn der Wert lautet True, installiert Systems Manager die erforderlichen Abhängigkeiten auf den Zielinstanzen. Der Standardwert ist True. Die Abhängigkeiten sind atddig,jq,ls, und tc.

Wenn Sie dieses Dokument verwenden, sind für die Rolle „Experiment“ die folgenden Berechtigungen erforderlich:

- `ec2:DescribeInstances`
- `ec2:DescribeSubnets`

Das Folgende ist ein Beispiel für die Zeichenfolge, die Sie in die Konsole eingeben können.

```
{"LossPercent": "15", "Sources": "S3,www.example.com,72.21.198.67", "Interface": "eth0", "TrafficType": "egress", "DurationSeconds": "60", "InstallDependencies": "True"}
```

Beispiele

Ein Beispiel für eine Versuchsvorlage finden Sie unter [the section called “Führen Sie ein vorkonfiguriertes AWS FIS SSM-Dokument aus”](#).

Ein Tutorial finden Sie unter [CPU-Belastung auf einer Instanz ausführen](#).

Einschränkungen

- Die folgenden Dokumente können nicht parallel ausgeführt werden:
 - AWSFIS-Run-Network-Blackhole-Port
 - AWSFIS-Run-Network-Latency
 - AWSFIS-Run-Network-Latency-Sources
 - AWSFIS-Run-Network-Packet-Loss
 - AWSFIS-Run-Network-Packet-Loss-Sources

Rollback-Skripten

AWS FIS SSM-Dokumente erstellen automatisch Rollback-Skripte als Sicherheitsmechanismus zur Wiederherstellung des Systemstatus nach Experimenten mit Fehlerinjektion. Diese Skripts stellen sicher, dass eingefügte Fehler behoben werden, auch wenn die Aktion fehlschlägt oder unerwartet beendet wird.

Erstellung eines Rollback-Skripts

Rollback-Skripte werden automatisch erstellt, wenn mit Experimenten zur Fehlerinjektion begonnen wird.

Einzelheiten zur Erstellung

- Speicherort — Skripte werden im `/var/lib/amazon/ssm/` Verzeichnis erstellt.
- Benennungsmuster — `FAULT_NAME-FAULT_IDENTIFIER-Rollback.sh` wo `FAULT_IDENTIFIER` ist eine zufällig generierte 32-stellige Zeichenfolge
- Timing — Wird zu Beginn jedes Fault-Injection-Experiments erstellt, bevor die Fehlerinjektion beginnt.
- Inhalt — Enthält alle erforderlichen Umgebungsvariablen und Befehle, um den spezifischen Fehler zu beheben.

Bei einem Experiment zur Netzwerklatenz könnte beispielsweise ein Rollback-Skript unter `/var/lib/amazon/ssm/NetworkLatency-abc123-Rollback.sh` erstellt werden.

Rollback-Protokollierung

Rollback-Skripts implementieren eine duale Protokollierung, um alle Rollback-Aktivitäten zu Problembehebungs- und Prüfungszwecken zu erfassen.

Speicherorte der Protokolldateien

Wenn ein Rollback-Skript ausgeführt wird, erstellt es Protokolle an zwei Speicherorten:

- Temporäre Dateien — `/tmp/aws-fis-rollback-TIMESTAMP-PID.log`
- Systemprotokolle — Mit der Einrichtung an das Syslog gesendet `local0.info`

Benennung von Protokolldateien

Temporäre Protokolldateien verwenden die folgende Benennungskonvention:

```
/tmp/aws-fis-rollback-YYYY-MM-DDTHH:MM:SSZ-PID.log
```

Wo *YYYY-MM-DDTHH:MM:SSZ* ist der UTC-Zeitstempel und *PID* ist die Prozess-ID des Rollback-Skripts.

Syslog-Konfiguration

Rollback-Logs werden mit der folgenden Konfiguration an Syslog gesendet:

- Tag — `aws-fis-rollback`
- Priorität — `local0.info`
- Format — `[YYYY-MM-DDTHH:MM:SSZ] log_message`

Um Rollback-Logs anzuzeigen

Verwenden Sie den folgenden Befehl, um alle Rollback-Logs aus dem Systemd-Journal anzuzeigen:

```
sudo journalctl -t aws-fis-rollback
```

Fehlerbehebung

Gehen Sie wie folgt vor, um Probleme zu beheben.

Um Probleme mit SSM-Dokumenten zu beheben

1. Öffnen Sie die AWS Systems Manager Konsole unter. <https://console.aws.amazon.com/systems-manager/>
2. Wählen Sie im Navigationsbereich Node Management, Run Command aus.
3. Verwenden Sie auf der Registerkarte Befehlsverlauf die Filter, um die Ausführung des Dokuments zu ermitteln.
4. Wählen Sie die ID des Befehls, um die zugehörige Detailseite zu öffnen.
5. Wählen Sie die ID der Instanz. Überprüfen Sie die Ausgabe und die Fehler für jeden Schritt.

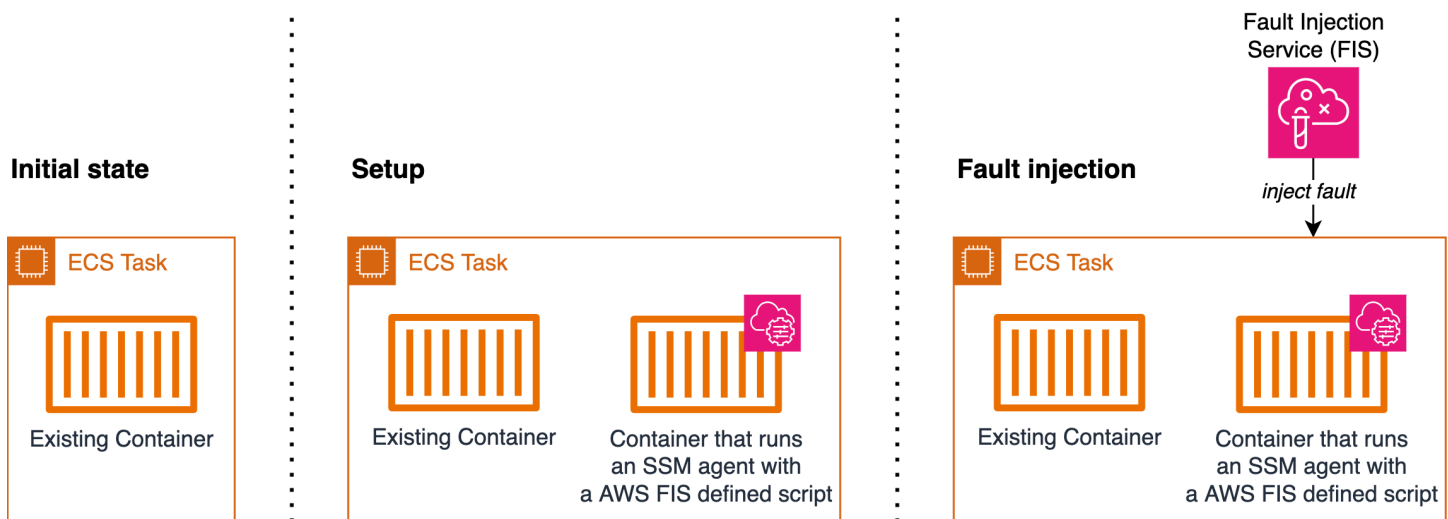
Verwenden Sie die AWS FIS-Aktionen `aws:ecs:task`

Sie können die `aws:ecs:task`-Aktionen verwenden, um Fehler in Ihre Amazon ECS-Aufgaben einzufügen. Die Kapazitätstypen Amazon EC2 und Fargate werden unterstützt.

Diese Aktionen verwenden [AWS Systems Manager \(SSM\) -Dokumente](#), um Fehler einzufügen. Um `aws:ecs:task` Aktionen verwenden zu können, müssen Sie Ihrer Amazon Elastic Container Service (Amazon ECS) -Aufgabendefinition einen Container mit einem SSM-Agenten hinzufügen. Der Container führt ein [AWS FIS-definiertes Skript](#) aus, das die Amazon ECS-Aufgabe als verwaltete Instance im SSM-Service registriert. Darüber hinaus ruft das Skript Aufgabenmetadaten ab, um der verwalteten Instance Tags hinzuzufügen. Das Setup ermöglicht es AWS FIS, die Zielaufgabe zu lösen. Dieser Absatz bezieht sich auf das Setup in der Abbildung unten.

Wenn Sie ein AWS FIS-Experiment-Targeting ausführen `aws:ecs:task`, ordnet AWS FIS die Amazon ECS-Zielaufgaben, die Sie in einer AWS FIS-Experimentvorlage angeben, mithilfe eines Ressourcen-Tags, einer Gruppe von SSM-verwalteten Instances zu. `ECS_TASK_ARN` Der Tag-Wert ist der ARN der zugehörigen Amazon ECS-Aufgabe, in der die SSM-Dokumente ausgeführt werden sollen. Dieser Absatz bezieht sich auf die Fault Injection in der Abbildung unten.

Das folgende Diagramm veranschaulicht die Einrichtung und die Fehlerinjektion bei einer Aufgabe mit einem vorhandenen Container.



Aktionen

- [the section called “aws:ecs:task-cpu-stress”](#)
- [the section called “aws:ecs:task-io-stress”](#)
- [the section called “aws:ecs:task-kill-process”](#)
- [the section called “aws:ecs:task-network-blackhole-port”](#)
- [the section called “aws:ecs:task-network-latency”](#)
- [the section called “aws:ecs:task-network-packet-loss”](#)

Einschränkungen

- Die folgenden Aktionen können nicht parallel ausgeführt werden:
 - aws:ecs:task-network-blackhole-port
 - aws:ecs:task-network-latency
 - aws:ecs:task-network-packet-loss
- Wenn Sie Amazon ECS Exec aktiviert haben, müssen Sie es deaktivieren, bevor Sie diese Aktionen verwenden können.
- Die Ausführung des SSM-Dokuments hat möglicherweise den Status Abgebrochen, auch wenn das Experiment den Status Abgeschlossen hat. Bei der Ausführung von Amazon ECS-Aktionen wird die vom Kunden angegebene Dauer sowohl für die Aktionsdauer im Experiment als auch für die Dauer des Amazon EC2 Systems Manager (SSM) -Dokuments verwendet. Nachdem die Aktion initiiert wurde, dauert es einige Zeit, bis das SSM-Dokument ausgeführt wird. Daher kann

es sein, dass dem SSM-Dokument bis zum Erreichen der angegebenen Aktionsdauer noch einige Sekunden verbleiben, um die Ausführung abzuschließen. Wenn die Aktionsdauer des Experiments erreicht ist, wird die Aktion gestoppt und die Ausführung des SSM-Dokuments abgebrochen. Die Fehlerinjektion war erfolgreich.

Voraussetzungen

- Fügen Sie der AWS [FIS-Experimentrolle](#) die folgenden Berechtigungen hinzu:
 - `ecs:DescribeTasks`
 - `ssm:SendCommand`
 - `ssm:ListCommands`
 - `ssm:CancelCommand`
- Fügen Sie der Amazon [ECS-Aufgaben-IAM-Rolle](#) die folgenden Berechtigungen hinzu:
 - `ssm:CreateActivation`
 - `ssm:AddTagsToResource`
 - `iam:PassRole`

Beachten Sie, dass Sie den ARN der verwalteten Instanzrolle als Ressource für `iam:PassRole` angeben können.

- Erstellen Sie eine [IAM-Rolle für die Ausführung von Amazon ECS-Aufgaben](#) und fügen Sie die von [Amazon ECSTask ExecutionRolePolicy](#) verwaltete Richtlinie hinzu.
- Stellen Sie in der Aufgabendefinition die Umgebungsvariable `MANAGED_INSTANCE_ROLE_NAME` auf den Namen der [verwalteten Instance-Rolle](#) ein. Dies ist die Rolle, die den Aufgaben zugewiesen wird, die als verwaltete Instanzen in SSM registriert sind.
- Fügen Sie der Rolle „Verwaltete Instanz“ die folgenden Berechtigungen hinzu:
 - `ssm>DeleteActivation`
 - `ssm:DeregisterManagedInstance`
- Fügen Sie die von [Amazon SSManaged InstanceCore](#) verwaltete Richtlinie zur Rolle der verwalteten Instanz hinzu.
- Fügen Sie der Amazon ECS-Aufgabendefinition einen SSM-Agent-Container hinzu. Das Befehlsskript registriert Amazon ECS-Aufgaben als verwaltete Instances.

```
{  
  "name": "amazon-ssm-agent",
```

```

    "image": "public.ecr.aws/amazon-ssm-agent/amazon-ssm-agent:latest",
    "cpu": 0,
    "links": [],
    "portMappings": [],
    "essential": false,
    "entryPoint": [],
    "command": [
        "/bin/bash",
        "-c",
        "set -e; dnf upgrade -y; dnf install jq procps awscli -y; term_handler()
{ echo \"Deleting SSM activation $ACTIVATION_ID\"; if ! aws ssm delete-
activation --activation-id $ACTIVATION_ID --region $ECS_TASK_REGION; then
echo \"SSM activation $ACTIVATION_ID failed to be deleted\" 1>&2; fi;
MANAGED_INSTANCE_ID=$(jq -e -r .ManagedInstanceID /var/lib/amazon/ssm/registration);
echo \"Deregistering SSM Managed Instance $MANAGED_INSTANCE_ID\"; if ! aws
ssm deregister-managed-instance --instance-id $MANAGED_INSTANCE_ID --region
$ECS_TASK_REGION; then echo \"SSM Managed Instance $MANAGED_INSTANCE_ID
failed to be deregistered\" 1>&2; fi; kill -SIGTERM $SSM_AGENT_PID; }; trap
term_handler SIGTERM SIGINT; if [[ -z $MANAGED_INSTANCE_ROLE_NAME ]]; then
echo \"Environment variable MANAGED_INSTANCE_ROLE_NAME not set, exiting\"
1>&2; exit 1; fi; if ! ps ax | grep amazon-ssm-agent | grep -v grep > /dev/
null; then if [[ -n $ECS_CONTAINER_METADATA_URI_V4 ]]; then echo \"Found ECS
Container Metadata, running activation with metadata\"; TASK_METADATA=$(curl
\"${ECS_CONTAINER_METADATA_URI_V4}/task\"); ECS_TASK_AVAILABILITY_ZONE=$(echo
$TASK_METADATA | jq -e -r '.AvailabilityZone'); ECS_TASK_ARN=$(echo $TASK_METADATA
| jq -e -r '.TaskARN'); ECS_TASK_REGION=$(echo $ECS_TASK_AVAILABILITY_ZONE | sed
's/.$//'); ECS_TASK_AVAILABILITY_ZONE_REGEX='^(af|ap|ca|cn|eu|me|sa|us|us-gov)-
(central|north|(north(east|west))|south|south(east|west)|east|west)-[0-9]{1}[a-z]
{1}$'; if ! [[ $ECS_TASK_AVAILABILITY_ZONE =~ $ECS_TASK_AVAILABILITY_ZONE_REGEX ]];
then echo \"Error extracting Availability Zone from ECS Container Metadata,
exiting\" 1>&2; exit 1; fi; ECS_TASK_ARN_REGEX='^arn:(aws|aws-cn|aws-us-gov):ecs:
[a-z0-9-]+:[0-9]{12}:task/[a-zA-Z0-9-]+/[a-zA-Z0-9]+$'; if ! [[ $ECS_TASK_ARN
=~ $ECS_TASK_ARN_REGEX ]]; then echo \"Error extracting Task ARN from ECS
Container Metadata, exiting\" 1>&2; exit 1; fi; CREATE_ACTIVATION_OUTPUT=
$(aws ssm create-activation --iam-role $MANAGED_INSTANCE_ROLE_NAME --
tags Key=ECS_TASK_AVAILABILITY_ZONE,Value=$ECS_TASK_AVAILABILITY_ZONE
Key=ECS_TASK_ARN,Value=$ECS_TASK_ARN Key=FAULT_INJECTION_SIDEAR,Value=true --
region $ECS_TASK_REGION); ACTIVATION_CODE=$(echo $CREATE_ACTIVATION_OUTPUT | jq
-e -r .ActivationCode); ACTIVATION_ID=$(echo $CREATE_ACTIVATION_OUTPUT | jq -e
-r .ActivationId); if ! amazon-ssm-agent -register -code $ACTIVATION_CODE -id
$ACTIVATION_ID -region $ECS_TASK_REGION; then echo \"Failed to register with AWS
Systems Manager (SSM), exiting\" 1>&2; exit 1; fi; amazon-ssm-agent & SSM_AGENT_PID=
$!; wait $SSM_AGENT_PID; else echo \"ECS Container Metadata not found, exiting\"

```

```

1>&2; exit 1; fi; else echo \"SSM agent is already running, exiting\" 1>&2; exit 1;
fi"
],
"environment": [
  {
    "name": "MANAGED_INSTANCE_ROLE_NAME",
    "value": "SSMManagedInstanceRole"
  }
],
"environmentFiles": [],
"mountPoints": [],
"volumesFrom": [],
"secrets": [],
"dnsServers": [],
"dnsSearchDomains": [],
"extraHosts": [],
"dockerSecurityOptions": [],
"dockerLabels": {},
"ulimits": [],
"logConfiguration": {},
"systemControls": []
}

```

Eine besser lesbare Version des Skripts finden Sie unter [the section called “Referenzversion des Skripts”](#).

- Aktivieren Sie Amazon ECS Fault Injection APIs, indem Sie das `enableFaultInjection` Feld in der Amazon ECS-Aufgabendefinition festlegen:

```
"enableFaultInjection": true,
```

- Wenn Sie die `aws:ecs:task-network-packet-loss` Aktionen `aws:ecs:task-network-blackhole-port` oder `aws:ecs:task-network-latency` verwenden, muss der `useEcsFaultInjectionEndpoints` Parameter für die Aktion auf `true` gesetzt sein.
- Wenn Sie die `aws:ecs:task-network-packet-loss` Aktionen `aws:ecs:task-kill-process` oder `aws:ecs:task-network-blackhole-port` verwenden, muss die Amazon ECS-Aufgabendefinition auf `pidMode` eingestellt sein.
- Wenn Sie die `aws:ecs:task-network-packet-loss` Aktionen `aws:ecs:task-network-blackhole-port` oder `aws:ecs:task-network-latency` verwenden, muss die [Netzwerkoption in der Aufgabendefinition](#) auf `awsvpc` oder `host` gesetzt sein.

Referenzversion des Skripts

Im Folgenden finden Sie eine besser lesbare Version des Skripts im Abschnitt Anforderungen als Referenz.

```
#!/usr/bin/env bash

# This is the activation script used to register ECS tasks as Managed Instances in SSM
# The script retrieves information from the ECS task metadata endpoint to add three
# tags to the Managed Instance
# - ECS_TASK_AVAILABILITY_ZONE: To allow customers to target Managed Instances / Tasks
#   in a specific Availability Zone
# - ECS_TASK_ARN: To allow customers to target Managed Instances / Tasks by using the
#   Task ARN
# - FAULT_INJECTION_SIDE CAR: To make it clear that the tasks were registered as
#   managed instance for fault injection purposes. Value is always 'true'.
# The script will leave the SSM Agent running in the background
# When the container running this script receives a SIGTERM or SIGINT signal, it will
# do the following cleanup:
# - Delete SSM activation
# - Deregister SSM managed instance

set -e # stop execution instantly as a query exits while having a non-zero

dnf upgrade -y
dnf install jq procps awscli -y

term_handler() {
    echo "Deleting SSM activation $ACTIVATION_ID"
    if ! aws ssm delete-activation --activation-id $ACTIVATION_ID --region
$ECS_TASK_REGION; then
        echo "SSM activation $ACTIVATION_ID failed to be deleted" 1>&2
    fi

    MANAGED_INSTANCE_ID=$(jq -e -r .ManagedInstanceID /var/lib/amazon/ssm/registration)
    echo "Deregistering SSM Managed Instance $MANAGED_INSTANCE_ID"
    if ! aws ssm deregister-managed-instance --instance-id $MANAGED_INSTANCE_ID --region
$ECS_TASK_REGION; then
        echo "SSM Managed Instance $MANAGED_INSTANCE_ID failed to be deregistered" 1>&2
    fi

    kill -SIGTERM $SSM_AGENT_PID
}
```

```

trap term_handler SIGTERM SIGINT

# check if the required IAM role is provided
if [[ -z $MANAGED_INSTANCE_ROLE_NAME ]] ; then
    echo "Environment variable MANAGED_INSTANCE_ROLE_NAME not set, exiting" 1>&2
    exit 1
fi

# check if the agent is already running (it will be if ECS Exec is enabled)
if ! ps ax | grep amazon-ssm-agent | grep -v grep > /dev/null; then

    # check if ECS Container Metadata is available
    if [[ -n $ECS_CONTAINER_METADATA_URI_V4 ]] ; then

        # Retrieve info from ECS task metadata endpoint
        echo "Found ECS Container Metadata, running activation with metadata"
        TASK_METADATA=$(curl "${ECS_CONTAINER_METADATA_URI_V4}/task")
        ECS_TASK_AVAILABILITY_ZONE=$(echo $TASK_METADATA | jq -e -r '.AvailabilityZone')
        ECS_TASK_ARN=$(echo $TASK_METADATA | jq -e -r '.TaskARN')
        ECS_TASK_REGION=$(echo $ECS_TASK_AVAILABILITY_ZONE | sed 's/.$//')

        # validate ECS_TASK_AVAILABILITY_ZONE
        ECS_TASK_AVAILABILITY_ZONE_REGEX='^(af|ap|ca|cn|eu|me|sa|us|us-gov)-(central|north|
(north(east|west))|south|south(east|west)|east|west)-[0-9]{1}[a-z]{1}$'
        if ! [[ $ECS_TASK_AVAILABILITY_ZONE =~ $ECS_TASK_AVAILABILITY_ZONE_REGEX ]] ; then
            echo "Error extracting Availability Zone from ECS Container Metadata, exiting"
            1>&2
            exit 1
        fi

        # validate ECS_TASK_ARN
        ECS_TASK_ARN_REGEX='^arn:(aws|aws-cn|aws-us-gov):ecs:[a-z0-9-]+:[0-9]{12}:task/[a-
zA-Z0-9-_-]+/[a-zA-Z0-9-]+$'
        if ! [[ $ECS_TASK_ARN =~ $ECS_TASK_ARN_REGEX ]] ; then
            echo "Error extracting Task ARN from ECS Container Metadata, exiting" 1>&2
            exit 1
        fi

        # Create activation tagging with Availability Zone and Task ARN
        CREATE_ACTIVATION_OUTPUT=$(aws ssm create-activation \
            --iam-role $MANAGED_INSTANCE_ROLE_NAME \
            --tags Key=ECS_TASK_AVAILABILITY_ZONE,Value=$ECS_TASK_AVAILABILITY_ZONE
Key=ECS_TASK_ARN,Value=$ECS_TASK_ARN Key=FAULT_INJECTION_SIDE CAR,Value=true \
            --region $ECS_TASK_REGION)

```



```

ACTIVATION_CODE=$(echo $CREATE_ACTIVATION_OUTPUT | jq -e -r .ActivationCode)
ACTIVATION_ID=$(echo $CREATE_ACTIVATION_OUTPUT | jq -e -r .ActivationId)

# Register with AWS Systems Manager (SSM)
if ! amazon-ssm-agent -register -code $ACTIVATION_CODE -id $ACTIVATION_ID -region
$ECS_TASK_REGION; then
    echo "Failed to register with AWS Systems Manager (SSM), exiting" 1>&2
    exit 1
fi

# the agent needs to run in the background, otherwise the trapped signal
# won't execute the attached function until this process finishes
amazon-ssm-agent &
SSM_AGENT_PID=$!

# need to keep the script alive, otherwise the container will terminate
wait $SSM_AGENT_PID

else
    echo "ECS Container Metadata not found, exiting" 1>&2
    exit 1
fi

else
    echo "SSM agent is already running, exiting" 1>&2
    exit 1
fi

```

Beispiel für eine Versuchsvorlage

Im Folgenden finden Sie ein Beispiel für eine Versuchsvorlage für die [the section called "aws:ecs:task-cpu-stress"](#) Aktion.

```

{
  "description": "Run CPU stress on the target ECS tasks",
  "targets": {
    "myTasks": {
      "resourceType": "aws:ecs:task",
      "resourceArns": [
        "arn:aws:ecs:us-east-1:111122223333:task/my-  
cluster/09821742c0e24250b187dfed8EXAMPLE"
      ],

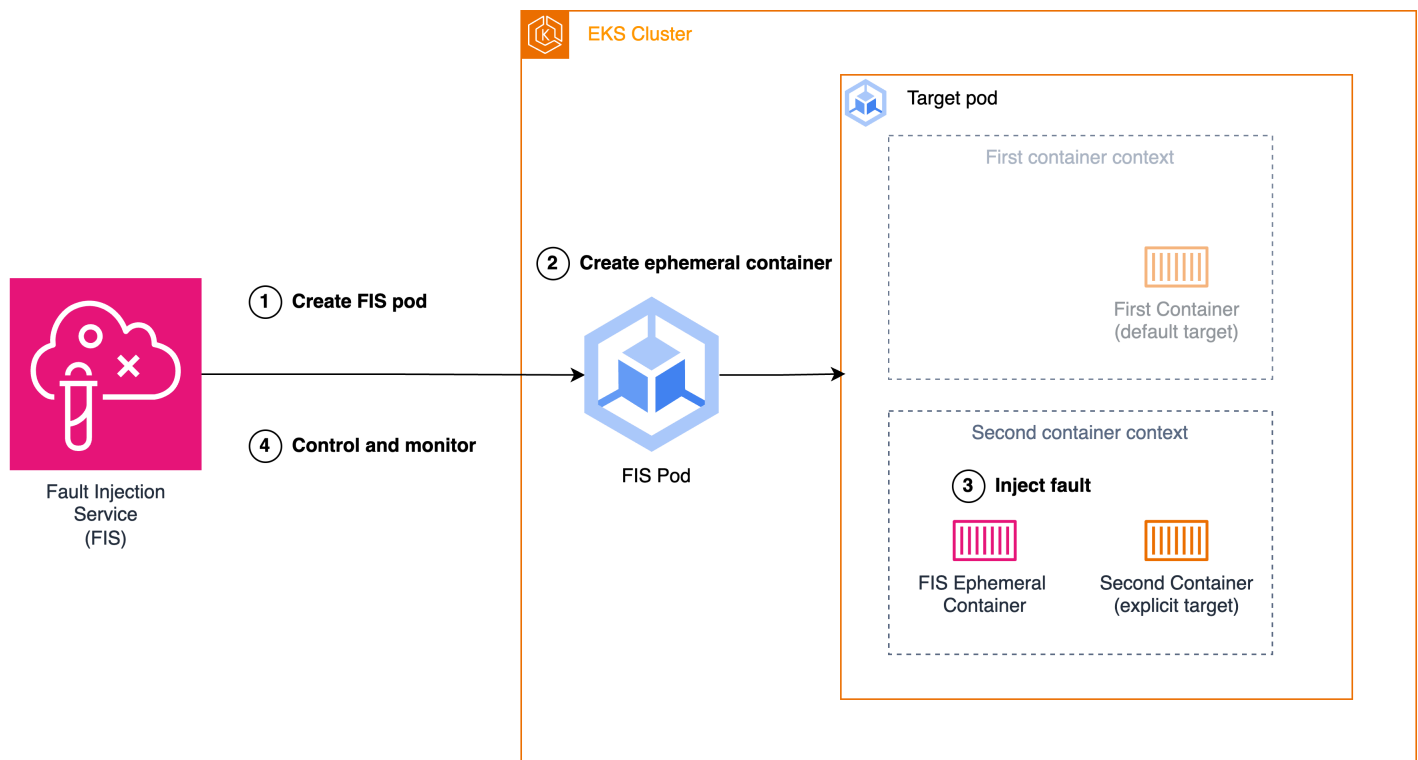
```

```
        "selectionMode": "ALL"
    },
    "actions": {
        "EcsTask-cpu-stress": {
            "actionId": "aws:ecs:task-cpu-stress",
            "parameters": {
                "duration": "PT1M"
            },
            "targets": {
                "Tasks": "myTasks"
            }
        }
    },
    "stopConditions": [
        {
            "source": "none",
        }
    ],
    "roleArn": "arn:aws:iam::111122223333:role/fis-experiment-role",
    "tags": {}
}
```

Verwenden Sie die AWS FIS-Aktionen `aws:eks:pod`

Sie können die `aws:eks:pod`-Aktionen verwenden, um Fehler in die Kubernetes-Pods zu injizieren, die in Ihren EKS-Clustern ausgeführt werden.

Wenn eine Aktion initiiert wird, ruft FIS das FIS-Pod-Container-Image ab. Dieses Image wird dann verwendet, um einen Pod im EKS-Zielcluster zu erstellen. Der neu erstellte Pod ist für die Injektion, Steuerung und Überwachung des Fehlers verantwortlich. Bei allen FIS EKS-Aktionen, mit Ausnahme von [aws:eks:pod-delete](#), wird die Fehlerinjektion durch die Verwendung von [ephemeren Containern erreicht, einer Kubernetes-Funktion, die die Erstellung temporärer Container](#) innerhalb eines vorhandenen Pods ermöglicht. Der ephemere Container wird im selben Namespace wie der Zielcontainer gestartet und führt die gewünschten Fault-Injection-Aufgaben aus. Wenn kein Zielcontainer angegeben ist, wird der erste Container in der Pod-Spezifikation als Ziel ausgewählt.



1. FIS erstellt den FIS-Pod in dem in der Experimentvorlage angegebenen Zielcluster.
2. Der FIS-Pod erstellt einen kurzlebigen Container im Ziel-Pod im selben Namespace wie der Zielcontainer.
3. Der ephemere Container fügt Fehler in den Namespace des Zielcontainers ein.
4. Der FIS-Pod steuert und überwacht die Fehlerinjektion des ephemeren Containers und FIS steuert und überwacht den FIS-Pod.

Nach Abschluss des Experiments oder wenn ein Fehler auftritt, werden der ephemere Behälter und der FIS-Pod entfernt.

Aktionen

- [the section called “aws:eks:pod-cpu-stress”](#)
- [the section called “aws:eks:pod-delete”](#)
- [the section called “aws:eks:pod-io-stress”](#)
- [the section called “aws:eks:pod-memory-stress”](#)
- [the section called “aws:eks:pod-network-blackhole-port”](#)

- [the section called “aws:eks:pod-network-latency”](#)
- [the section called “aws:eks:pod-network-packet-loss”](#)

Einschränkungen

- Die folgenden Aktionen funktionieren nicht mit: AWS Fargate
 - aws:eks:pod-network-blackhole-port
 - aws:eks:pod-network-latency
 - aws:eks:pod-network-packet-loss
- Die folgenden Aktionen unterstützen den bridge [Netzwerkmodus](#) nicht:
 - aws:eks:pod-network-blackhole-port
 - aws:eks:pod-network-latency
 - aws:eks:pod-network-packet-loss
- Für die folgenden Aktionen sind Root-Rechte innerhalb des kurzlebigen Containers erforderlich.
 - aws:eks:pod-network-blackhole-port
 - aws:eks:pod-network-latency
 - aws:eks:pod-network-packet-loss

Der kurzlebige Container erbt seine Berechtigungen aus dem Sicherheitskontext des Ziel-Pods. Wenn Sie die Container im Pod als Nicht-Root-Benutzer ausführen müssen, können Sie separate Sicherheitskontexte für die Container im Ziel-Pod festlegen.

- Sie können Ziele vom Typ aws:eks:pod in Ihrer Experimentvorlage nicht mithilfe von Ressourcen- oder Ressourcen-Tags identifizieren. ARNs Sie müssen Ziele anhand der erforderlichen Ressourcenparameter identifizieren.
- Die Aktionen aws:eks: pod-network-latency und aws:eks: pod-network-packet-loss sollten nicht parallel ausgeführt werden und auf denselben Pod abzielen. Je nach Wert des von Ihnen angegebenen maxErrors Parameters kann die Aktion mit dem Status Abgeschlossen oder Fehlgeschlagen enden:
 - Wenn maxErrorsPercent der Wert 0 ist (Standard), endet die Aktion mit dem Status Fehlgeschlagen.
 - Andernfalls summiert sich der Fehler auf das maxErrorsPercent Budget. Wenn die Anzahl der fehlgeschlagenen Injektionen die angegebene Anzahl nicht erreichtmaxErrors, wird die Aktion als abgeschlossen angezeigt.

- Sie können diese Fehler anhand der Protokolle des injizierten kurzlebigen Containers im Ziel-Pod identifizieren. Es wird fehlschlagen mit. `Exit Code: 16`
- Die Aktion `aws:eks: pod-network-blackhole-port` sollte nicht parallel zu anderen Aktionen ausgeführt werden, die auf denselben Pod abzielen und denselben verwenden. `trafficType` Parallele Aktionen, die unterschiedliche Verkehrsarten verwenden, werden unterstützt.
- FIS kann den Status der Fehlerinjektion nur überwachen, wenn `securityContext` der Ziel-Pods auf `readOnlyRootFilesystem: false` eingestellt ist. Ohne diese Konfiguration schlagen alle EKS-Pod-Aktionen fehl.

Voraussetzungen

- Installieren Sie das AWS CLI auf Ihrem Computer. Dies ist nur erforderlich, wenn Sie die verwenden AWS CLI , um IAM-Rollen zu erstellen. Weitere Informationen finden Sie unter [Installation oder Aktualisierung von](#). AWS CLI
- Installieren Sie kubectl auf Ihrem Computer. Dies ist nur erforderlich, um mit dem EKS-Cluster zu interagieren und die Zielanwendung zu konfigurieren oder zu überwachen. Weitere Informationen finden Sie unter <https://kubernetes.io/docs/tasks/tools/>.
- Die unterstützte Mindestversion von EKS ist 1.23.

Erstellen Sie eine Experimentrolle

Um ein Experiment durchzuführen, müssen Sie eine IAM-Rolle für das Experiment konfigurieren. Weitere Informationen finden Sie unter [the section called “Die Rolle des Experiments”](#). Die erforderlichen Berechtigungen für diese Rolle hängen von der Aktion ab, die Sie verwenden. Die erforderlichen Berechtigungen für Ihre Aktion finden Sie in den [AWS FIS-Aktionen, die darauf abzielen `aws:eks:pod`](#).

Das Kubernetes-Servicekonto konfigurieren

Konfigurieren Sie ein Kubernetes-Dienstkonto, um Experimente mit Zielen im angegebenen Kubernetes-Namespace durchzuführen. Im folgenden Beispiel ist *myserviceaccount* das Dienstkonto und der Namespace. *default* Beachten Sie, dass default dies einer der Standard-Kubernetes-Namespaces ist.

Um Ihr Kubernetes-Dienstkonto zu konfigurieren

1. Erstellen Sie eine Datei mit dem Namen `rbac.yaml` und fügen Sie Folgendes hinzu.

```
kind: ServiceAccount
apiVersion: v1
metadata:
  namespace: default
  name: myserviceaccount

---
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: role-experiments
rules:
- apiGroups: [""]
  resources: ["configmaps"]
  verbs: [ "get", "create", "patch", "delete"]
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["create", "list", "get", "delete", "deletecollection"]
- apiGroups: [""]
  resources: ["pods/ephemeralcontainers"]
  verbs: ["update"]
- apiGroups: [""]
  resources: ["pods/exec"]
  verbs: ["create"]
- apiGroups: ["apps"]
  resources: ["deployments"]
  verbs: ["get"]

---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: bind-role-experiments
  namespace: default
subjects:
- kind: ServiceAccount
  name: myserviceaccount
  namespace: default
```

```
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: fis-experiment
roleRef:
  kind: Role
  name: role-experiments
  apiGroup: rbac.authorization.k8s.io
```

2. Führen Sie den folgenden Befehl aus.

```
kubectl apply -f rbac.yaml
```

Gewähren Sie IAM-Benutzern und -Rollen Zugriff auf Kubernetes APIs

Folgen Sie den Schritten, die in der Dokumentation unter [IAM-Identitäten mit Kubernetes-Berechtigungen verknüpfen](#) beschrieben werden. EKS

Option 1: Zugriffseinträge erstellen

Wir empfehlen die Verwendung von Access Entries. Sie können den folgenden Befehl verwenden, um einen Zugriffseintrag zu erstellen, der die IAM-Rolle dem Kubernetes-Benutzer zuordnet. *fis-experiment* Weitere Informationen finden Sie unter [Gewähren Sie IAM-Benutzern Zugriff auf Kubernetes mit EKS-Zugriffseinträgen](#).

```
aws eks create-access-entry \
    --principal-arn arn:aws:iam::123456789012:role/fis-experiment-role \
    --username fis-experiment \
    --cluster-name my-cluster
```

Important

Um Zugriffseinträge nutzen zu können, muss der Authentifizierungsmodus des EKS-Clusters entweder auf den API_AND_CONFIG_MAP Modus oder konfiguriert werden. API

Option 2: Fügen Sie Einträge zur AWS-Auth hinzu ConfigMap

Sie können auch den folgenden Befehl verwenden, um eine Identitätszuordnung zu erstellen. Weitere Informationen finden Sie in der Dokumentation zu eksctl unter [Manage IAM users and roles](#).

```
eksctl create iamidentitymapping \
    --arn arn:aws:iam::123456789012:role/fis-experiment-role \
    --username fis-experiment \
    --cluster my-cluster
```

Important

Die Nutzung des eksctl-Toolkits zur Konfiguration von Identitätszuordnungen führt zur Erstellung von Einträgen innerhalb von. aws-auth ConfigMap Es ist wichtig zu beachten, dass diese generierten Einträge die Aufnahme einer Pfadkomponente nicht unterstützen. Folglich darf der als Eingabe bereitgestellte ARN kein Pfadsegment enthalten (z. B. `arn:aws:iam::123456789012:role/service-role/fis-experiment-role`).

Pod-Container-Bilder

Die von AWS FIS bereitgestellten Pod-Container-Images werden in Amazon ECR gehostet. Wenn Sie auf ein Bild von Amazon ECR verweisen, müssen Sie die vollständige Bild-URI verwenden.

Das Pod-Container-Image ist auch in der [AWS ECR Public Gallery](#) verfügbar.

AWS-Region	Image-URI
US East (Ohio)	<code>051821878176.dkr.ecr.us-east-2.amazonaws.com/aws-fis-pod:0.1</code>
USA Ost (Nord-Virginia)	<code>731367659002.dkr.ecr.us-east-1.amazonaws.com/aws-fis-pod:0.1</code>
USA West (Nordkalifornien)	<code>080694859247.dkr.ecr.us-west-1.amazonaws.com/aws-fis-pod:0.1</code>
USA West (Oregon)	<code>864386544765.dkr.ecr.us-west-2.amazonaws.com/aws-fis-pod:0.1</code>
Africa (Cape Town)	<code>056821267933.dkr.ecr.af-south-1.amazonaws.com/aws-fis-pod:0.1</code>

AWS-Region	Image-URI
Asien-Pazifik (Hongkong)	246405402639.dkr.ecr.ap-east-1.amazonaws.com/aws-fis-pod:0.1
Asien-Pazifik (Mumbai)	524781661239.dkr.ecr.ap-south-1.amazonaws.com/aws-fis-pod:0.1
Asien-Pazifik (Osaka)	148336246925.dkr.ecr.ap-northeast-3.amazonaws.com/aws-fis-pod:0.1
Asien-Pazifik (Seoul)	526524659354.dkr.ecr.ap-northeast-2.amazonaws.com/aws-fis-pod:0.1
Asien-Pazifik (Singapur)	316401638346.dkr.ecr.ap-southeast-1.amazonaws.com/aws-fis-pod:0.1
Asien-Pazifik (Sydney)	488104106298.dkr.ecr.ap-southeast-2.amazonaws.com/aws-fis-pod:0.1
Asien-Pazifik (Tokio)	635234321696.dkr.ecr.ap-northeast-1.amazonaws.com/aws-fis-pod:0.1
Canada (Central)	490658072207.dkr.ecr.ca-central-1.amazonaws.com/aws-fis-pod:0.1
Europe (Frankfurt)	713827034473.dkr.ecr.eu-central-1.amazonaws.com/aws-fis-pod:0.1
Europa (Irland)	205866052826.dkr.ecr.eu-west-1.amazonaws.com/aws-fis-pod:0.1
Europa (London)	327424803546.dkr.ecr.eu-west-2.amazonaws.com/aws-fis-pod:0.1
Europa (Milan)	478809367036.dkr.ecr.eu-south-1.amazonaws.com/aws-fis-pod:0.1
Europa (Paris)	154605889247.dkr.ecr.eu-west-3.amazonaws.com/aws-fis-pod:0.1

AWS-Region	Image-URI
Europa (Spain)	395402409451.dkr.ecr.eu-south-2.amazonaws.com/ aws-fis-pod:0.1
Europa (Stockholm)	263175118295.dkr.ecr.eu-north-1.amazonaws.com/ aws-fis-pod:0.1
Europa (Zürich)	604225987275.dkr.ecr.eu-central-2.amazonaws.com/ aws-fis-pod:0.1
Middle East (Bahrain)	065825543785.dkr.ecr.me-south-1.amazonaws.com/ aws-fis-pod:0.1
Naher Osten (VAE)	438374459301.dkr.ecr.me-central-1.amazonaws.com/ aws-fis-pod:0.1
Südamerika (São Paulo)	767113787785.dkr.ecr.sa-east-1.amazonaws.com/aws- fis-pod:0.1
AWS GovCloud (US-Ost)	246533647532.dkr.ecr.us-gov-east-1.amazonaws.com/ aws-fis-pod:0.1
AWS GovCloud (US-West)	246529956514.dkr.ecr.us-gov-west-1.amazonaws.com/ aws-fis-pod:0.1

Beispiel für eine Versuchsvorlage

Im Folgenden finden Sie ein Beispiel für eine Versuchsvorlage für die [the section called "aws:eks:pod-network-latency"](#) Aktion.

```
{
  "description": "Add latency and jitter to the network interface for the target EKS Pods",
  "targets": {
    "myPods": {
      "resourceType": "aws:eks:pod",
      "parameters": {
        "clusterIdentifier": "mycluster",
```

```

        "namespace": "default",
        "selectorType": "labelSelector",
        "selectorValue": "mylabel=mytarget"
    },
    "selectionMode": "COUNT(3)"
}
},
"actions": {
    "EksPod-latency": {
        "actionId": "aws:eks:pod-network-latency",
        "description": "Add latency",
        "parameters": {
            "kubernetesServiceAccount": "myserviceaccount",
            "duration": "PT5M",
            "delayMilliseconds": "200",
            "jitterMilliseconds": "10",
            "sources": "0.0.0.0/0"
        },
        "targets": {
            "Pods": "myPods"
        }
    }
},
"stopConditions": [
    {
        "source": "none",
    }
],
"roleArn": "arn:aws:iam::111122223333:role/fis-experiment-role",
"tags": {
    "Name": "EksPodNetworkLatency"
}
}

```

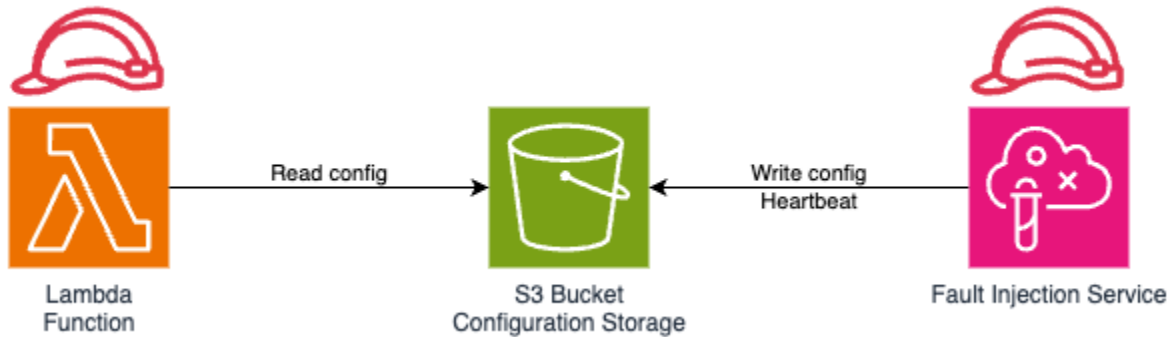
Verwenden Sie die Aktionen AWS FIS aws:lambda:function

Sie können die Aktionen aws:lambda:function verwenden, um Fehler in Aufrufe Ihrer Funktionen einzufügen. AWS Lambda

Diese Aktionen verwenden eine verwaltete Erweiterung, um Fehler einzufügen. AWS FIS Um aws:lambda:function actions zu verwenden, müssen Sie die Erweiterung als Ebene an Ihre Lambda-

Funktionen anhängen und einen Amazon S3 S3-Bucket für die Kommunikation zwischen und der Erweiterung konfigurieren. AWS FIS

Wenn Sie ein AWS FIS Experiment ausführen, das auf `aws:lambda:function` abzielt, AWS FIS liest die Amazon S3 S3-Konfiguration aus Ihrer Lambda-Funktion und schreibt Informationen zur Fehlerinjektion an den angegebenen Amazon S3 S3-Speicherort, wie in der folgenden Abbildung dargestellt.



Aktionen

- [the section called “aws:lambda:invocation-add-delay”](#)
- [the section called “aws:lambda:invocation-error”](#)
- [the section called “aws:lambda:invocation-http-integration-response”](#)

Einschränkungen

- Die AWS FIS Lambda-Erweiterung kann nicht mit Funktionen verwendet werden, die Antwort-Streaming verwenden. Selbst wenn keine Fehler auftreten, unterdrückt die AWS FIS Lambda-Erweiterung Streaming-Konfigurationen. Weitere Informationen finden Sie im AWS Lambda Benutzerhandbuch unter [Antwortstreaming für Lambda-Funktionen](#).

Voraussetzungen

Stellen Sie vor der Verwendung von AWS FIS Lambda-Aktionen sicher, dass Sie diese einmaligen Aufgaben abgeschlossen haben:

- Erstellen Sie einen Amazon S3 S3-Bucket in der Region, von der aus Sie ein Experiment starten möchten. Sie können einen einzelnen Amazon S3 S3-Bucket für mehrere Experimente verwenden

und den Bucket von mehreren AWS Konten gemeinsam nutzen. Sie müssen jedoch für jedes Bucket einen eigenen Bucket haben AWS-Region.

- Erstellen Sie eine IAM-Richtlinie, um der Lambda-Erweiterung Lesezugriff auf den Amazon S3 S3-Bucket zu gewähren. `my-config-distribution-bucket` Ersetzen Sie ihn in der folgenden Vorlage durch den Namen des Amazon S3 S3-Buckets, den Sie oben erstellt haben, und `FisConfigs` durch den Namen eines Ordners in Ihrem Amazon S3 S3-Bucket, den Sie verwenden möchten.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowListingConfigLocation",
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::my-config-distribution-bucket"],
      "Condition": {
        "StringLike": {
          "s3:prefix": ["FisConfigs/*"]
        }
      }
    },
    {
      "Sid": "AllowReadingObjectFromConfigLocation",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": ["arn:aws:s3:::my-config-distribution-bucket/FisConfigs/
*"]
    }
  ]
}
```

- Erstellen Sie eine IAM-Richtlinie, um Schreibzugriff für das AWS FIS Experiment auf den Amazon S3 S3-Bucket zu gewähren. `my-config-distribution-bucket` Ersetzen Sie ihn in der folgenden Vorlage durch den Namen des Amazon S3 S3-Buckets, den Sie oben erstellt haben, und `FisConfigs` durch den Namen eines Ordners in Ihrem Amazon S3 S3-Bucket, den Sie verwenden möchten.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowFisToWriteAndDeleteFaultConfigurations",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::my-config-distribution-bucket/FisConfigs/*"
    },
    {
      "Sid": "AllowFisToInspectLambdaFunctions",
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowFisToDoTagLookups",
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": "*"
    }
  ]
}
```

Konfigurieren von Lambda-Funktionen

Gehen Sie für jede Lambda-Funktion, auf die Sie Einfluss nehmen möchten, wie folgt vor:

1. Hängen Sie die oben erstellte Amazon S3 S3-Lesezugriffsrichtlinie an die Lambda-Funktion an.
2. Hängen Sie die AWS FIS Erweiterung als Ebene an die Funktion an. Weitere Informationen zur Ebene finden Sie ARNs unter [Verfügbare Versionen der AWS FIS Erweiterung für Lambda](#).

3. Setzen Sie die `AWS_FIS_CONFIGURATION_LOCATION` Variable beispielsweise auf den ARN des Amazon S3 S3-Konfigurationsordners `arn:aws:s3:::my-config-distribution-bucket/FisConfigs/`.
4. Setzen Sie die Variable `AWS_LAMBDA_EXEC_WRAPPER` auf `/opt/aws-fis/bootstrap`.

Konfigurieren Sie ein AWS FIS Experiment

Bevor Sie Ihr Experiment ausführen, stellen Sie sicher, dass Sie die Amazon S3 S3-Schreibzugriffsrichtlinie, die Sie in den Voraussetzungen erstellt haben, an die Experimentrollen angehängt haben, die AWS FIS Lambda-Aktionen verwenden werden. Weitere Informationen zur Einrichtung eines AWS FIS Experiments finden Sie unter [Verwaltung von AWS FIS-Experimentvorlagen](#).

Protokollierung

Die AWS FIS Lambda-Erweiterung schreibt Protokolle in die Konsole und CloudWatch protokolliert. Die Protokollierung kann mithilfe der `AWS_FIS_LOG_LEVEL` Variablen konfiguriert werden. Unterstützte Werte sind `INFO`, `WARN` und `ERROR`. Protokolle werden in dem für Ihre Lambda-Funktion konfigurierten Protokollformat geschrieben.

Das Folgende ist ein Beispiel für ein Protokoll im Textformat:

```
2024-08-09T18:51:38.599984Z INFO AWS FIS EXTENSION - extension enabled 1.0.1
```

Im Folgenden finden Sie ein Beispiel für ein Protokoll im JSON-Format:

```
{
  "timestamp": "2024-10-08T17:15:36.953905Z",
  "level": "INFO",
  "fields": {
    "message": "AWS FIS EXTENSION - adding 5000 milliseconds of latency to function invocation",
    "requestId": "0608bf70-908f-4a17-bbfe-3782cd783d8b"
  }
}
```

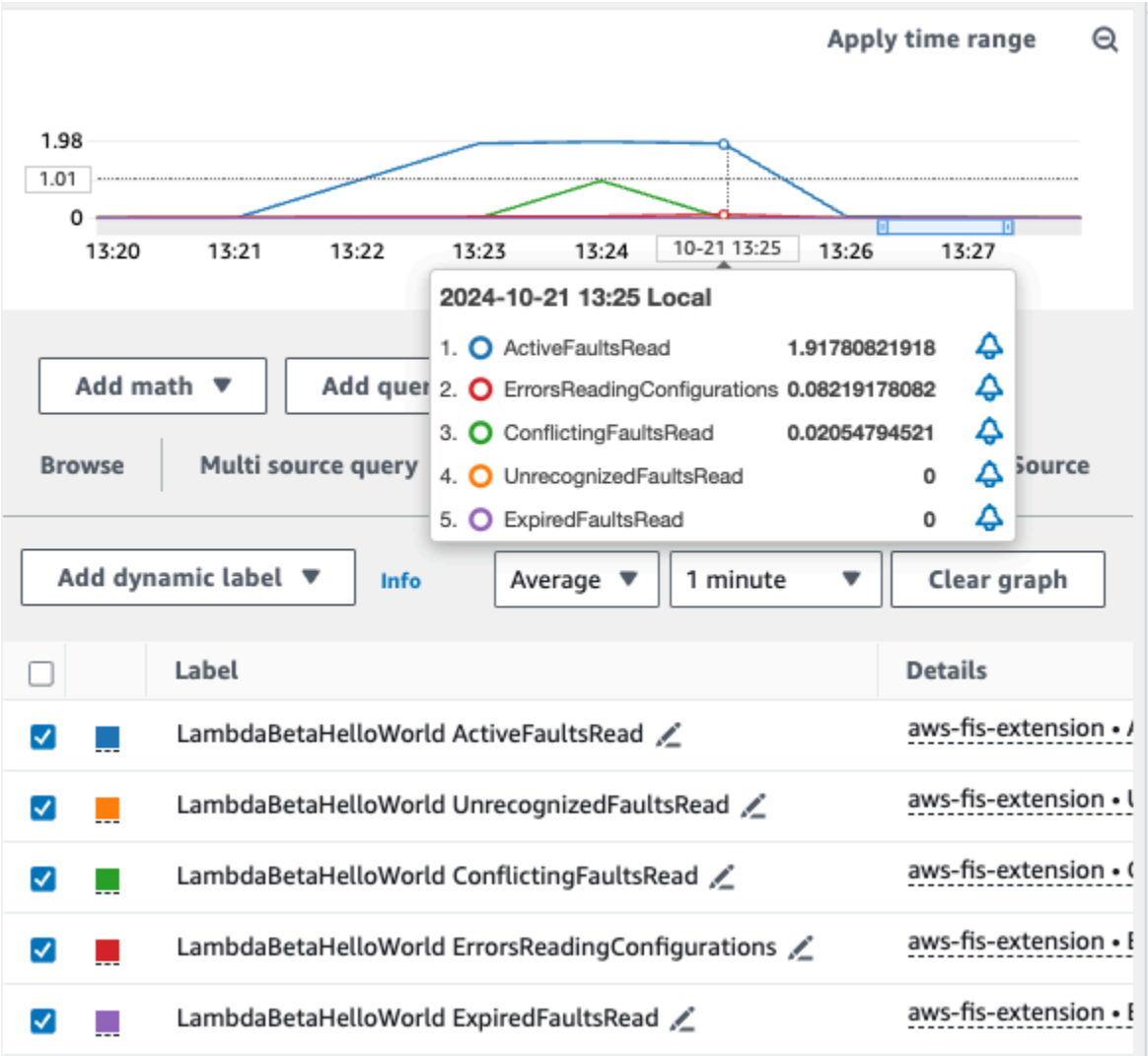
Die ausgegebenen Protokolle können mit CloudWatch Amazon-Metrikfiltern verwendet werden, um benutzerdefinierte Metriken zu generieren. Weitere Informationen zu Metrikfiltern finden Sie

unter [Metriken aus Protokollereignissen mithilfe von Filtern erstellen](#) im Amazon CloudWatch Logs-Benutzerhandbuch.

Verwenden des CloudWatch eingebetteten metrischen Formats (EMF)

Sie können die AWS FIS Lambda-Erweiterung so konfigurieren, dass sie EMF-Protokolle ausgibt, indem Sie die `AWS_FIS_EXTENSION_METRICS` Variable auf `set` setzen. Standardmäßig gibt die Erweiterung keine EMF-Protokolle aus und `AWS_FIS_EXTENSION_METRICS` ist standardmäßig auf `none`. EMF-Protokolle werden in der `aws-fis-extension` namespace auf der Konsole veröffentlicht. CloudWatch

Innerhalb des `aws-fis-extension` Namespace können Sie bestimmte Metriken auswählen, die in einem Diagramm angezeigt werden sollen. Das folgende Beispiel zeigt einige der verfügbaren Metriken im `aws-fis-extension` Namespace.



Fortschrittliche Themen

Dieser Abschnitt enthält zusätzliche Informationen zur AWS FIS Funktionsweise der Lambda-Erweiterung und zu speziellen Anwendungsfällen.

Themen

- [Grundlegendes zu Umfragen](#)
- [Parallelität verstehen](#)
- [Den Prozentsatz der Aufrufe verstehen](#)
- [Besondere Überlegungen für SnapStart](#)
- [Besondere Überlegungen für schnelle, seltene Funktionen](#)
- [Konfiguration mehrerer Erweiterungen mit dem Lambda Runtime API-Proxy](#)
- [Verwendung AWS FIS mit Container-Laufzeiten](#)
- [AWS FIS Lambda-Umgebungsvariablen](#)

Grundlegendes zu Umfragen

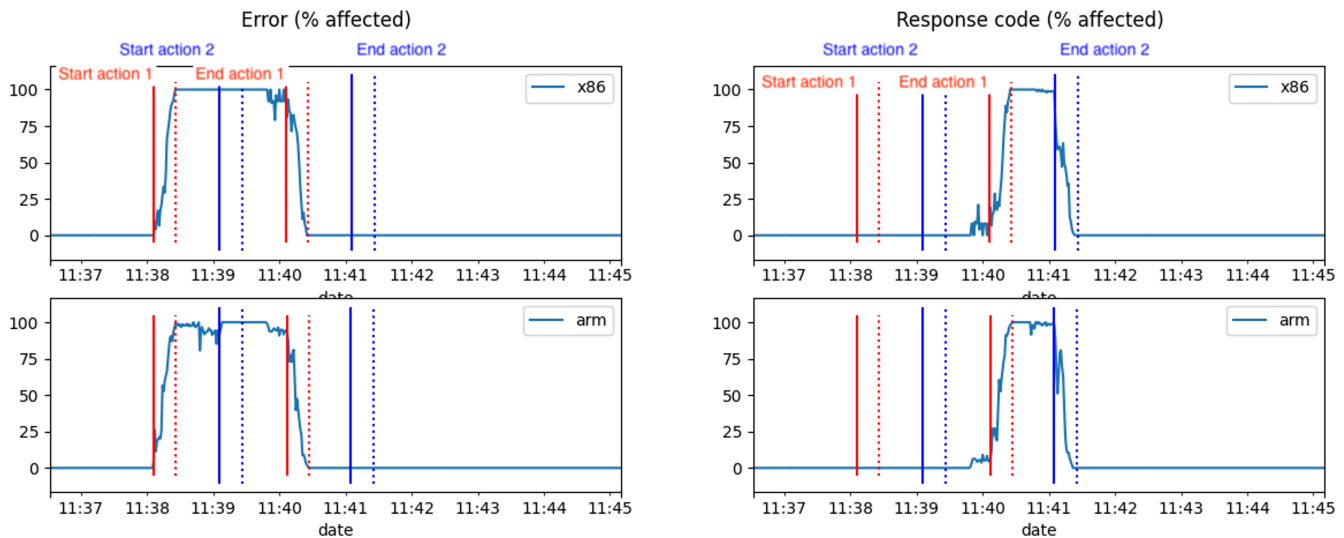
Möglicherweise stellen Sie eine Anlaufzeit von bis zu 60 Sekunden fest, bevor sich Fehler auf alle Aufrufe auswirken. Dies liegt daran, dass die Lambda-Erweiterung selten nach Konfigurationsinformationen fragt, während sie auf den Start eines Experiments wartet. Sie können das Abfrageintervall anpassen, indem Sie die `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS` Umgebungsvariable festlegen (Standard 60s). Bei einem niedrigeren Wert werden häufiger Abfragen durchgeführt, dies hat jedoch größere Auswirkungen auf die Leistung und die Kosten. Möglicherweise stellen Sie auch fest, dass der Fehler bis zu 20 Sekunden heruntergefahren wird. Das liegt daran, dass die Erweiterung während der Ausführung von Experimenten häufiger Abfragen durchführt.

Parallelität verstehen

Sie können auf dieselben Lambda-Funktionen mit mehreren Aktionen gleichzeitig abzielen. Wenn sich die Aktionen alle voneinander unterscheiden, werden alle Aktionen angewendet. Sie können beispielsweise eine anfängliche Verzögerung hinzufügen, bevor ein Fehler zurückgegeben wird. Wenn zwei identische oder widersprüchliche Aktionen auf dieselbe Funktion angewendet werden, wird nur die Aktion mit dem frühesten Startdatum angewendet.

Die folgende Abbildung zeigt, dass sich zwei widersprüchliche Aktionen, `aws:lambda:invocation-error` und `aws:lambda: invocation-http-integration-response` Anfänglich überschneiden. `invocation-http-integration-response` wird anfänglich um 11:38 Uhr hochgefahren und läuft 2 Minuten lang. Dann `invocation-error`

http-integration-response versucht aws:lambda:, um 11:39 Uhr zu starten, wird aber erst um 11:40 Uhr wirksam, nachdem die erste Aktion abgeschlossen ist. Um das Timing des Experiments beizubehalten, wird aws:lambda: invocation-http-integration-response immer noch zur ursprünglich vorgesehenen Zeit von 11:41 Uhr beendet.



Den Prozentsatz der Aufrufe verstehen

Die AWS Fault Injection Service Lambda-Aktionen verwenden ein `aws:lambda:function` -Ziel, mit dem Sie eine oder mehrere Funktionen auswählen können. AWS Lambda ARNs Mithilfe dieser ARNs können die AWS Fault Injection Service Lambda-Aktionen bei jedem Aufruf der ausgewählten Lambda-Funktion Fehler auslösen. Damit Sie Fehler nur bei einem Bruchteil der Aufrufe einfügen können, können Sie für jede Aktion einen `invocationPercentage` Parameter mit Werten von 0 bis 100 angeben. Mithilfe des `invocationPercentage` Parameters können Sie sicherstellen, dass Aktionen auch bei Aufruf-Prozentsätzen unter 100% gleichzeitig ausgeführt werden.

Besondere Überlegungen für SnapStart

AWS Lambda Bei SnapStart aktivierten Funktionen ist die Wahrscheinlichkeit höher, dass sie die gesamte Zeit warten, `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS` bis sie die erste Fehlerkonfiguration aufnehmen, selbst wenn bereits ein Experiment läuft. Dies liegt daran, dass Lambda einen einzelnen Snapshot als Ausgangszustand für mehrere Ausführungsumgebungen SnapStart verwendet und temporären Speicher beibehält. Für die AWS Fault Injection Service Lambda-Erweiterung wird die Abfragefrequenz beibehalten und die anfängliche Konfigurationsprüfung bei der Initialisierung der Ausführungsumgebung übersprungen. Weitere Informationen zu Lambda SnapStart finden Sie SnapStart im AWS Lambda Benutzerhandbuch unter [Verbesserung der Startleistung mit Lambda](#).

Besondere Überlegungen für schnelle, seltene Funktionen

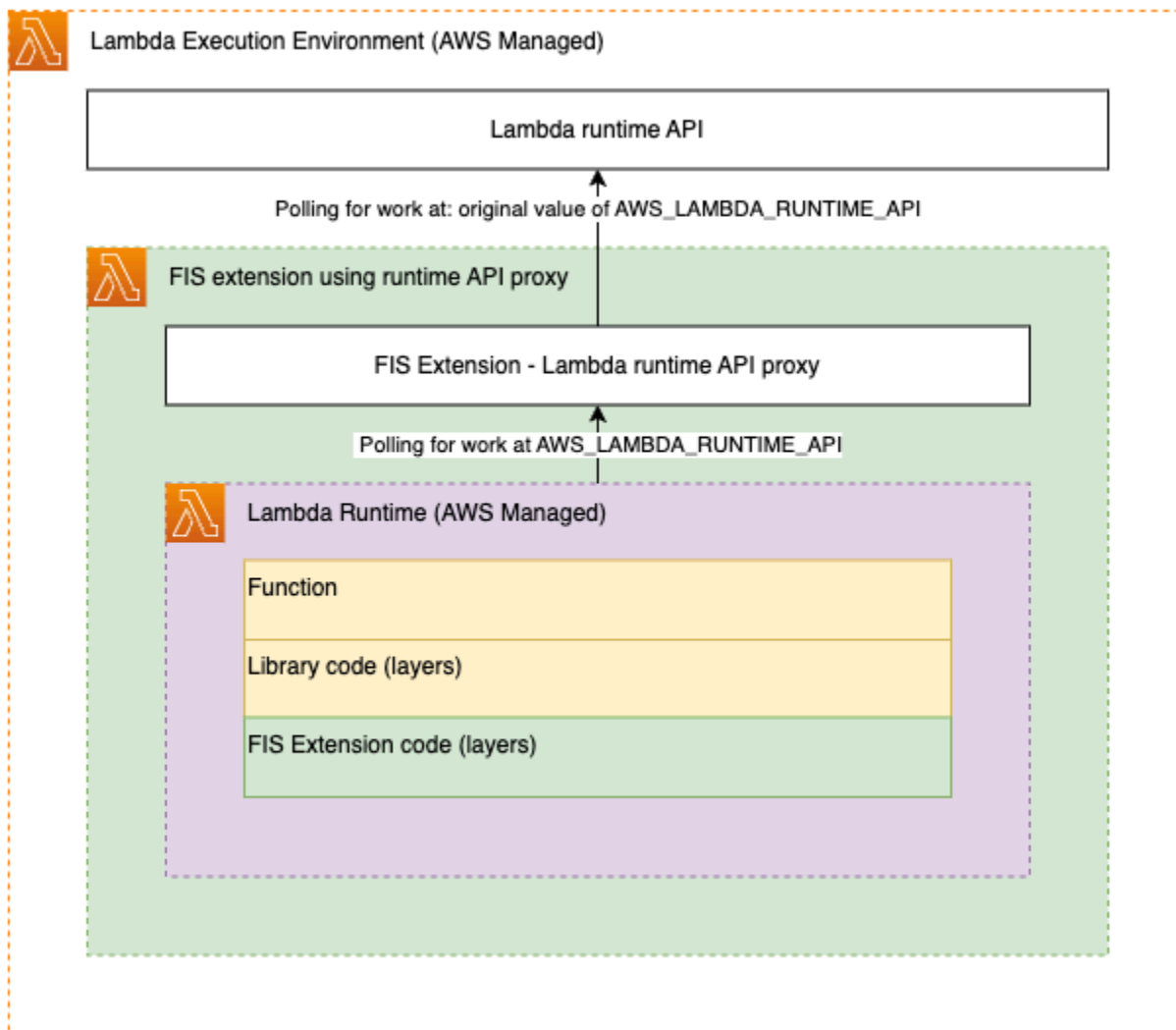
Wenn Ihre Lambda-Funktion weniger als die durchschnittliche Abfragedauer von 70 Millisekunden läuft, benötigt der Abfrage-Thread möglicherweise mehrere Aufrufe, um Fehlerkonfigurationen abzurufen. Wenn die Funktion selten ausgeführt wird, z. B. einmal alle 15 Minuten, wird die Abfrage nie abgeschlossen. Um sicherzustellen, dass der Abfragethread abgeschlossen werden kann, legen Sie den `AWS_FIS_POLL_MAX_WAIT_MILLISECONDS` Parameter fest. Die Erweiterung wartet bis zu der Dauer, die Sie für das Ende einer Umfrage während des Fluges festgelegt haben, bevor sie die Funktion startet. Beachten Sie, dass dadurch die in Rechnung gestellte Funktionsdauer verlängert wird und es bei einigen Aufrufen zu einer zusätzlichen Verzögerung kommt.

Konfiguration mehrerer Erweiterungen mit dem Lambda Runtime API-Proxy

Die Lambda-Erweiterung verwendet den AWS Lambda Runtime-API-Proxy, um Funktionsaufrufen abzufangen, bevor sie die Laufzeit erreichen. Dazu wird der Runtime ein Proxy für die AWS Lambda Runtime-API zur Verfügung gestellt und dessen Position in der Variablen bekannt gegeben.

`AWS_LAMBDA_RUNTIME_API`

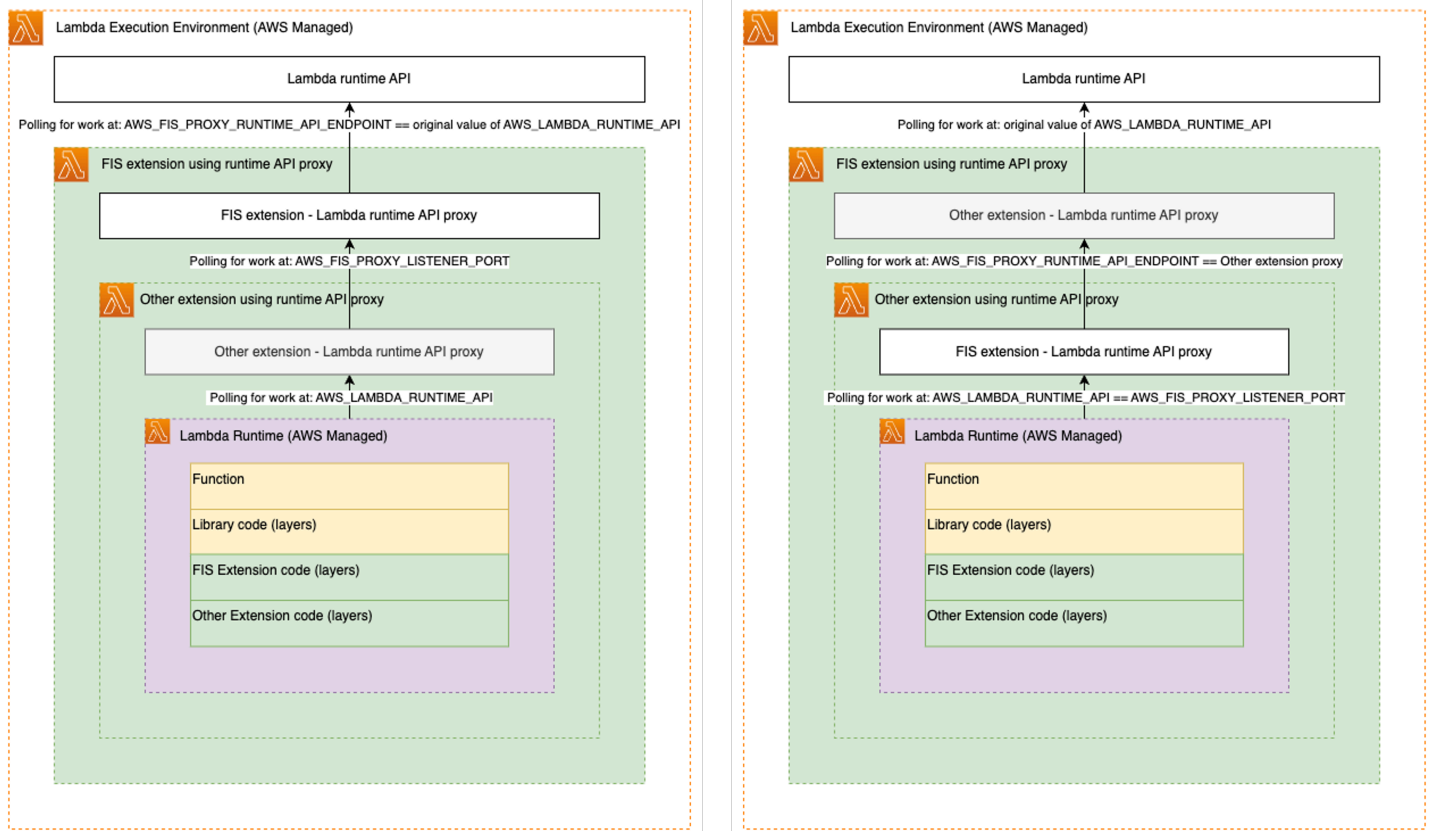
Das folgende Diagramm zeigt die Konfiguration für eine einzelne Erweiterung, die den Lambda Runtime API-Proxy verwendet:



Um die AWS FIS Lambda-Erweiterung mit einer anderen Erweiterung zu verwenden, die das AWS Lambda Runtime-API-Proxymuster verwendet, müssen Sie die Proxys mithilfe eines benutzerdefinierten Bootstrap-Skripts verketteten. Die AWS FIS Lambda-Erweiterung akzeptiert die folgenden Umgebungsvariablen:

- **AWS_FIS_PROXY_RUNTIME_API_ENDPOINT**- Akzeptiert eine Zeichenfolge in der Form, die die lokale IP und den Listener-Port für die AWS Lambda Runtime-API `127.0.0.1:9876` darstellt. Dies kann der ursprüngliche Wert `AWS_LAMBDA_RUNTIME_API` oder der Standort eines anderen Proxys sein.
- **AWS_FIS_PROXY_LISTENER_PORT**- Akzeptiert standardmäßig eine Portnummer, auf der die AWS FIS Erweiterung ihren eigenen Proxy starten soll `9100`.

Mit diesen Einstellungen können Sie die AWS FIS Erweiterung mithilfe des Lambda Runtime API-Proxys in zwei verschiedenen Reihenfolgen mit einer anderen Erweiterung verketteten.



Weitere Informationen zum AWS Lambda Runtime-API-Proxy finden Sie unter [Verbesserung der Runtime-Sicherheit und -Governance mit der AWS Lambda Runtime-API-Proxyerweiterung](#) und [Verwenden der Lambda-Laufzeit-API für benutzerdefinierte Laufzeiten](#) im AWS Lambda Benutzerhandbuch.

Verwendung AWS FIS mit Container-Laufzeiten

Für AWS Lambda Funktionen, die Container-Images verwenden, die die `AWS_LAMBDA_RUNTIME_API` Umgebungsvariable akzeptieren, können Sie die AWS FIS Lambda-Erweiterung in Ihr Container-Image packen, indem Sie die folgenden Schritte ausführen:

1. Ermitteln Sie den ARN der Ebene, aus der die Erweiterung extrahiert werden soll. Weitere Informationen zum Auffinden des ARN finden Sie unter [Konfigurieren von Lambda-Funktionen](#).
2. Verwenden Sie die AWS Command Line Interface (CLI), um Details zur Erweiterung anzufordern: `aws lambda get-layer-version-by-arn --arn fis-extension-arn`. Die Antwort enthält ein `Location` Feld mit einer vorsegnierten URL, von der Sie die FIS-Erweiterung als ZIP-Datei herunterladen können.

3. Entpacken Sie den Inhalt der Erweiterung in Ihr /opt Docker-Dateisystem. Im Folgenden finden Sie ein Beispiel für ein Dockerfile, das auf der NodeJS Lambda-Laufzeit basiert:

```
# extension installation #
FROM amazon/aws-lambda-nodejs:12 AS builder
COPY extension.zip extension.zip
RUN yum install -y unzip
RUN mkdir -p /opt
RUN unzip extension.zip -d /opt
RUN rm -f extension.zip
FROM amazon/aws-lambda-nodejs:12
WORKDIR /opt
COPY --from=builder /opt .
# extension installation finished #
# JS example. Modify as required by your runtime
WORKDIR ${LAMBDA_TASK_ROOT}
COPY index.js package.json .
RUN npm install
CMD [ "index.handler" ]
```

Weitere Informationen zu Container-Images finden Sie im AWS Lambda Benutzerhandbuch unter [Erstellen einer Lambda-Funktion mithilfe eines Container-Images](#).

AWS FIS Lambda-Umgebungsvariablen

Im Folgenden finden Sie eine Liste von Umgebungsvariablen für die AWS FIS Lambda-Erweiterung

- **AWS_FIS_CONFIGURATION_LOCATION**- Erforderlich. Ort, an AWS FIS den aktive Fehlerkonfigurationen geschrieben und die Erweiterung Fehlerkonfigurationen liest. Die Speicherorte sollten im Amazon S3 S3-ARN-Format vorliegen, einschließlich eines Buckets und eines Pfads. Beispiel, `arn:aws:s3:::my-fis-config-bucket/FisConfigs/`.
- **AWS_LAMBDA_EXEC_WRAPPER**- Erforderlich. Speicherort des AWS Lambda [Wrapper-Skripts](#), das zur Konfiguration der AWS FIS Lambda-Erweiterung verwendet wurde. Dies sollte auf das `/opt/aws-fis/bootstrap` Skript gesetzt werden, das in der Erweiterung enthalten ist.
- **AWS_FIS_LOG_LEVEL**- Fakultativ. Protokollebene für Nachrichten, die von der AWS FIS Lambda-Erweiterung ausgegeben werden. Unterstützte Werte sind INFO, WARN und ERROR. Wenn nicht festgelegt, wird die AWS FIS Erweiterung standardmäßig auf INFO gesetzt.

- `AWS_FIS_EXTENSION_METRICS`- Fakultativ. Mögliche Werte sind `all` und `none`. Wenn diese Option aktiviert `all` ist, werden EMF-Metriken unter dem `aws-fis-extension` namespace ausgegeben.
- `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS`- Fakultativ. Wenn diese Option gesetzt ist, wird das Abfrageintervall (in Sekunden) außer Kraft gesetzt, solange die Erweiterung keine Fehler einschleust und darauf wartet, dass eine Fehlerkonfiguration zum Konfigurationsspeicherort hinzugefügt wird. Standardeinstellung: `60`.
- `AWS_FIS_PROXY_RUNTIME_API_ENDPOINT`- Fakultativ. Falls gesetzt, wird der Wert von überschrieben `AWS_LAMBDA_RUNTIME_API`, um zu definieren, wo die AWS FIS Erweiterung mit der AWS Lambda Runtime-API interagiert, um den Funktionsaufruf zu steuern. Erwartet `IP:PORT`, zum Beispiel. `127.0.0.1:9000` Weitere Informationen dazu finden Sie im `AWS_LAMBDA_RUNTIME_API` AWS Lambda Benutzerhandbuch [unter Verwenden der Lambda-Laufzeit-API für benutzerdefinierte Laufzeiten](#).
- `AWS_FIS_PROXY_LISTENER_PORT`- Fakultativ. Definiert den Port, auf dem die AWS FIS Lambda-Erweiterung einen AWS Lambda Runtime-API-Proxy verfügbar macht, der von einer anderen Erweiterung oder der Laufzeit verwendet werden kann. Standardeinstellung: `9100`.
- `AWS_FIS_POLL_MAX_WAIT_MILLISECONDS`- Fakultativ. Wenn diese Variable auf einen Wert ungleich Null gesetzt ist, definiert sie, wie viele Millisekunden die Erweiterung wartet, bis eine asynchrone Abfrage während des Fluges abgeschlossen ist, bevor sie Fehlerkonfigurationen auswertet und den Aufruf der Runtime startet. Standardeinstellung: `0`.

Verfügbare Versionen der AWS FIS Erweiterung für Lambda

Dieser Abschnitt enthält Informationen zu den Versionen der AWS FIS Lambda-Erweiterung. Die Erweiterung unterstützt Lambda-Funktionen, die für die Plattformen x86-64 und ARM64 (Graviton2) entwickelt wurden. Ihre Lambda-Funktion muss so konfiguriert sein, dass sie den spezifischen Amazon-Ressourcennamen (ARN) für den AWS-Region Ort verwendet, an dem sie derzeit gehostet wird. Sie können die ARN-Details unten einsehen `AWS-Region` .

Themen

- [AWS FIS Versionshinweise zur Lambda-Erweiterung](#)
- [Zugriffshandbuch für Lambda Extension ARNs](#)
- [Finden Sie die Versionsnummer Ihrer Lambda-Erweiterung](#)

AWS FIS Versionshinweise zur Lambda-Erweiterung

In der folgenden Tabelle werden die Änderungen beschrieben, die an den letzten Versionen der AWS FIS Lambda-Erweiterung vorgenommen wurden.

Version	Startdatum	Hinweise
1.0.0	2024-10-29	Erstversion

Zugriffshandbuch für Lambda Extension ARNs

Sie müssen mindestens einen Parameter in Ihrem AWS-Konto und haben, AWS-Region bevor Sie mit der Konsole nach öffentlichen Parametern suchen können. Informationen zum Ermitteln öffentlicher Parameter finden Sie [unter Öffentliche Parameter im Parameterspeicher](#) ermitteln.

Zugriff auf die Konsole:

1. Öffnen Sie die AWS Systems Manager Konsole unter <https://console.aws.amazon.com/systems-manager/>.
2. Wählen Sie im Navigationsbereich Parameter Store (Parameterspeicher) aus.
3. Wählen Sie die Registerkarte Öffentliche Parameter aus.
4. Wählen Sie das Dropdown-Menü Einen Service auswählen aus. Wählen Sie fis Sie aus den Dropdown-Optionen.
5. (Optional) Filtern Sie die ausgewählten Parameter, indem Sie weitere Informationen in die Suchleiste eingeben. Bei Arm64-Architekturen filtern Sie die Parameter, indem Sie „arm64“ eingeben. Filtern Sie die Parameter für x86_64-Architekturen, indem Sie „x86_64“ eingeben.
6. Wählen Sie den zu verwendenden öffentlichen Parameter aus.
7. Suchen Sie in den Parameterdetails den ARN-Wert. Kopieren Sie den ARN, der bei der Konfiguration von Layer-Erweiterungen für Ihre Lambda-Zielfunktionen verwendet werden soll.

AWS CLI Zugriff:

SSM-Parameternamen

Die folgenden SSM-Parameternamen sind für verschiedene Architekturen verfügbar:

1. arm64: /aws/service/fis/lambda-extension/AWS-FIS-extension-arm64/1.x.x
2. x86_64: /aws/service/fis/lambda-extension/AWS-FIS-extension-x86_64/1.x.x

AWS CLI Befehlsformat

Um die Erweiterung abzurufen ARNs, verwenden Sie das folgende AWS CLI Befehlsformat, wobei ParameterName der Name für die Architektur und Region das Ziel AWS-Region ist:

```
aws ssm get-parameter --name parameterName --region region
```

Beispielverwendung

```
aws ssm get-parameter --name /aws/service/fis/lambda-extension/AWS-FIS-extension-x86_64/1.x.x --region ap-southeast-2
```

Reaktion-Format

Der Befehl gibt ein JSON-Objekt zurück, das die folgenden Parameterdetails enthält. Der ARN der Lambda-Schicht ist im Value-Feld des Parameter-Objekts enthalten. Kopieren Sie den ARN, der bei der Konfiguration von Layer-Erweiterungen für Ihre Lambda-Zielfunktionen verwendet werden soll.

```
{
  "Parameter": {
    "Name": "/aws/service/fis/lambda-extension/AWS-FIS-extension-x86_64/1.x.x",
    "Type": "String",
    "Value": "arn:aws:lambda:ap-southeast-2:211125361907:layer:aws-fis-extension-x86_64:9",
    "Version": 1,
    "LastModifiedDate": "2025-01-02T15:13:54.465000-05:00",
    "ARN": "arn:aws:ssm:ap-southeast-2::parameter/aws/service/fis/lambda-extension/AWS-FIS-extension-x86_64/1.x.x",
    "DataType": "text"
  }
}
```

Programmatischer Zugriff:

Rufen Sie diese öffentlichen Parameter programmgesteuert ab, wenn Sie Ihre Lambda-Funktionen mithilfe von Infrastructure as Code (IaC) erstellen oder konfigurieren. Dieser Ansatz hilft dabei, Ihre Lambda-Funktionen mit der neuesten Layer-Version ARN aufrechtzuerhalten, ohne dass manuelle Codeaktualisierungen erforderlich wären, die erforderlich wären, wenn der ARN der AWS FIS Erweiterungsschicht hartcodiert wäre. Die folgenden Ressourcen zeigen, wie Sie öffentliche Parameter mithilfe gängiger IaC-Plattformen abrufen können:

- [Rufen Sie öffentliche Parameter mit dem SDK ab AWS](#)
- [Rufen Sie mithilfe des AWS CDK öffentliche AWS Systems Manager Parameter aus dem Parameter Store ab](#)
- [Holen Sie sich öffentliche Parameter mit Terraform](#)

Finden Sie die Versionsnummer Ihrer Lambda-Erweiterung

Gehen Sie wie folgt vor, um die Versionsnummer Ihrer aktuell konfigurierten AWS FIS Lambda-Erweiterung zu ermitteln.

1. Öffnen Sie die AWS Lambda Konsole unter <https://console.aws.amazon.com/lambda/>
2. Wählen Sie die Lambda-Funktion aus, der Sie die AWS-FIS-Extension Ebene hinzufügen möchten.
3. Klicken Sie im Abschnitt Ebenen auf Bearbeiten.
4. Wählen Sie im Abschnitt Ebenen bearbeiten die Option Ebene hinzufügen aus.
5. Wählen Sie im Abschnitt Layer auswählen die Option ARN angeben aus.
6. Geben Sie den ARN für die AWS FIS Erweiterungsschicht ein, der Ihrer AWS-Region AND-Architektur entspricht. Sie können den ARN mithilfe der Konsole oder der in den vorherigen Abschnitten beschriebenen programmatischen Zugriffsmethoden finden. AWS CLI
7. Wählen Sie Überprüfen aus, um zu bestätigen, dass der Layer-ARN gültig ist, und wählen Sie dann Hinzufügen aus.
8. Verwenden Sie die Registerkarte Test, um die Funktion zu testen.
9. Sehen Sie sich nach Abschluss des Tests die Protokollausgabe an. Suchen Sie die Version der AWS FIS Lambda-Erweiterung im Abschnitt Details der Ausführung.

Verwaltung von AWS FIS-Experimentvorlagen

Sie können Versuchsvorlagen mit der AWS FIS-Konsole oder der Befehlszeile erstellen und verwalten. Eine Experimentvorlage enthält eine oder mehrere Aktionen, die während eines Experiments auf bestimmten Zielen ausgeführt werden sollen. Sie enthält auch die Stoppbedingungen, die verhindern, dass das Experiment die Grenzwerte überschreitet. Weitere Informationen zu den Komponenten einer Versuchsvorlage finden Sie unter [Komponenten der Versuchsvorlage](#). Nachdem Sie eine Versuchsvorlage erstellt haben, können Sie sie verwenden, um ein Experiment durchzuführen.

Aufgaben

- [Erstellen Sie eine Versuchsvorlage](#)
- [Vorlagen für Experimente anzeigen](#)
- [Generieren Sie eine Zielvorschau aus einer Experimentvorlage](#)
- [Starten Sie ein Experiment mit einer Vorlage](#)
- [Aktualisieren Sie eine Experimentvorlage](#)
- [Versuchs-Vorlagen mit Tags versehen](#)
- [Löschen Sie eine Experimentvorlage](#)
- [Beispiele für AWS FIS-Experimentvorlagen](#)

Erstellen Sie eine Versuchsvorlage

Führen Sie als Erstes die folgenden Schritte aus:

- [Plane dein Experiment](#).
- Erstellen Sie eine IAM-Rolle, die dem AWS FIS-Dienst die Erlaubnis erteilt, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter [IAM-Rollen für AWS FIS-Experimente](#).
- Stellen Sie sicher, dass Sie Zugriff auf FIS haben. AWS Weitere Informationen finden Sie unter Beispiele für [AWS FIS-Richtlinien](#).

Um eine Versuchsvorlage mit der Konsole zu erstellen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.

2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Experimentvorlage erstellen aus.
4. Gehen Sie für Schritt 1, Vorlagendetails angeben, wie folgt vor:
 - a. Geben Sie unter Beschreibung und Name eine Beschreibung für die Vorlage ein, z. Amazon S3 Network Disrupt Connectivity B.
 - b. (Optional) Wählen Sie für das Konten-Targeting die Option Mehrere Konten aus, um eine Versuchsvorlage für mehrere Konten zu konfigurieren.
 - c. Wählen Sie Weiter und fahren Sie mit Schritt 2, Aktionen und Ziele angeben, fort.
5. Geben Sie unter Aktionen den Aktionssatz für die Vorlage an. Wählen Sie für jede Aktion Aktion hinzufügen aus und führen Sie die folgenden Schritte aus:
 - Geben Sie unter Name einen Namen für die Aktion ein.

Zulässige Zeichen sind alphanumerische Zeichen, Bindestriche (-) und Unterstriche (_). Der Name muss mit einem Buchstaben beginnen. Leerzeichen sind nicht zulässig. Jeder Aktionsname muss in dieser Vorlage eindeutig sein.
 - (Optional) Geben Sie unter Beschreibung eine Beschreibung für die Aktion ein. Die maximale Länge beträgt 512 Zeichen.
 - (Optional) Wählen Sie für Start danach eine andere in dieser Vorlage definierte Aktion aus, die abgeschlossen sein muss, bevor die aktuelle Aktion gestartet wird. Andernfalls wird die Aktion zu Beginn des Experiments ausgeführt.
 - Wählen Sie als Aktionstyp die AWS FIS-Aktion aus.
 - Wählen Sie für Ziel ein Ziel aus, das Sie im Abschnitt Ziele definiert haben. Wenn Sie noch kein Ziel für diese Aktion definiert haben, erstellt AWS FIS ein neues Ziel für Sie.
 - Geben Sie unter Aktionsparameter die Parameter für die Aktion an. Dieser Abschnitt wird nur angezeigt, wenn die AWS FIS-Aktion Parameter hat.
 - Wählen Sie Speichern.
6. Definieren Sie für Ziele die Zielressourcen, auf denen die Aktionen ausgeführt werden sollen. Sie müssen mindestens eine Ressourcen-ID oder ein Ressourcen-Tag als Ziel angeben. Wählen Sie Bearbeiten, um das Ziel zu bearbeiten, AWS das FIS im vorherigen Schritt für Sie erstellt hat, oder wählen Sie Ziel hinzufügen. Gehen Sie für jedes Ziel wie folgt vor:
 - Geben Sie unter Name einen Namen für das Ziel ein.

Zulässige Zeichen sind alphanumerische Zeichen, Bindestriche (-) und Unterstriche (_). Der Name muss mit einem Buchstaben beginnen. Leerzeichen sind nicht zulässig. Jeder Zielname muss in dieser Vorlage eindeutig sein.

- Wählen Sie unter Ressourcentyp einen Ressourcentyp aus, der für die Aktion unterstützt wird.
- Führen Sie für die Target-Methode einen der folgenden Schritte aus:
 - Wählen Sie Ressource IDs und wählen Sie dann die Ressource aus oder fügen Sie sie hinzu IDs.
 - Wählen Sie Ressourcen-Tags, Filter und Parameter aus und fügen Sie dann die benötigten Tags und Filter hinzu. Weitere Informationen finden Sie unter [the section called "Identifizieren Sie die Zielressourcen"](#).
 - Wählen Sie für den Auswahlmodus Anzahl, um die Aktion für die angegebene Anzahl identifizierter Ziele auszuführen, oder wählen Sie Prozent, um die Aktion für den angegebenen Prozentsatz der identifizierten Ziele auszuführen. Standardmäßig wird die Aktion auf allen identifizierten Zielen ausgeführt.
 - Wählen Sie Speichern.
- 7. Um eine Aktion mit dem Ziel zu aktualisieren, das Sie erstellt haben, suchen Sie die Aktion unter Aktionen, wählen Sie Bearbeiten aus, und aktualisieren Sie dann Ziel. Sie können dasselbe Ziel für mehrere Aktionen verwenden.
- 8. (Optional) Wählen Sie für die Experimentoptionen das Verhalten des Auflösungsmodus mit leerem Ziel aus.
- 9. Wählen Sie Weiter, um mit Schritt 3, Servicezugriff konfigurieren, fortzufahren.
- 10. Wählen Sie für Service Access die Option Bestehende IAM-Rolle verwenden und wählen Sie dann die IAM-Rolle aus, die Sie wie in den Voraussetzungen für dieses Tutorial beschrieben erstellt haben. Wenn Ihre Rolle nicht angezeigt wird, stellen Sie sicher, dass sie über die erforderliche Vertrauensstellung verfügt. Weitere Informationen finden Sie unter [the section called "Die Rolle des Experiments"](#).
- 11. (Nur Experimente mit mehreren Konten) Fügen Sie für Target-Kontokonfigurationen einen Rollen-ARN und eine optionale Beschreibung für jedes Zielkonto hinzu. Um die Rolle des Zielkontos ARNs mit einer CSV-Datei hochzuladen, wählen Sie Rolle ARNs für alle Zielkonten hochladen und anschließend CSV-Datei auswählen
- 12. Wählen Sie Weiter, um mit Schritt 4, Optionale Einstellungen konfigurieren, fortzufahren.
- 13. (Optional) Wählen Sie unter Stoppbedingungen die CloudWatch Amazon-Alarme für die Stoppbedingungen aus. Weitere Informationen finden Sie unter [Stoppbedingungen für AWS FIS](#).

14. (Optional) Konfigurieren Sie für Logs die Zieloption. Um Logs an einen S3-Bucket zu senden, wählen Sie An einen Amazon S3 S3-Bucket senden und geben Sie den Bucket-Namen und das Präfix ein. Um Logs an Logs zu CloudWatch senden, wählen Sie Send to CloudWatch Logs und geben Sie die Log-Gruppe ein.
15. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus und geben Sie einen Tag-Schlüssel und einen Tag-Wert an. Die von Ihnen hinzugefügten Tags werden auf Ihre Experimentvorlage angewendet, nicht auf die Experimente, die mit der Vorlage ausgeführt werden.
16. Wählen Sie Weiter, um mit Schritt 5, Überprüfen und erstellen, fortzufahren.
17. Prüfen Sie die Vorlage und wählen Sie Experimentvorlage erstellen aus. Wenn Sie zur Bestätigung aufgefordert werden `create`, geben Sie ein und wählen Sie dann Experimentvorlage erstellen.

So erstellen Sie eine Experimentvorlage mit der CLI

Verwenden Sie den Befehl [create-experiment-template](#).

Sie können eine Experimentvorlage aus einer JSON-Datei laden.

Verwenden Sie den Parameter `--cli-input-json`.

```
aws fis create-experiment-template --cli-input-json fileb://<path-to-json-file>
```

Weitere Informationen finden Sie im AWS Command Line Interface Benutzerhandbuch unter [Generieren einer CLI-Skeleton-Vorlage](#). Beispielvorlagen finden Sie unter [Beispiele für AWS FIS-Experimentvorlagen](#).

Vorlagen für Experimente anzeigen

Sie können die von Ihnen erstellten Experimentvorlagen anzeigen.

So zeigen Sie eine Experimentvorlage mit der Konsole an

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Um Informationen zu einer bestimmten Vorlage anzuzeigen, wählen Sie die ID der Experimentvorlage aus.

4. Im Abschnitt Details können Sie die Beschreibung und die Stoppbedingungen für die Vorlage einsehen.
5. Um die Aktionen für die Experimentvorlage anzuzeigen, wählen Sie Aktionen.
6. Um die Ziele für die Experimentvorlage anzuzeigen, wählen Sie Ziele aus.
7. Um die Tags für die Experimentvorlage anzuzeigen, wählen Sie „Tags“.

So zeigen Sie eine Experimentvorlage mit der CLI an

Verwenden Sie den [list-experiment-templates](#)Befehl, um eine Liste von Experimentvorlagen abzurufen, und verwenden Sie den [get-experiment-template](#)Befehl, um Informationen zu einer bestimmten Experimentvorlage abzurufen.

Generieren Sie eine Zielvorschau aus einer Experimentvorlage

Bevor Sie ein Experiment starten, können Sie eine Zielvorschau generieren, um zu überprüfen, ob Ihre Experimentvorlage so konfiguriert ist, dass sie auf die erwarteten Ressourcen abzielt. Die Ressourcen, auf die Sie zu Beginn des eigentlichen Experiments abzielen, können sich von denen in der Vorschau unterscheiden, da Ressourcen entfernt, aktualisiert oder nach dem Zufallsprinzip ausgewählt werden können. Wenn Sie eine Zielvorschau generieren, starten Sie ein Experiment, bei dem alle Aktionen übersprungen werden.

Note

Durch das Generieren einer Zielvorschau können Sie nicht überprüfen, ob Sie über die erforderlichen Berechtigungen verfügen, um Aktionen mit Ihren Ressourcen durchzuführen.

Um eine Zielvorschau mit der Konsole zu starten

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Um die Ziele für die Experimentvorlage anzuzeigen, wählen Sie Ziele aus.
4. Um Ihre Zielressourcen für die Experimentvorlage zu überprüfen, wählen Sie „Vorschau generieren“. Wenn Sie ein Experiment ausführen, wird diese Zielvorschau automatisch mit den Zielen aus dem letzten Experiment aktualisiert.

So starten Sie eine Zielvorschau mit der CLI

- Führen Sie den folgenden Befehl [start-experiment](#) aus. Ersetzen Sie die kursiven Werte durch Ihre eigenen Werte.

```
aws fis start-experiment \  
  --experiment-options actionsMode=skip-all \  
  --experiment-template-id EXTxxxxxxxxx
```

Starten Sie ein Experiment mit einer Vorlage

Nachdem Sie eine Experimentvorlage erstellt haben, können Sie Experimente mit dieser Vorlage starten.

Wenn Sie ein Experiment starten, erstellen wir einen Snapshot der angegebenen Vorlage und verwenden diesen Snapshot, um das Experiment auszuführen. Wenn die Versuchsvorlage während der Ausführung des Experiments aktualisiert oder gelöscht wird, haben diese Änderungen daher keine Auswirkungen auf das laufende Experiment.

Wenn Sie ein Experiment starten, erstellt AWS FIS in Ihrem Namen eine dienstbezogene Rolle. Weitere Informationen finden Sie unter [Verwenden Sie serviceverknüpfte Rollen für AWS den Fault Injection Service](#).

Nachdem Sie das Experiment gestartet haben, können Sie es jederzeit beenden. Weitere Informationen finden Sie unter [Stoppen eines Experiments](#).

Um ein Experiment mit der Konsole zu starten

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie auf Experiment starten.
4. (Optional) Um Ihrem Experiment ein Tag hinzuzufügen, wählen Sie Neues Tag hinzufügen und geben Sie einen Tag-Schlüssel und einen Tag-Wert ein.
5. Wählen Sie Start Experiment (Experiment starten) aus. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den **start** Text ein und wählen Sie Experiment starten.

Um ein Experiment mit der CLI zu starten

Verwenden Sie den Befehl [start-experiment](#).

Aktualisieren Sie eine Experimentvorlage

Sie können eine bestehende Experimentvorlage aktualisieren. Wenn Sie eine Experimentvorlage aktualisieren, wirken sich die Änderungen nicht auf laufende Experimente aus, die die Vorlage verwenden.

Um eine Experimentvorlage mithilfe der Konsole zu aktualisieren

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Experimentvorlage aktualisieren.
4. Ändern Sie die Vorlagendetails nach Bedarf und wählen Sie „Experimentvorlage aktualisieren“.

So aktualisieren Sie eine Experimentvorlage mit der CLI

Verwenden Sie den Befehl [update-experiment-template](#).

Versuchs-Vorlagen mit Tags versehen

Sie können Ihre eigenen Tags auf Versuchsvorlagen anwenden, um sie besser organisieren zu können. Sie können auch [tagbasierte IAM-Richtlinien](#) implementieren, um den Zugriff auf Versuchsvorlagen zu kontrollieren.

Um eine Versuchsvorlage mit der Konsole zu taggen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie auf Aktionen, Tags verwalten.
4. Um ein neues Tag hinzuzufügen, wählen Sie Neues Tag hinzufügen und geben Sie dann einen Schlüssel und einen Wert an.

Um ein Tag zu entfernen, wählen Sie Entfernen für das Tag aus.

5. Wählen Sie Speichern.

Um eine Experimentvorlage mit der CLI zu taggen

Verwenden Sie den Befehl [tag-resource](#).

Löschen Sie eine Experimentvorlage

Wenn Sie eine Experimentvorlage nicht mehr benötigen, können Sie sie löschen. Wenn Sie eine Experimentvorlage löschen, sind alle laufenden Experimente, die die Vorlage verwenden, nicht betroffen. Das Experiment wird so lange ausgeführt, bis es abgeschlossen oder gestoppt wird. Gelöschte Versuchsvorlagen können jedoch nicht auf der Seite Experimente in der Konsole angezeigt werden.

Um eine Experimentvorlage mit der Konsole zu löschen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Experimentvorlage löschen.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie die Option Experimentvorlage löschen ein **delete** und wählen Sie sie aus.

Um eine Experimentvorlage mit der CLI zu löschen

Verwenden Sie den Befehl [delete-experiment-template](#).

Beispiele für AWS FIS-Experimentvorlagen

Wenn Sie die AWS FIS-API oder ein Befehlszeilentool verwenden, um eine Experimentvorlage zu erstellen, können Sie die Vorlage in JavaScript Object Notation (JSON) erstellen. Weitere Informationen zu den Komponenten einer Experimentvorlage finden Sie unter [AWS Komponenten der FIS-Versuchsvorlage](#).

Um ein Experiment mit einer der Beispielvorlagen zu erstellen, speichern Sie es in einer JSON-Datei (z. B. `my-template.json`), ersetzen Sie die Platzhalterwerte in *italics* durch Ihre eigenen Werte und führen Sie dann den folgenden [create-experiment-template](#) Befehl aus.

```
aws fis create-experiment-template --cli-input-json file://my-template.json
```

Beispiel--Vorlagen

- [Stoppen Sie EC2 Instanzen, die auf Filtern basieren](#)
- [Stoppt eine bestimmte Anzahl von Instanzen EC2](#)
- [Führen Sie ein vorkonfiguriertes AWS FIS SSM-Dokument aus](#)
- [Führen Sie ein vordefiniertes Automatisierungs-Runbook aus](#)
- [Drosseln Sie API-Aktionen auf EC2 Instances mit der IAM-Zielrolle](#)
- [Stresstest der CPU von Pods in einem Kubernetes-Cluster](#)
- [Bereitgestellte Durchsatzausnahme für die angegebene Anzahl von Kinesis Data Streams](#)
- [Beispiel für ein Experiment, Rollenberechtigungen](#)

Stoppen Sie EC2 Instanzen, die auf Filtern basieren

Im folgenden Beispiel werden alle laufenden EC2 Amazon-Instances in der angegebenen Region mit dem angegebenen Tag in der angegebenen VPC gestoppt. Sie werden nach zwei Minuten neu gestartet.

```
{
  "tags": {
    "Name": "StopEC2InstancesWithFilters"
  },
  "description": "Stop and restart all instances in us-east-1b with the tag env=prod in the specified VPC",
  "targets": {
    "myInstances": {
      "resourceType": "aws:ec2:instance",
      "resourceTags": {
        "env": "prod"
      },
      "filters": [
        {
          "path": "Placement.AvailabilityZone",
          "values": ["us-east-1b"]
        },
        {
          "path": "State.Name",
          "values": ["running"]
        }
      ]
    }
  }
}
```

```

        "path": "VpcId",
        "values": [ "vpc-aabbcc11223344556" ]
    },
    "selectionMode": "ALL"
},
"actions": {
    "StopInstances": {
        "actionId": "aws:ec2:stop-instances",
        "description": "stop the instances",
        "parameters": {
            "startInstancesAfterDuration": "PT2M"
        },
        "targets": {
            "Instances": "myInstances"
        }
    }
},
"stopConditions": [
    {
        "source": "aws:cloudwatch:alarm",
        "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
    }
],
"roleArn": "arn:aws:iam::111122223333:role/role-name"
}

```

Stoppt eine bestimmte Anzahl von Instanzen EC2

Im folgenden Beispiel werden drei Instanzen mit dem angegebenen Tag gestoppt. AWS FIS wählt die spezifischen Instanzen, die gestoppt werden sollen, nach dem Zufallsprinzip aus. Diese Instanzen werden nach zwei Minuten neu gestartet.

```

{
    "tags": {
        "Name": "StopEC2InstancesByCount"
    },
    "description": "Stop and restart three instances with the specified tag",
    "targets": {
        "myInstances": {
            "resourceType": "aws:ec2:instance",
            "resourceTags": {

```

```

        "env": "prod"
      },
      "selectionMode": "COUNT(3)"
    }
  },
  "actions": {
    "StopInstances": {
      "actionId": "aws:ec2:stop-instances",
      "description": "stop the instances",
      "parameters": {
        "startInstancesAfterDuration": "PT2M"
      },
      "targets": {
        "Instances": "myInstances"
      }
    }
  },
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
    }
  ],
  "roleArn": "arn:aws:iam::111122223333:role/role-name"
}

```

Führen Sie ein vorkonfiguriertes AWS FIS SSM-Dokument aus

[Im folgenden Beispiel wird eine CPU-Fault-Injection für 60 Sekunden auf der angegebenen EC2 Instanz ausgeführt, wobei ein vorkonfiguriertes AWS FIS SSM-Dokument, -CPU-Stress, verwendet wird. AWSFIS-Run](#) AWS FIS überwacht das Experiment zwei Minuten lang.

```

{
  "tags": {
    "Name": "CPUStress"
  },
  "description": "Run a CPU fault injection on the specified instance",
  "targets": {
    "myInstance": {
      "resourceType": "aws:ec2:instance",
      "resourceArns": ["arn:aws:ec2:us-east-1:111122223333:instance/instance-id"],
      "selectionMode": "ALL"
    }
  }
}

```

```

    }
  },
  "actions": {
    "CPUStress": {
      "actionId": "aws:ssm:send-command",
      "description": "run cpu stress using ssm",
      "parameters": {
        "duration": "PT2M",
        "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-CPU-Stress",
        "documentParameters": "{\"DurationSeconds\": \"60\",
\\\"InstallDependencies\\\": \"True\\\", \\\"CPU\\\": \"0\\\"}"
      },
      "targets": {
        "Instances": "myInstance"
      }
    }
  },
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
    }
  ],
  "roleArn": "arn:aws:iam::111122223333:role/role-name"
}

```

Führen Sie ein vordefiniertes Automatisierungs-Runbook aus

[Das folgende Beispiel veröffentlicht eine Benachrichtigung an Amazon SNS mithilfe eines Runbooks, das von Systems Manager, AWS-Publish, bereitgestellt wird. SNSNotification](#) Die Rolle muss über Berechtigungen zum Veröffentlichen von Benachrichtigungen zum angegebenen SNS-Thema verfügen.

```

{
  "description": "Publish event through SNS",
  "stopConditions": [
    {
      "source": "none"
    }
  ],
  "targets": {
  },
  "actions": {

```

```

    "sendToSns": {
      "actionId": "aws:ssm:start-automation-execution",
      "description": "Publish message to SNS",
      "parameters": {
        "documentArn": "arn:aws:ssm:us-east-1::document/AWS-
PublishSNSNotification",
        "documentParameters": "{\"Message\": \"Hello, world\", \"TopicArn\":
\\\"arn:aws:sns:us-east-1:111122223333:topic-name\\\"}\",
        "maxDuration": "PT1M"
      },
      "targets": {
    }
  },
  "roleArn": "arn:aws:iam::111122223333:role/role-name"
}

```

Drosseln Sie API-Aktionen auf EC2 Instances mit der IAM-Zielrolle

Im folgenden Beispiel werden 100% der API-Aufrufe gedrosselt, die in der Aktionsdefinition für API-Aufrufe angegeben sind, die von den in der Zieldefinition angegebenen IAM-Rolle (n) getätigt wurden.

Note

Wenn Sie EC2 Instances als Ziel verwenden möchten, die Mitglieder einer Auto Scaling Scaling-Gruppe sind, verwenden Sie bitte die Aktion `aws:ec2: asg-insufficient-instance-capacity -error` und wählen Sie stattdessen nach Auto Scaling Scaling-Gruppe. Weitere Informationen finden Sie unter [aws:ec2:asg-insufficient-instance-capacity-error](#).

```

{
  "tags": {
    "Name": "ThrottleEC2APIActions"
  },
  "description": "Throttle the specified EC2 API actions on the specified IAM role",
  "targets": {
    "myRole": {
      "resourceType": "aws:iam:role",
      "resourceArns": ["arn:aws:iam::111122223333:role/role-name"],
      "selectionMode": "ALL"
    }
  }
}

```

```

},
"actions": {
  "ThrottleAPI": {
    "actionId": "aws:fis:inject-api-throttle-error",
    "description": "Throttle APIs for 5 minutes",
    "parameters": {
      "service": "ec2",
      "operations": "DescribeInstances,DescribeVolumes",
      "percentage": "100",
      "duration": "PT2M"
    },
    "targets": {
      "Roles": "myRole"
    }
  }
},
"stopConditions": [
  {
    "source": "aws:cloudwatch:alarm",
    "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
  }
],
"roleArn": "arn:aws:iam::111122223333:role/role-name"
}

```

Stresstest der CPU von Pods in einem Kubernetes-Cluster

Im folgenden Beispiel wird Chaos Mesh verwendet, um die CPU von Pods in einem Amazon EKS Kubernetes-Cluster für eine Minute einem Stresstest zu unterziehen.

```

{
  "description": "ChaosMesh StressChaos example",
  "targets": {
    "Cluster-Target-1": {
      "resourceType": "aws:eks:cluster",
      "resourceArns": [
        "arn:aws:eks:arn:aws::111122223333:cluster/cluster-id"
      ],
      "selectionMode": "ALL"
    }
  },
  "actions": {
    "TestCPUSTress": {

```



```

    "actionId": "aws:eks:inject-kubernetes-custom-resource",
    "parameters": {
      "maxDuration": "PT2M",
      "kubernetesApiVersion": "chaos-mesh.org/v1alpha1",
      "kubernetesKind": "StressChaos",
      "kubernetesNamespace": "default",
      "kubernetesSpec": "{\"selector\":{\"namespaces\":[\"default\"],\n\"labelSelectors\":{\"run\":{\"nginx\"}},\"mode\":{\"all\"},\"stressors\":{\"cpu\":{\"workers\":1,\"load\":50}},\"duration\":{\"1m\"}}",
    },
    "targets": {
      "Cluster": "Cluster-Target-1"
    }
  },
  "stopConditions": [{
    "source": "none"
  }],
  "roleArn": "arn:aws:iam::<111122223333>:role/role-name",
  "tags": {}
}

```

Im folgenden Beispiel wird Litmus verwendet, um die CPU von Pods in einem Amazon EKS Kubernetes-Cluster eine Minute lang einem Stresstest zu unterziehen.

```

{
  "description": "Litmus CPU Hog",
  "targets": {
    "MyCluster": {
      "resourceType": "aws:eks:cluster",
      "resourceArns": [
        "arn:aws:eks:arn:aws::<111122223333>:cluster/cluster-id"
      ],
      "selectionMode": "ALL"
    }
  },
  "actions": {
    "MyAction": {
      "actionId": "aws:eks:inject-kubernetes-custom-resource",
      "parameters": {
        "maxDuration": "PT2M",
        "kubernetesApiVersion": "litmuschaos.io/v1alpha1",
        "kubernetesKind": "ChaosEngine",

```

```

        "kubernetesNamespace": "litmus",
        "kubernetesSpec": "{\\"engineState\\":\\"active\\",\\"appinfo\\":
        {\\"appns\\":\\"default\\",\\"applabel\\":\\"run=nginx\\",\\"appkind\\":\\"deployment\\"},
        \\"chaosServiceAccount\\":\\"litmus-admin\\",\\"experiments\\":[{\\"name\\":\\"pod-cpu-hog
        \",\\"spec\\":{\\"components\\":{\\"env\\":[{\\"name\\":\\"TOTAL_CHAOS_DURATION\\",\\"value\\":
        \\"60\\"},{\\"name\\":\\"CPU_CORES\\",\\"value\\":\\"1\\"},{\\"name\\":\\"PODS_AFFECTED_PERC\\",
        \\"value\\":\\"100\\"},{\\"name\\":\\"CONTAINER_RUNTIME\\",\\"value\\":\\"docker\\"},{\\"name\\":
        \\"SOCKET_PATH\\",\\"value\\":\\"/var/run/docker.sock\\"}]}},\\"probe\\":[]}]},\\"annotationCheck
        \":\\"false\\"}"
    },
    "targets": {
        "Cluster": "MyCluster"
    }
}
},
"stopConditions": [{
    "source": "none"
}],
"roleArn": "arn:aws:iam::111122223333:role/role-name",
"tags": {}
}

```

Bereitgestellte Durchsatzausnahme für die angegebene Anzahl von Kinesis Data Streams

Im folgenden Beispiel wird für 100% der Anfragen bis zu fünf Kinesis Data Streams mit dem angegebenen Tag eine Ausnahme für den bereitgestellten Durchsatz ausgelöst. AWS FIS wählt die Streams, die beeinflusst werden sollen, nach dem Zufallsprinzip aus. Nach 5 Minuten ist der Fehler behoben.

```

{
    "description": "Kinesis stream experiment",
    "targets": {
        "KinesisStreams-Target-1": {
            "resourceType": "aws:kinesis:stream",
            "resourceTags": {
                "tag-key": "tag-value"
            },
            "selectionMode": "COUNT(5)"
        }
    },
    "actions": {

```

```

    "kinesis": {
      "actionId": "aws:kinesis:stream-provisioned-throughput-exception",
      "description": "my-stream",
      "parameters": {
        "duration": "PT5M",
        "percentage": "100",
        "service": "kinesis"
      },
      "targets": {
        "KinesisStreams": "KinesisStreams-Target-1"
      }
    },
    "stopConditions": [
      {
        "source": "none"
      }
    ],
    "roleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {},
    "experimentOptions": {
      "accountTargeting": "single-account",
      "emptyTargetResolutionMode": "fail"
    }
  }
}

```

Beispiel für ein Experiment, Rollenberechtigungen

Mit der folgenden Berechtigung können Sie die `aws:kinesis:stream-expired-iterator-exception` Aktionen `aws:kinesis:stream-provisioned-throughput-exception` und für einen bestimmten Stream ausführen, was sich auf 50% der Anfragen auswirkt.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:InjectApiError",
      "Resource": "*"
      "Condition": {
        "ForAllValues:StringEquals": {
          "kinesis:FisActionId": [
            "aws:kinesis:stream-provisioned-throughput-exception",

```

```
        "aws:kinesis:stream-expired-iterator-exception"
      ],
      "kinesis:FisTargetArns": [
        "arn:aws:kinesis:us-east-1:111122223333:stream/stream-name"
      ],
    },
    "NumericEquals": {
      "kinesis:FisInjectPercentage": "50"
    }
  },
  {
    "Action": [
      "kinesis:DescribeStreamSummary",
    ],
    "Resource": "*",
    "Effect": "Allow"
  }
]
```

Verwaltung Ihrer AWS FIS-Experimente

AWS Mit FIS können Sie Experimente zur Fehlerinjektion an Ihren AWS Workloads durchführen. Erstellen Sie zunächst eine [Experimentvorlage](#). Nachdem Sie eine Experimentvorlage erstellt haben, können Sie sie verwenden, um ein Experiment zu starten.

Ein Experiment ist abgeschlossen, wenn einer der folgenden Fälle eintritt:

- Alle [Aktionen](#) in der Vorlage wurden erfolgreich abgeschlossen.
- Eine [Stoppbedingung](#) wird ausgelöst.
- Eine Aktion kann aufgrund eines Fehlers nicht abgeschlossen werden. Zum Beispiel, wenn das [Ziel](#) nicht gefunden werden kann.
- Das Experiment wird [manuell gestoppt](#).

Sie können ein gestopptes oder fehlgeschlagenes Experiment nicht fortsetzen. Sie können ein abgeschlossenes Experiment auch nicht erneut ausführen. Sie können jedoch mit derselben Experimentvorlage ein neues Experiment starten. Sie können die Experimentvorlage optional aktualisieren, bevor Sie sie in einem neuen Experiment erneut angeben.

Aufgaben

- [Starten Sie ein Experiment](#)
- [Sehen Sie sich Ihre Experimente an](#)
- [Kennzeichnen Sie ein Experiment](#)
- [Stoppen eines Experiments](#)
- [Aufgelöste Ziele auflisten](#)

Starten Sie ein Experiment

Sie starten ein Experiment anhand einer Experimentvorlage. Weitere Informationen finden Sie unter [Starten Sie ein Experiment mit einer Vorlage](#).

Sie können Ihre Experimente als einmalige Aufgabe oder als wiederkehrende Aufgaben planen mit Amazon EventBridge. Weitere Informationen finden Sie unter [Tutorial: Ein wiederkehrendes Experiment planen](#).

Sie können Ihr Experiment mit einer der folgenden Funktionen überwachen:

- Sehen Sie sich Ihre Experimente in der AWS FIS-Konsole an. Weitere Informationen finden Sie unter [Sehen Sie sich Ihre Experimente an](#).
- Sehen Sie sich CloudWatch Amazon-Metriken für die Zielressourcen in Ihren Experimenten an oder sehen Sie sich AWS FIS-Nutzungsmetriken an. Weitere Informationen finden Sie unter [Überwachen Sie mit CloudWatch](#).
- Aktivieren Sie die Protokollierung von Experimenten, um während der Ausführung detaillierte Informationen über Ihr Experiment zu erfassen. Weitere Informationen finden Sie unter [Protokollierung von Experimenten](#).

Sehen Sie sich Ihre Experimente an

Sie können den Fortschritt eines laufenden Experiments sowie Experimente anzeigen, die abgeschlossen, beendet oder fehlgeschlagen sind.

Abgebrochene, abgeschlossene und fehlgeschlagene Experimente werden nach 120 Tagen automatisch aus Ihrem Konto entfernt.

Um Experimente über die Konsole anzusehen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimente aus.
3. Wählen Sie die Experiment-ID des Experiments aus, um die zugehörige Detailseite zu öffnen.
4. Führen Sie eine oder mehrere der folgenden Aktionen aus:
 - Unter Details, Status finden Sie den [Status des Experiments](#).
 - Wählen Sie die Registerkarte Aktionen, um Informationen zu den Aktionen des Experiments zu erhalten.
 - Wählen Sie die Registerkarte Ziele, um Informationen zu den Versuchszielen zu erhalten.
 - Wählen Sie die Registerkarte Zeitleiste für eine visuelle Darstellung der Aktionen auf der Grundlage ihrer Start- und Endzeiten.

So zeigen Sie Experimente mit der CLI an

Verwenden Sie den Befehl [list-experiments](#), um eine Liste von Experimenten abzurufen, und verwenden Sie den Befehl [get-experiment](#), um Informationen zu einem bestimmten Experiment abzurufen.

Status des Experiments

Ein Experiment kann sich in einem der folgenden Zustände befinden:

- ausstehend — Das Experiment steht noch aus.
- einleiten — Das Experiment bereitet sich auf den Start vor.
- läuft — Das Experiment läuft.
- abgeschlossen — Alle Aktionen des Experiments wurden erfolgreich abgeschlossen.
- Beenden — Die Stopp-Bedingung wurde ausgelöst oder das Experiment wurde manuell gestoppt.
- gestoppt — Alle laufenden oder ausstehenden Aktionen im Experiment werden gestoppt.
- fehlgeschlagen — Das Experiment ist aufgrund eines Fehlers fehlgeschlagen, z. B. aufgrund unzureichender Berechtigungen oder falscher Syntax.
- abgebrochen — Das Experiment wurde gestoppt oder konnte nicht gestartet werden, weil der Sicherheitshebel betätigt wurde.

Status der Aktion

Eine Aktion kann sich in einem der folgenden Zustände befinden:

- ausstehend — Die Aktion steht noch aus, entweder weil das Experiment noch nicht gestartet wurde oder weil die Aktion zu einem späteren Zeitpunkt im Experiment beginnen soll.
- einleiten — Die Aktion bereitet sich auf den Start vor.
- läuft — Die Aktion läuft.
- abgeschlossen — Die Aktion wurde erfolgreich abgeschlossen.
- abgebrochen — Das Experiment wurde beendet, bevor die Aktion gestartet wurde.
- übersprungen — Die Aktion wurde übersprungen.
- Beenden — Die Aktion wird beendet.
- gestoppt — Alle laufenden oder ausstehenden Aktionen im Experiment werden gestoppt.
- fehlgeschlagen — Die Aktion ist aufgrund eines Client-Fehlers fehlgeschlagen, z. B. aufgrund unzureichender Berechtigungen oder falscher Syntax.

Kennzeichnen Sie ein Experiment

Sie können Experimente mit Tags versehen, um sie besser organisieren zu können. Sie können auch [tagbasierte IAM-Richtlinien](#) implementieren, um den Zugriff auf Experimente zu kontrollieren.

Um ein Experiment mit der Konsole zu taggen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimente aus.
3. Wählen Sie das Experiment aus und wählen Sie Aktionen, Tags verwalten aus.
4. Um ein neues Tag hinzuzufügen, wählen Sie Neues Tag hinzufügen und geben Sie einen Schlüssel und einen Wert an.

Um ein Tag zu entfernen, wählen Sie Entfernen für das Tag aus.

5. Wählen Sie Speichern.

Um ein Experiment mit der CLI zu taggen

Verwenden Sie den Befehl [tag-resource](#).

Stoppen eines Experiments

Sie können ein laufendes Experiment jederzeit beenden. Wenn Sie ein Experiment beenden, werden alle nachträglichen Aktionen, die für eine Aktion noch nicht abgeschlossen wurden, abgeschlossen, bevor das Experiment beendet wird. Sie können ein gestopptes Experiment nicht fortsetzen.

Um ein Experiment mit der Konsole zu beenden

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimente aus.
3. Wählen Sie das Experiment aus und klicken Sie dann auf Experiment beenden.
4. Wählen Sie im Bestätigungsdialogfeld die Option Experiment beenden aus.

Um ein Experiment mit der CLI zu beenden

Verwenden Sie den Befehl [stop-experiment](#).

Aufgelöste Ziele auflisten

Sie können Informationen zu gelösten Zielen für ein Experiment anzeigen, nachdem die Zielauflösung beendet ist.

So zeigen Sie gelöste Ziele mit der Konsole an

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimente aus.
3. Wählen Sie das Experiment und anschließend Bericht aus.
4. Informationen zu den gelösten Zielen finden Sie unter Ressourcen.

So zeigen Sie gelöste Ziele mit der CLI an

Verwenden Sie den Befehl [list-experiment-resolved-targets](#).

Anleitungen für den AWS Fault Injection Service

In den folgenden Tutorials erfahren Sie, wie Sie Experimente mit dem AWS Fault Injection Service (FIS) erstellen und ausführen.

Lernprogramme

- [Tutorial: Stoppen und starten der Testinstanz mit AWS FIS](#)
- [Tutorial: CPU-Belastung auf einer Instance mithilfe von AWS FIS ausführen](#)
- [Tutorial: Testen Sie Spot-Instance-Unterbrechungen mit AWS FIS](#)
- [Tutorial: Simulieren Sie ein Konnektivitätsereignis](#)
- [Tutorial: Ein wiederkehrendes Experiment planen](#)

Tutorial: Stoppen und starten der Testinstanz mit AWS FIS

Sie können den AWS Fault Injection Service (FIS) verwenden, um zu testen, wie Ihre Anwendungen das Stoppen und Starten von Instanzen handhaben. Verwenden Sie dieses Tutorial, um eine Versuchsvorlage zu erstellen, die die `aws:ec2:stop-instances` FIS-Aktion verwendet, um eine Instanz und dann eine zweite Instanz zu stoppen.

Voraussetzungen

Um dieses Tutorial abzuschließen, stellen Sie sicher, dass Sie wie folgt vorgehen:

- Starten Sie zwei EC2 Testinstanzen in Ihrem Konto. Nachdem Sie Ihre Instances gestartet haben, notieren Sie sich IDs der beiden Instances.
- Erstellen Sie eine IAM-Rolle, die es dem AWS FIS-Dienst ermöglicht, die `aws:ec2:stop-instances` Aktion in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter [IAM-Rollen für AWS FIS-Experimente](#).
- Stellen Sie sicher, dass Sie Zugriff auf FIS haben. AWS Weitere Informationen finden Sie unter Beispiele für [AWS FIS-Richtlinien](#).

Schritt 1: Erstellen Sie eine Versuchsvorlage

Erstellen Sie die Experimentvorlage mit der AWS FIS-Konsole. In der Vorlage geben Sie zwei Aktionen an, die nacheinander für jeweils drei Minuten ausgeführt werden. Die erste Aktion stoppt

eine der Testinstanzen, die AWS FIS nach dem Zufallsprinzip auswählt. Die zweite Aktion stoppt beide Testinstanzen.

Um eine Versuchsvorlage zu erstellen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Experimentvorlage erstellen aus.
4. Gehen Sie für Schritt 1, Vorlagendetails angeben, wie folgt vor:
 - a. Geben Sie unter Beschreibung und Name eine Beschreibung für die Vorlage ein, z. Amazon S3 Network Disrupt Connectivity B.
 - b. Wählen Sie Weiter und fahren Sie mit Schritt 2, Aktionen und Ziele angeben, fort.
5. Nehmen Sie bei Aktionen die folgenden Einstellungen vor:
 - a. Wählen Sie Aktion hinzufügen aus.
 - b. Geben Sie einen Namen für die Aktion ein. Geben Sie z. B. **stopOneInstance**.
 - c. Wählen Sie als Aktionstyp `aws:ec2:stop-instances` aus.
 - d. Behalten Sie für Target das Ziel bei, das FIS für Sie erstellt. AWS
 - e. Geben Sie für Aktionsparameter für Instanzen nach Dauer starten den Wert 3 Minuten (PT3M) an.
 - f. Wählen Sie Save (Speichern) aus.
6. Führen Sie für Targets (Ziele) Folgendes aus:
 - a. Wählen Sie Bearbeiten für das Ziel, das AWS FIS im vorherigen Schritt automatisch für Sie erstellt hat.
 - b. Ersetzen Sie den Standardnamen durch einen aussagekräftigeren Namen. Geben Sie z. B. **oneRandomInstance**.
 - c. Stellen Sie sicher, dass der Ressourcentyp `aws:ec2:instance` lautet.
 - d. Wählen Sie für Target-Methode die Option Resource IDs und dann die der beiden IDs Testinstanzen aus.
 - e. Wählen Sie für den Auswahlmodus die Option Count aus. Geben Sie für Anzahl der Ressourcen den Wert ein**1**.
 - f. Wählen Sie Save (Speichern) aus.
7. Wählen Sie Ziel hinzufügen und gehen Sie wie folgt vor:

- a. Geben Sie einen Namen für das Ziel ein. Geben Sie z. B. ei **bothInstances**.
 - b. Wählen Sie als Ressourcentyp `aws:ec2:instance` aus.
 - c. Wählen Sie für Target-Methode die Option Resource IDs und dann die der beiden IDs Testinstanzen aus.
 - d. Wählen Sie für den Auswahlmodus die Option Alle aus.
 - e. Wählen Sie Save (Speichern) aus.
8. Wählen Sie im Abschnitt Aktionen die Option Aktion hinzufügen aus. Gehen Sie wie folgt vor:
- a. Geben Sie unter Name einen Namen für die Aktion ein. Geben Sie z. B. ei **stopBothInstances**.
 - b. Wählen Sie als Aktionstyp `aws:ec2:stop-instances` aus.
 - c. Wählen Sie für Start danach die erste Aktion aus, die Sie hinzugefügt haben ().
stopOneInstance
 - d. Wählen Sie für Ziel das zweite Ziel aus, das Sie hinzugefügt haben (**bothInstances**).
 - e. Geben Sie für Aktionsparameter für Instanzen nach Dauer starten den Wert 3 Minuten (PT3M) an.
 - f. Wählen Sie Save (Speichern) aus.
9. Wählen Sie Weiter, um mit Schritt 3, Dienstzugriff konfigurieren, fortzufahren.
10. Wählen Sie für Service Access die Option Bestehende IAM-Rolle verwenden und wählen Sie dann die IAM-Rolle aus, die Sie wie in den Voraussetzungen für dieses Tutorial beschrieben erstellt haben. Wenn Ihre Rolle nicht angezeigt wird, stellen Sie sicher, dass sie über die erforderliche Vertrauensstellung verfügt. Weitere Informationen finden Sie unter [the section called "Die Rolle des Experiments"](#).
11. Wählen Sie Weiter, um mit Schritt 4, Optionale Einstellungen konfigurieren, fortzufahren.
12. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus und geben Sie einen Tag-Schlüssel und einen Tag-Wert an. Die von Ihnen hinzugefügten Tags werden auf Ihre Experimentvorlage angewendet, nicht auf die Experimente, die mit der Vorlage ausgeführt werden.
13. Wählen Sie Weiter, um mit Schritt 5, Überprüfen und erstellen, fortzufahren.
14. Prüfen Sie die Vorlage und wählen Sie Experimentvorlage erstellen. Wenn Sie zur Bestätigung aufgefordert werden `create`, geben Sie ein und wählen Sie dann Experimentvorlage erstellen.

(Optional) Um die JSON-Version der Experimentvorlage anzuzeigen

Wählen Sie die Registerkarte Exportieren. Im Folgenden finden Sie ein Beispiel für das JSON, das mit der vorherigen Konsolenprozedur erstellt wurde.

```
{
  "description": "Test instance stop and start",
  "targets": {
    "bothInstances": {
      "resourceType": "aws:ec2:instance",
      "resourceArns": [
        "arn:aws:ec2:region:123456789012:instance/instance_id_1",
        "arn:aws:ec2:region:123456789012:instance/instance_id_2"
      ],
      "selectionMode": "ALL"
    },
    "oneRandomInstance": {
      "resourceType": "aws:ec2:instance",
      "resourceArns": [
        "arn:aws:ec2:region:123456789012:instance/instance_id_1",
        "arn:aws:ec2:region:123456789012:instance/instance_id_2"
      ],
      "selectionMode": "COUNT(1)"
    }
  },
  "actions": {
    "stopBothInstances": {
      "actionId": "aws:ec2:stop-instances",
      "parameters": {
        "startInstancesAfterDuration": "PT3M"
      },
      "targets": {
        "Instances": "bothInstances"
      },
      "startAfter": [
        "stopOneInstance"
      ]
    },
    "stopOneInstance": {
      "actionId": "aws:ec2:stop-instances",
      "parameters": {
        "startInstancesAfterDuration": "PT3M"
      },
      "targets": {
        "Instances": "oneRandomInstance"
      }
    }
  }
}
```

```
    }  
  }  
},  
"stopConditions": [  
  {  
    "source": "none"  
  }  
],  
"roleArn": "arn:aws:iam::123456789012:role/AllowFISEC2Actions",  
"tags": {}  
}
```

Schritt 2: Starten Sie das Experiment

Wenn Sie mit der Erstellung Ihrer Experimentvorlage fertig sind, können Sie sie verwenden, um ein Experiment zu starten.

Um ein Experiment zu starten

1. Sie sollten sich auf der Detailseite der Experimentvorlage befinden, die Sie gerade erstellt haben. Andernfalls wählen Sie Experimentvorlagen und dann die ID der Experimentvorlage aus, um die Detailseite zu öffnen.
2. Wählen Sie Start Experiment (Experiment starten) aus.
3. (Optional) Um Ihrem Experiment ein Tag hinzuzufügen, wählen Sie Neues Tag hinzufügen und geben Sie einen Tag-Schlüssel und einen Tag-Wert ein.
4. Wählen Sie Start Experiment (Experiment starten) aus. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den **start** Text ein und wählen Sie Experiment starten.

Schritt 3: Verfolgen Sie den Fortschritt des Experiments

Sie können den Fortschritt eines laufenden Experiments verfolgen, bis das Experiment abgeschlossen, gestoppt oder fehlgeschlagen ist.

Um den Fortschritt eines Experiments zu verfolgen

1. Sie sollten sich auf der Detailseite für das Experiment befinden, das Sie gerade gestartet haben. Andernfalls wählen Sie Experimente und dann die ID des Experiments aus, um die Detailseite zu öffnen.

2. Um den Status des Experiments einzusehen, aktivieren Sie im Detailbereich die Option Status. Weitere Informationen finden Sie unter [Versuchsstatus](#).
3. Wenn der Status des Experiments „Wird ausgeführt“ lautet, fahren Sie mit dem nächsten Schritt fort.

Schritt 4: Überprüfen Sie das Ergebnis des Experiments

Sie können überprüfen, ob die Instanzen wie erwartet durch das Experiment gestoppt und gestartet wurden.

Um das Ergebnis des Experiments zu überprüfen

1. Öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/> in einem neuen Browser-Tab oder Fenster. Auf diese Weise können Sie den Fortschritt des Experiments weiterhin in der AWS FIS-Konsole verfolgen und gleichzeitig das Ergebnis des Experiments in der EC2 Amazon-Konsole anzeigen.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wenn sich der Status der ersten Aktion von Ausstehend in Running (AWS FIS-Konsole) ändert, ändert sich der Status einer der Ziel-Instances von Running in Stopped (EC2 Amazon-Konsole).
4. Nach drei Minuten ändert sich der Status der ersten Aktion in Abgeschlossen, der Status der zweiten Aktion ändert sich in Wird ausgeführt und der Status der anderen Ziel-Instance ändert sich in Gestoppt.
5. Nach drei Minuten ändert sich der Status der zweiten Aktion in Abgeschlossen, der Status der Zielinstanzen ändert sich in Wird ausgeführt und der Status des Experiments ändert sich in Abgeschlossen.

Schritt 5: Bereinigen

Wenn Sie die EC2 Testinstanzen, die Sie für dieses Experiment erstellt haben, nicht mehr benötigen, können Sie sie beenden.

So beenden Sie die Instances

1. Öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Instances aus.

3. Wählen Sie beide Test-Instances aus und wählen Sie dann Instance state (Instance-Status), Terminate instance (Instance beenden).
4. Wählen Sie Terminate (Kündigen) aus, wenn Sie zur Bestätigung aufgefordert werden.

Wenn Sie die Versuchsvorlage nicht mehr benötigen, können Sie sie löschen.

Um eine Experimentvorlage mit der AWS FIS-Konsole zu löschen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Experimentvorlage löschen.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie die Eingabe ein **delete** und wählen Sie dann Experimentvorlage löschen.

Tutorial: CPU-Belastung auf einer Instance mithilfe von AWS FIS ausführen

Sie können den AWS AWS Fault Injection Service (FIS) verwenden, um zu testen, wie Ihre Anwendungen mit CPU-Belastung umgehen. Verwenden Sie dieses Tutorial, um eine Experimentvorlage zu erstellen, die AWS FIS verwendet, um ein vorkonfiguriertes SSM-Dokument auszuführen, das die CPU-Belastung auf einer Instanz ausführt. Das Tutorial verwendet eine Stopp-Bedingung, um das Experiment anzuhalten, wenn die CPU-Auslastung der Instanz einen konfigurierten Schwellenwert überschreitet.

Weitere Informationen finden Sie unter [the section called “Vorkonfigurierte AWS FIS SSM-Dokumente”](#).

Voraussetzungen

Bevor Sie AWS FIS zum Ausführen von CPU-Belastungen verwenden können, müssen Sie die folgenden Voraussetzungen erfüllen.

Erstellen einer IAM-Rolle

Erstellen Sie eine Rolle und fügen Sie eine Richtlinie hinzu, die es AWS FIS ermöglicht, die `aws:ssm:send-command` Aktion in Ihrem Namen zu verwenden. Weitere Informationen finden Sie unter [IAM-Rollen für AWS FIS-Experimente](#).

Überprüfen Sie den Zugriff auf FIS AWS

Stellen Sie sicher, dass Sie Zugriff auf AWS FIS haben. Weitere Informationen finden Sie unter Beispiele für [AWS FIS-Richtlinien](#).

Bereiten Sie eine EC2-Testinstanz vor

- Starten Sie eine EC2-Instanz mit Amazon Linux 2 oder Ubuntu, wie in den vorkonfigurierten SSM-Dokumenten gefordert.
- Die Instanz muss von SSM verwaltet werden. Um zu überprüfen, ob die Instanz von SSM verwaltet wird, öffnen Sie die [Fleet Manager-Konsole](#). Wenn die Instanz nicht von SSM verwaltet wird, stellen Sie sicher, dass der SSM-Agent installiert ist und dass der Instanz eine IAM-Rolle mit der Amazon-Richtlinie zugewiesen ist. `SSMManaged InstanceCore` Um den installierten SSM-Agent zu überprüfen, stellen Sie eine Verbindung zu Ihrer Instanz her und führen Sie den folgenden Befehl aus.

Amazon Linux 2

```
yum info amazon-ssm-agent
```

Ubuntu

```
apt list amazon-ssm-agent
```

- Aktivieren Sie die detaillierte Überwachung für die Instanz. Dadurch werden Daten in Zeitabständen von 1 Minute gegen Aufpreis bereitgestellt. Wählen Sie die Instanz aus und klicken Sie auf Aktionen, Überwachung und Fehlerbehebung, Detaillierte Überwachung verwalten.

Schritt 1: Erstellen Sie einen CloudWatch Alarm für eine Stopp-Bedingung

Konfigurieren Sie einen CloudWatch Alarm, sodass Sie das Experiment beenden können, wenn die CPU-Auslastung den von Ihnen angegebenen Schwellenwert überschreitet. Das folgende Verfahren legt den Schwellenwert auf 50% CPU-Auslastung für die Zielinstanz fest. Weitere Informationen finden Sie unter [Bedingungen beenden](#).

Um einen Alarm zu erstellen, der anzeigt, wenn die CPU-Auslastung einen Schwellenwert überschreitet

1. Öffnen Sie die Amazon-EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die Zielinstantz aus und wählen Sie Aktionen, Überwachung und Fehlerbehebung, CloudWatchAlarmerne verwalten aus.
4. Verwenden Sie für die Alarmbenachrichtigung den Schalter, um Amazon SNS SNS-Benachrichtigungen zu deaktivieren.
5. Verwenden Sie für Alarmschwellenwerte die folgenden Einstellungen:
 - Gruppieren Sie die Stichproben nach: Maximum
 - Art der zu samplenden Daten: CPU-Auslastung
 - Prozent: **50**
 - Zeitraum: **1 Minute**
6. Wenn Sie mit der Konfiguration des Alarms fertig sind, wählen Sie Erstellen.

Schritt 2: Erstellen Sie eine Experimentvorlage

Erstellen Sie die Experimentvorlage mit der AWS FIS-Konsole. In der Vorlage geben Sie an, dass die folgende Aktion ausgeführt werden soll: [AWSFIS-Runaws:ssm:send-command/](#) -CPU-Stress.

So erstellen Sie eine Experimentvorlage

1. AWS Öffnen <https://console.aws.amazon.com/fis/>Sie die FIS-Konsole unter.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Experimentvorlage erstellen aus.
4. Gehen Sie für Schritt 1, Vorlagendetails angeben, wie folgt vor:
 - a. Geben Sie unter Beschreibung und Name eine Beschreibung für die Vorlage ein.
 - b. Wählen Sie Weiter und fahren Sie mit Schritt 2, Aktionen und Ziele angeben, fort.
5. Nehmen Sie bei Aktionen die folgenden Einstellungen vor:
 - a. Wählen Sie Aktion hinzufügen aus.
 - b. Geben Sie einen Namen für die Aktion ein. Geben Sie z. B. ei **runCpuStress**.

- c. Wählen Sie als Aktionstyp AWSFIS-Run `aws:ssm:send-command/ -CPU-Stress` aus. Dadurch wird der ARN des SSM-Dokuments automatisch zum Dokument-ARN hinzugefügt.
- d. Behalten Sie für Target das Ziel bei, das AWS FIS für Sie erstellt.
- e. Geben Sie für Aktionsparameter, Dokumentparameter Folgendes ein:

```
{"DurationSeconds": "120"}
```

- f. Geben Sie für Aktionsparameter für Dauer den Wert 5 Minuten (PT5M) ein.
 - g. Wählen Sie Speichern.
6. Führen Sie für Targets (Ziele) Folgendes aus:
- a. Wählen Sie Bearbeiten für das Ziel, das AWS FIS im vorherigen Schritt automatisch für Sie erstellt hat.
 - b. Ersetzen Sie den Standardnamen durch einen aussagekräftigeren Namen. Geben Sie z. B. **testInstance**.
 - c. Stellen Sie sicher, dass der Ressourcentyp `aws:ec2:instance` lautet.
 - d. Wählen Sie für Target-Methode die Option Resource IDs und dann die ID der Testinstanz aus.
 - e. Wählen Sie für den Auswahlmodus die Option Alle aus.
 - f. Wählen Sie Speichern.
7. Wählen Sie Weiter, um mit Schritt 3, Servicezugriff konfigurieren, fortzufahren.
8. Wählen Sie für Service Access die Option Bestehende IAM-Rolle verwenden und wählen Sie dann die IAM-Rolle aus, die Sie wie in den Voraussetzungen für dieses Tutorial beschrieben erstellt haben. Wenn Ihre Rolle nicht angezeigt wird, stellen Sie sicher, dass sie über die erforderliche Vertrauensstellung verfügt. Weitere Informationen finden Sie unter [the section called “Die Rolle des Experiments”](#).
9. Wählen Sie Weiter, um mit Schritt 4, Optionale Einstellungen konfigurieren, fortzufahren.
10. Wählen Sie unter Stoppbedingungen den CloudWatch Alarm aus, den Sie in Schritt 1 erstellt haben.
11. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus und geben Sie einen Tag-Schlüssel und einen Tag-Wert an. Die von Ihnen hinzugefügten Tags werden auf Ihre Experimentvorlage angewendet, nicht auf die Experimente, die mit der Vorlage ausgeführt werden.
12. Wählen Sie Weiter, um mit Schritt 5, Überprüfen und erstellen, fortzufahren.

13. Prüfen Sie die Vorlage und wählen Sie Experimentvorlage erstellen aus. Wenn Sie zur Bestätigung aufgefordert werden `create`, geben Sie ein und wählen Sie dann Experimentvorlage erstellen.

(Optional) Um die JSON-Version der Experimentvorlage anzuzeigen

Wählen Sie die Registerkarte Exportieren. Im Folgenden finden Sie ein Beispiel für die JSON-Datei, die mit der vorherigen Konsolenprozedur erstellt wurde.

```
{
  "description": "Test CPU stress predefined SSM document",
  "targets": {
    "testInstance": {
      "resourceType": "aws:ec2:instance",
      "resourceArns": [
        "arn:aws:ec2:region:123456789012:instance/instance_id"
      ],
      "selectionMode": "ALL"
    }
  },
  "actions": {
    "runCpuStress": {
      "actionId": "aws:ssm:send-command",
      "parameters": {
        "documentArn": "arn:aws:ssm:region::document/AWSFIS-Run-CPU-Stress",
        "documentParameters": "{\"DurationSeconds\":\"120\"}",
        "duration": "PT5M"
      },
      "targets": {
        "Instances": "testInstance"
      }
    }
  },
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": "arn:aws:cloudwatch:region:123456789012:alarm:awsec2-instance_id-
GreaterThanOrEqualToThreshold-CPUUtilization"
    }
  ],
  "roleArn": "arn:aws:iam::123456789012:role/AllowFISSSMActions",
  "tags": {}
}
```

```
}
```

Schritt 3: Starten Sie das Experiment

Wenn Sie mit der Erstellung Ihrer Experimentvorlage fertig sind, können Sie sie verwenden, um ein Experiment zu starten.

So starten Sie ein Experiment

1. Sie sollten sich auf der Detailseite der Experimentvorlage befinden, die Sie gerade erstellt haben. Andernfalls wählen Sie Experimentvorlagen und dann die ID der Experimentvorlage aus, um die Detailseite zu öffnen.
2. Wählen Sie Start Experiment (Experiment starten) aus.
3. (Optional) Um Ihrem Experiment ein Tag hinzuzufügen, wählen Sie Neues Tag hinzufügen und geben Sie einen Tag-Schlüssel und einen Tag-Wert ein.
4. Wählen Sie Start Experiment (Experiment starten) aus. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **start** ein. Wählen Sie Start Experiment (Experiment starten) aus.

Schritt 4: Verfolgen Sie den Fortschritt des Experiments

Sie können den Fortschritt eines laufenden Experiments verfolgen, bis das Experiment abgeschlossen, beendet oder fehlschlägt.

Um den Fortschritt eines Experiments zu verfolgen

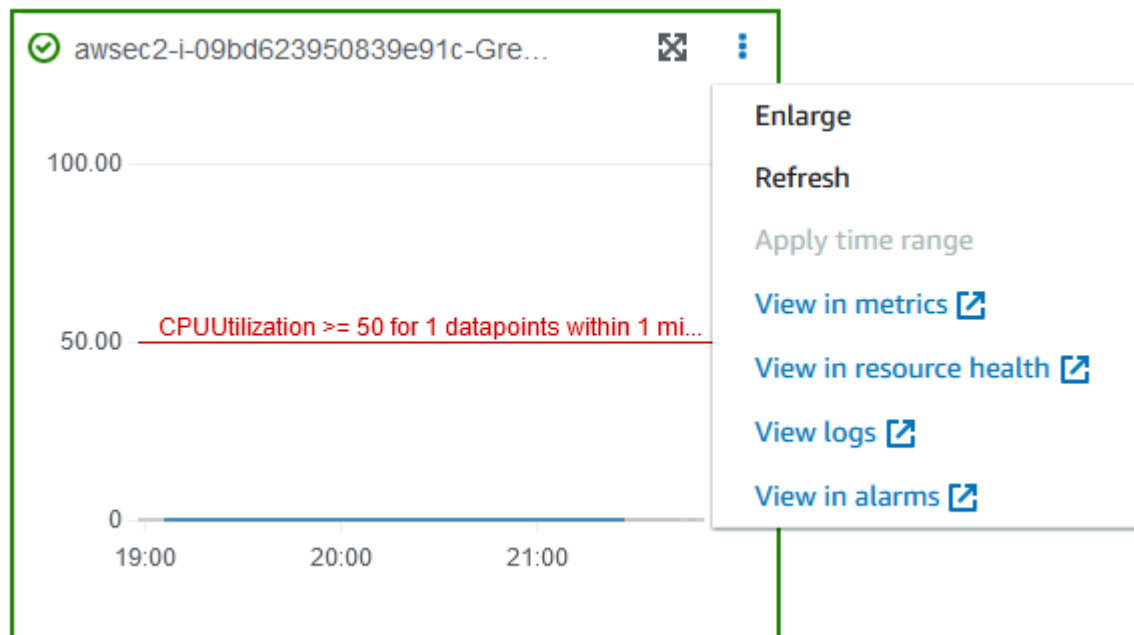
1. Sie sollten sich auf der Detailseite für das Experiment befinden, das Sie gerade gestartet haben. Andernfalls wählen Sie Experimente und dann die ID des Experiments aus, um die Detailseite für das Experiment zu öffnen.
2. Um den Status des Experiments anzuzeigen, aktivieren Sie im Detailbereich die Option Status. Weitere Informationen finden Sie unter [Versuchsstatus](#).
3. Wenn der Status des Experiments „Wird ausgeführt“ lautet, fahren Sie mit dem nächsten Schritt fort.

Schritt 5: Überprüfen Sie die Ergebnisse des Experiments

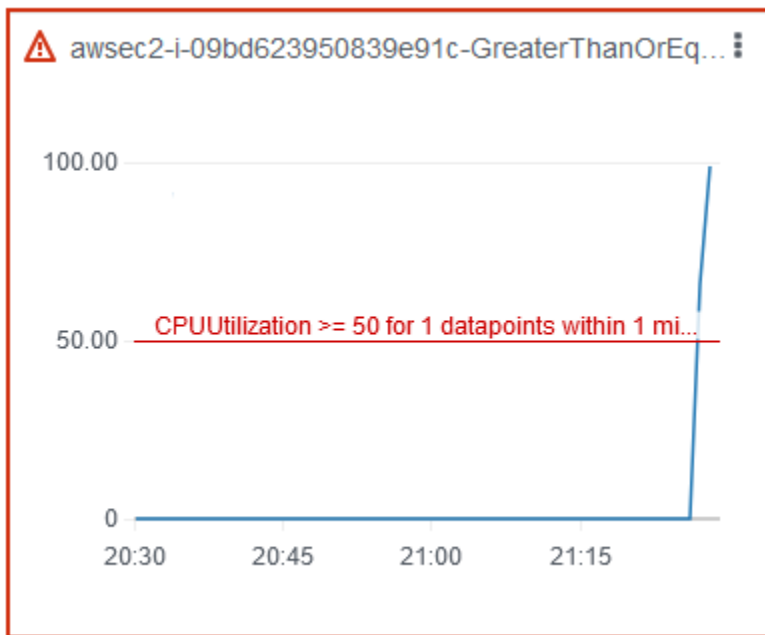
Sie können die CPU-Auslastung Ihrer Instance überwachen, während das Experiment läuft. Wenn die CPU-Auslastung den Schwellenwert erreicht, wird der Alarm ausgelöst und das Experiment wird durch die Stopp-Bedingung angehalten.

Um die Ergebnisse des Experiments zu überprüfen

1. Wählen Sie die Registerkarte Stoppbedingungen. Der grüne Rand und das grüne Häkchensymbol zeigen an, dass sich der Alarm im Ausgangszustand befindet OK. Die rote Linie gibt den Alarmschwellenwert an. Wenn Sie ein detaillierteres Diagramm bevorzugen, wählen Sie im Widget-Menü die Option „Vergrößern“.



2. Wenn die CPU-Auslastung den Schwellenwert überschreitet, weisen der rote Rand und das rote Ausrufezeichen auf der Registerkarte Stoppbedingungen darauf hin, dass der Alarmstatus auf geändert wurde. ALARM Im Detailbereich lautet der Status des Experiments Gestoppt. Wenn Sie den Status auswählen, wird die Meldung „Das Experiment wurde durch die Abbruchbedingung angehalten“ angezeigt.



3. Wenn die CPU-Auslastung unter den Schwellenwert fällt, weisen der grüne Rand und das grüne Häkchensymbol darauf hin, dass der Alarmstatus auf OK geändert wurde.
4. (Optional) Wählen Sie im Widget-Menü die Option In Alarmen anzeigen aus. Dadurch wird die Seite mit den Alarmdetails in der CloudWatch Konsole geöffnet, auf der Sie weitere Informationen zum Alarm abrufen oder die Alarmeinstellungen bearbeiten können.

Schritt 6: Bereinigen

Wenn Sie die EC2-Testinstanz, die Sie für dieses Experiment erstellt haben, nicht mehr benötigen, können Sie sie beenden.

Beenden der Instances

1. Öffnen Sie die Amazon-EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die Testinstanzen aus und wählen Sie Instanzstatus, Instanz beenden aus.
4. Wählen Sie Terminate (Kündigen) aus, wenn Sie zur Bestätigung aufgefordert werden.

Wenn Sie die Experimentvorlage nicht mehr benötigen, können Sie sie löschen.

Um eine Experimentvorlage mit der AWS FIS-Konsole zu löschen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Experimentvorlage löschen.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie die Eingabe ein **delete** und wählen Sie dann Experimentvorlage löschen.

Tutorial: Testen Sie Spot-Instance-Unterbrechungen mit AWS FIS

Spot-Instances nutzen verfügbare EC2 Reservekapazitäten und bieten einen discount von bis zu 90% im Vergleich zu On-Demand-Preisen. Amazon EC2 kann Ihre Spot-Instances jedoch unterbrechen, wenn die Kapazität wieder benötigt wird. Wenn Sie Spot-Instances verwenden, müssen Sie auf mögliche Unterbrechungen vorbereitet sein. Weitere Informationen finden Sie unter [Spot-Instance-Unterbrechungen](#) im EC2 Amazon-Benutzerhandbuch.

Sie können den AWS AWS Fault Injection Service (FIS) verwenden, um zu testen, wie Ihre Anwendungen mit einer Spot-Instance-Unterbrechung umgehen. Verwenden Sie dieses Tutorial, um eine Versuchsvorlage zu erstellen, die die AWS `aws:ec2:send-spot-instance-interruptions` FIS-Aktion verwendet, um eine Ihrer Spot-Instances zu unterbrechen.

Alternativ können Sie das Experiment über die EC2 Amazon-Konsole starten. Weitere Informationen finden Sie unter [Initiieren einer Spot-Instance-Unterbrechung](#) im EC2 Amazon-Benutzerhandbuch.

Voraussetzungen

Bevor Sie AWS FIS verwenden können, um eine Spot-Instance zu unterbrechen, müssen Sie die folgenden Voraussetzungen erfüllen.

1. Erstellen einer IAM-Rolle

Erstellen Sie eine Rolle und fügen Sie eine Richtlinie hinzu, die es AWS FIS ermöglicht, die `aws:ec2:send-spot-instance-interruptions` Aktion in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter [IAM-Rollen für AWS FIS-Experimente](#).

2. Überprüfen Sie den Zugriff auf FIS AWS

Stellen Sie sicher, dass Sie Zugriff auf AWS FIS haben. Weitere Informationen finden Sie unter Beispiele für [AWS FIS-Richtlinien](#).

3. (Optional) Erstellen Sie eine Spot-Instance-Anfrage

Wenn Sie eine neue Spot-Instance für dieses Experiment verwenden möchten, verwenden Sie den Befehl [run-instances](#), um eine Spot-Instance anzufordern. Standardmäßig werden Spot-Instances, die unterbrochen wurden, beendet. Wenn Sie das Unterbrechungsverhalten auf einstellenstop, müssen Sie auch den Typ auf einstellenpersistent. Stellen Sie für dieses Tutorial das Unterbrechungsverhalten nicht auf einhibernate, da der Ruhezustand sofort beginnt.

```
aws ec2 run-instances \  
  --image-id ami-0ab193018fEXAMPLE \  
  --instance-type "t2.micro" \  
  --count 1 \  
  --subnet-id subnet-1234567890abcdef0 \  
  --security-group-ids sg-111222333444aaab \  
  --instance-market-options file://spot-options.json \  
  --query Instances[*].InstanceId
```

Das folgende Beispiel zeigt eine spot-options.json-Datei.

```
{  
  "MarketType": "spot",  
  "SpotOptions": {  
    "SpotInstanceType": "persistent",  
    "InstanceInterruptionBehavior": "stop"  
  }  
}
```

Die --query Option im Beispielbefehl sorgt dafür, dass der Befehl nur die Instance-ID der Spot-Instance zurückgibt. Es folgt eine Beispielausgabe.

```
[  
  "i-0abcdef1234567890"  
]
```

4. Fügen Sie ein Tag hinzu, damit AWS FIS die Ziel-Spot-Instance identifizieren kann

Verwenden Sie den Befehl [create-tags](#), um das Tag hinzuzufügen Name=interruptMe zu Ihrer Ziel-Spot-Instance.

```
aws ec2 create-tags \  
  --resources i-0abcdef1234567890 \  
  --tags Key=Name,Value=interruptMe
```

Schritt 1: Erstellen Sie eine Experimentvorlage

Erstellen Sie die Experimentvorlage mit der AWS FIS-Konsole. In der Vorlage geben Sie die Aktion an, die ausgeführt werden soll. Die Aktion unterbricht die Spot-Instance mit dem angegebenen Tag. Wenn es mehr als eine Spot-Instance mit dem Tag gibt, wählt AWS FIS nach dem Zufallsprinzip eine davon aus.

Um eine Versuchsvorlage zu erstellen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Experimentvorlage erstellen aus.
4. Gehen Sie für Schritt 1, Vorlagendetails angeben, wie folgt vor:
 - a. Geben Sie unter Beschreibung und Name eine Beschreibung und einen Namen für die Vorlage ein.
 - b. Wählen Sie Weiter und fahren Sie mit Schritt 2, Aktionen und Ziele angeben, fort.
5. Nehmen Sie bei Aktionen die folgenden Einstellungen vor:
 - a. Wählen Sie Aktion hinzufügen aus.
 - b. Geben Sie einen Namen für die Aktion ein. Geben Sie z. B. **interruptSpotInstance**.
 - c. Wählen Sie als Aktionstyp `aws:ec2:` aus. `send-spot-instance-interruptions`
 - d. Behalten Sie für Target das Ziel bei, das AWS FIS für Sie erstellt.
 - e. Geben Sie für Aktionsparameter, Dauer vor der Unterbrechung, den Wert 2 Minuten (PT2M) ein.
 - f. Wählen Sie Save (Speichern) aus.
6. Führen Sie für Targets (Ziele) Folgendes aus:
 - a. Wählen Sie Bearbeiten für das Ziel, das AWS FIS im vorherigen Schritt automatisch für Sie erstellt hat.
 - b. Ersetzen Sie den Standardnamen durch einen aussagekräftigeren Namen. Geben Sie z. B. **oneSpotInstance**.

- c. Stellen Sie sicher, dass der Ressourcentyp `aws:ec2:spot-instance` lautet.
 - d. Wählen Sie für Target-Methode die Option Ressourcen-Tags, Filter und Parameter aus.
 - e. Wählen Sie für Ressourcen-Tags die Option Neues Tag hinzufügen aus und geben Sie den Tag-Schlüssel und den Tag-Wert ein. Verwenden Sie das Tag, das Sie der Spot-Instance hinzugefügt haben, um zu unterbrechen, wie in den Voraussetzungen für dieses Tutorial beschrieben.
 - f. Wählen Sie für Ressourcenfilter die Option Neuen Filter hinzufügen und geben Sie **State.Name** als Pfad und **running** als Wert ein.
 - g. Wählen Sie für den Auswahlmodus die Option Anzahl aus. Geben Sie für Anzahl der Ressourcen den Wert ein**1**.
 - h. Wählen Sie Save (Speichern) aus.
7. Wählen Sie Weiter, um mit Schritt 3, Servicezugriff konfigurieren, fortzufahren.
 8. Wählen Sie für Service Access die Option Bestehende IAM-Rolle verwenden und wählen Sie dann die IAM-Rolle aus, die Sie wie in den Voraussetzungen für dieses Tutorial beschrieben erstellt haben. Wenn Ihre Rolle nicht angezeigt wird, stellen Sie sicher, dass sie über die erforderliche Vertrauensstellung verfügt. Weitere Informationen finden Sie unter [the section called "Die Rolle des Experiments"](#).
 9. Wählen Sie Weiter, um mit Schritt 4, Optionale Einstellungen konfigurieren, fortzufahren.
 10. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus und geben Sie einen Tag-Schlüssel und einen Tag-Wert an. Die von Ihnen hinzugefügten Tags werden auf Ihre Experimentvorlage angewendet, nicht auf die Experimente, die mit der Vorlage ausgeführt werden.
 11. Wählen Sie Weiter, um mit Schritt 5, Überprüfen und erstellen, fortzufahren.
 12. Prüfen Sie die Vorlage und wählen Sie Experimentvorlage erstellen. Wenn Sie zur Bestätigung aufgefordert werden `create`, geben Sie ein und wählen Sie dann Experimentvorlage erstellen.

(Optional) Um die JSON-Version der Experimentvorlage anzuzeigen

Wählen Sie die Registerkarte Exportieren. Im Folgenden finden Sie ein Beispiel für das JSON, das mit der vorherigen Konsolenprozedur erstellt wurde.

```
{
  "description": "Test Spot Instance interruptions",
  "targets": {
    "oneSpotInstance": {
```

```

        "resourceType": "aws:ec2:spot-instance",
        "resourceTags": {
            "Name": "interruptMe"
        },
        "filters": [
            {
                "path": "State.Name",
                "values": [
                    "running"
                ]
            }
        ],
        "selectionMode": "COUNT(1)"
    }
},
"actions": {
    "interruptSpotInstance": {
        "actionId": "aws:ec2:send-spot-instance-interruptions",
        "parameters": {
            "durationBeforeInterruption": "PT2M"
        },
        "targets": {
            "SpotInstances": "oneSpotInstance"
        }
    }
},
"stopConditions": [
    {
        "source": "none"
    }
],
"roleArn": "arn:aws:iam::123456789012:role/AllowFISSpotInterruptionActions",
"tags": {
    "Name": "my-template"
}
}

```

Schritt 2: Starten Sie das Experiment

Wenn Sie mit der Erstellung Ihrer Experimentvorlage fertig sind, können Sie sie verwenden, um ein Experiment zu starten.

Um ein Experiment zu starten

1. Sie sollten sich auf der Detailseite der Experimentvorlage befinden, die Sie gerade erstellt haben. Andernfalls wählen Sie Experimentvorlagen und dann die ID der Experimentvorlage aus, um die Detailseite zu öffnen.
2. Wählen Sie Start Experiment (Experiment starten) aus.
3. (Optional) Um Ihrem Experiment ein Tag hinzuzufügen, wählen Sie Neues Tag hinzufügen und geben Sie einen Tag-Schlüssel und einen Tag-Wert ein.
4. Wählen Sie Start Experiment (Experiment starten) aus. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den **start** Text ein und wählen Sie Experiment starten.

Schritt 3: Verfolgen Sie den Fortschritt des Experiments

Sie können den Fortschritt eines laufenden Experiments verfolgen, bis das Experiment abgeschlossen, gestoppt oder fehlgeschlagen ist.

Um den Fortschritt eines Experiments zu verfolgen

1. Sie sollten sich auf der Detailseite für das Experiment befinden, das Sie gerade gestartet haben. Andernfalls wählen Sie Experimente und dann die ID des Experiments aus, um die Detailseite zu öffnen.
2. Um den Status des Experiments einzusehen, aktivieren Sie im Detailbereich die Option Status. Weitere Informationen finden Sie unter [Versuchsstatus](#).
3. Wenn der Status des Experiments „Wird ausgeführt“ lautet, fahren Sie mit dem nächsten Schritt fort.

Schritt 4: Überprüfen Sie das Versuchsergebnis

Wenn die Aktion für dieses Experiment abgeschlossen ist, passiert Folgendes:

- Die Ziel-Spot-Instance erhält eine [Empfehlung zur Neuverteilung der Instance](#).
- Zwei Minuten, bevor Amazon Ihre [Instance EC2 beendet oder stoppt, wird eine Benachrichtigung zur Unterbrechung der Spot-Instance](#) ausgegeben.
- Nach zwei Minuten wird die Spot-Instance beendet oder gestoppt.
- Eine Spot-Instance, die von AWS FIS gestoppt wurde, bleibt gestoppt, bis Sie sie neu starten.

Um zu überprüfen, ob die Instance durch das Experiment unterbrochen wurde

1. Öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Öffnen Sie im Navigationsbereich Spot Requests (Spot-Anforderungen) und Instances in separaten Browser-Registerkarten oder -Fenstern.
3. Wählen Sie unter Spot Requests (Spot-Anforderungen) die Spot-Instance-Anforderung aus. Der ursprüngliche Status ist `fulfilled`. Nach Abschluss des Experiments ändert sich der Status wie folgt:
 - `terminate`- Der Status ändert sich zu `instance-terminated-by-experiment`.
 - `stop`- Der Status ändert sich zu `marked-for-stop-by-experiment` und dann `instance-stopped-by-experiment`.
4. Wählen Sie unter Instances die Spot Instance aus. Der ursprüngliche Status ist `Running`. Zwei Minuten, nachdem Sie die Benachrichtigung über die Unterbrechung der Spot-Instance erhalten haben, ändert sich der Status wie folgt:
 - `stop`— Der Status ändert sich zu `Stopping` und dann `Stopped`.
 - `terminate`- Der Status ändert sich zu `Shutting-down` und dann `Terminated`.

Schritt 5: Bereinigen

Wenn Sie die Test-Spot-Instance für dieses Experiment mit einem Unterbrechungsverhalten von erstellt haben `stop` und Sie sie nicht mehr benötigen, können Sie die Spot-Instance-Anfrage stornieren und die Spot-Instance beenden.

Um die Anfrage zu stornieren und die Instance zu beenden, verwenden Sie AWS CLI

1. Verwenden Sie den [cancel-spot-instance-requests](#) Befehl, um die Spot-Instance-Anfrage zu stornieren.

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-ksie869j
```

2. Verwenden Sie den Befehl [terminate-instances](#), um die Instance zu beenden.

```
aws ec2 terminate-instances --instance-ids i-0abcdef1234567890
```

Wenn Sie die Experimentvorlage nicht mehr benötigen, können Sie sie löschen.

Um eine Experimentvorlage mit der AWS FIS-Konsole zu löschen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Experimentvorlage löschen.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie die Eingabe ein **delete** und wählen Sie dann Experimentvorlage löschen.

Tutorial: Simulieren Sie ein Konnektivitätsereignis

Sie können den AWS AWS Fault Injection Service (FIS) verwenden, um eine Vielzahl von Konnektivitätsereignissen zu simulieren. AWS FIS simuliert Konnektivitätsereignisse, indem Netzwerkverbindungen auf eine der folgenden Arten blockiert werden:

- **all**— Verweigert jeglichen Datenverkehr, der in das Subnetz eingeht und es verlässt. Beachten Sie, dass diese Option subnetzinternen Verkehr zulässt, einschließlich Verkehr zu und von Netzwerkschnittstellen im Subnetz.
- **availability-zone**— Verweigert internen VPC-Verkehr zu und von Subnetzen in anderen Availability Zones.
- **dynamodb**— Verweigert den Verkehr zum und vom regionalen Endpunkt für DynamoDB in der aktuellen Region.
- **prefix-list**— Verweigert den Verkehr zur und von der angegebenen Präfixliste.
- **s3**— Verweigert den Verkehr zum und vom regionalen Endpunkt für Amazon S3 in der aktuellen Region.
- **s3express**— Verweigert den Verkehr zum und vom zonalen Endpunkt für Amazon S3 Express One Zone in der AZ der Zielsubnetze. Die Zielsubnetze müssen sich dort befinden, AZs wo S3 Express One Zone derzeit verfügbar ist. Weitere Informationen finden Sie unter [S3 Express One Zone Availability Zones and Regions](#).
- **vpc**— Verweigert den Verkehr, der in die VPC eingeht und sie verlässt.

Verwenden Sie dieses Tutorial, um eine Versuchsvorlage zu erstellen, die die AWS `aws:network:disrupt-connectivity` FIS-Aktion verwendet, um einen Verbindungsverlust mit Amazon S3 in einem Zielsubnetz einzuführen.

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie eine AWS FIS-Versuchsvorlage](#)
- [Schritt 2: Pingen Sie einen Amazon S3 S3-Endpunkt](#)
- [Schritt 3: Starten Sie Ihr AWS FIS-Experiment](#)
- [Schritt 4: Verfolgen Sie den Fortschritt Ihres AWS FIS-Experiments](#)
- [Schritt 5: Überprüfen Sie die Amazon S3 S3-Netzwerkunterbrechung](#)
- [Schritt 5: Bereinigen](#)

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, benötigen Sie eine Rolle mit den entsprechenden Berechtigungen in Ihrer AWS-Konto Amazon EC2-Instance und einer Testinstanz von Amazon EC2:

Eine Rolle mit Berechtigungen in Ihrem AWS-Konto

Erstellen Sie eine Rolle und fügen Sie eine Richtlinie hinzu, die AWS es FIS ermöglicht, die `aws:network:disrupt-connectivity` Aktion in Ihrem Namen durchzuführen.

Ihre IAM-Rolle erfordert die folgende Richtlinie:

- [AWSFaultInjectionSimulatorNetworkAccess](#)— Erteilt AWS FIS-Serviceberechtigungen in Amazon EC2 EC2-Netzwerken und anderen erforderlichen Diensten zur Durchführung von AWS FIS-Aktionen im Zusammenhang mit der Netzwerkinfrastruktur.

Note

Der Einfachheit halber verwendet dieses Tutorial eine AWS verwaltete Richtlinie. Für den produktiven Einsatz empfehlen wir, dass Sie stattdessen nur die für Ihren Anwendungsfall erforderlichen Mindestberechtigungen gewähren.

Weitere Informationen zum Erstellen einer IAM-Rolle finden Sie unter [IAM-Rollen für AWS FIS-Experimente \(AWS CLI\)](#) oder [Erstellen einer IAM-Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Eine Amazon EC2 EC2-Testinstanz

Starten Sie eine Amazon EC2-Testinstanz und stellen Sie eine Verbindung zu ihr her. Sie können das folgende Tutorial verwenden, um eine Amazon EC2 EC2-Instance zu starten und eine Verbindung zu ihr herzustellen: [Tutorial: Erste Schritte mit Amazon EC2 EC2-Linux-Instances](#) im Amazon EC2 EC2-Benutzerhandbuch.

Schritt 1: Erstellen Sie eine AWS FIS-Versuchsvorlage

Erstellen Sie die Experimentvorlage mithilfe des AWS FIS AWS-Managementkonsole. Eine AWS FIS-Vorlage besteht aus Aktionen, Zielen, Stoppbedingungen und einer Experimentierrolle. Weitere Informationen zur Funktionsweise der Vorlagen finden Sie unter [Experimentvorlagen für AWS FIS](#).

Bevor Sie beginnen, stellen Sie sicher, dass Sie Folgendes bereit haben:

- Eine IAM-Rolle mit den richtigen Berechtigungen.
- Eine Amazon-EC2-Instance.
- Die Subnetz-ID Ihrer Amazon EC2 EC2-Instance.

So erstellen Sie eine Experimentvorlage

1. Öffnen Sie die AWS FIS-Konsole unter. <https://console.aws.amazon.com/fis/>
2. Wählen Sie im linken Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Experimentvorlage erstellen.
4. Gehen Sie für Schritt 1, Vorlagendetails angeben, wie folgt vor:
 - a. Geben Sie unter Beschreibung und Name eine Beschreibung für die Vorlage ein, z. Amazon S3 Network Disrupt Connectivity B.
 - b. Wählen Sie Weiter und fahren Sie mit Schritt 2, Aktionen und Ziele angeben, fort.
5. Wählen Sie unter Aktionen die Option Aktion hinzufügen aus.
 - a. Geben Sie als Namen `eindisruptConnectivity`.
 - b. Wählen Sie als Aktionstyp `aws:network:disrupt-connectivity` aus.
 - c. Stellen Sie unter Aktionsparameter die Dauer auf ein. 2 minutes
 - d. Wählen Sie unter Umfang die Option s3 aus.
 - e. Wählen Sie oben Speichern aus.
6. Unter Ziele sollten Sie das Ziel sehen, das automatisch erstellt wurde. Wählen Sie Bearbeiten aus.

- a. Stellen Sie sicher, dass der Ressourcentyp `istaws:ec2:subnet`.
 - b. Wählen Sie unter Zielmethode die Option Ressource IDs aus und wählen Sie dann in den Schritten [Voraussetzungen](#) das Subnetz aus, das Sie bei der Erstellung Ihrer Amazon EC2 EC2-Instance verwendet haben.
 - c. Stellen Sie sicher, dass der Auswahlmodus auf Alle gesetzt ist.
 - d. Wählen Sie Speichern.
7. Wählen Sie Weiter, um mit Schritt 3, Dienstzugriff konfigurieren, fortzufahren.
 8. Wählen Sie unter Dienstzugriff die IAM-Rolle aus, die Sie wie in den [Voraussetzungen](#) für dieses Tutorial beschrieben erstellt haben. Wenn Ihre Rolle nicht angezeigt wird, stellen Sie sicher, dass sie über die erforderliche Vertrauensstellung verfügt. Weitere Informationen finden Sie unter [the section called "Die Rolle des Experiments"](#).
 9. Wählen Sie Weiter, um mit Schritt 4, Optionale Einstellungen konfigurieren, fortzufahren.
 10. (Optional) Unter Bedingungen beenden können Sie einen CloudWatch Alarm auswählen, um das Experiment zu beenden, wenn die Bedingung eintritt. Weitere Informationen finden Sie unter [Stopbedingungen für AWS FIS](#).
 11. (Optional) Unter Logs können Sie einen Amazon S3 S3-Bucket auswählen oder Logs CloudWatch für Ihr Experiment an senden.
 12. Wählen Sie Weiter, um mit Schritt 5, Überprüfen und Erstellen, fortzufahren.
 13. Prüfen Sie die Vorlage und wählen Sie Experimentvorlage erstellen aus. Wenn Sie zur Bestätigung aufgefordert werden `create`, geben Sie ein und wählen Sie dann Experimentvorlage erstellen.

Schritt 2: Pingen Sie einen Amazon S3 S3-Endpunkt

Stellen Sie sicher, dass Ihre Amazon EC2 EC2-Instance einen Amazon S3 S3-Endpunkt erreichen kann.

1. Connect zu der Amazon EC2 EC2-Instance her, die Sie in den Schritten „[Voraussetzungen](#)“ erstellt haben.

Informationen zur Fehlerbehebung finden Sie unter [Problembehandlung beim Herstellen einer Verbindung zu Ihrer Instance](#) im Amazon EC2 EC2-Benutzerhandbuch.

2. Prüfen Sie, AWS-Region wo sich Ihre Instance befindet. Sie können dies in der Amazon EC2 EC2-Konsole oder durch Ausführen des folgenden Befehls tun.

hostname

Wenn Sie beispielsweise eine Amazon EC2 EC2-Instance gestartet haben us-west-2, sehen Sie die folgende Ausgabe.

```
[ec2-user@ip-172.16.0.0 ~]$ hostname  
ip-172.16.0.0.us-west-2.compute.internal
```

3. Pingen Sie einen Amazon S3 S3-Endpunkt in Ihrem AWS-Region. Ersetzen Sie **AWS-Region** durch Ihre Region.

```
ping -c 1 s3.AWS-Region.amazonaws.com
```

Für die Ausgabe sollte ein erfolgreicher Ping mit einem Paketverlust von 0% angezeigt werden, wie im folgenden Beispiel gezeigt.

```
PING s3.us-west-2.amazonaws.com (x.x.x.x) 56(84) bytes of data.  
64 bytes from s3-us-west-2.amazonaws.com (x.x.x.x: icmp_seq=1 ttl=249 time=1.30 ms  
  
--- s3.us-west-2.amazonaws.com ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 1.306/1.306/1.306/0.000 ms
```

Schritt 3: Starten Sie Ihr AWS FIS-Experiment

Starten Sie ein Experiment mit der Experimentvorlage, die Sie gerade erstellt haben.

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im linken Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die ID der von Ihnen erstellten Experimentvorlage aus, um die zugehörige Detailseite zu öffnen.
4. Wählen Sie Start Experiment (Experiment starten) aus.
5. (Optional) Fügen Sie auf der Bestätigungsseite Tags für Ihr Experiment hinzu.
6. Wählen Sie auf der Bestätigungsseite die Option Experiment starten aus.

Schritt 4: Verfolgen Sie den Fortschritt Ihres AWS FIS-Experiments

Sie können den Fortschritt eines laufenden Experiments verfolgen, bis das Experiment abgeschlossen, gestoppt oder fehlgeschlagen ist.

1. Sie sollten sich auf der Detailseite für das Experiment befinden, das Sie gerade gestartet haben. Wenn nicht, wählen Sie Experimente und dann die ID des Experiments aus, um die zugehörige Detailseite zu öffnen.
2. Um den Status des Experiments einzusehen, klicken Sie im Detailbereich auf Status. Weitere Informationen finden Sie unter [Versuchsstatus](#).
3. Wenn der Status des Experiments „Wird ausgeführt“ lautet, fahren Sie mit dem nächsten Schritt fort.

Schritt 5: Überprüfen Sie die Amazon S3 S3-Netzwerkunterbrechung

Sie können den Fortschritt des Experiments überprüfen, indem Sie den Amazon S3 S3-Endpunkt anpingen.

- Pingen Sie von Ihrer Amazon EC2 EC2-Instance aus den Amazon S3 S3-Endpunkt in Ihrem AWS-Region. Ersetzen Sie *AWS-Region* durch Ihre Region.

```
ping -c 1 s3.AWS-Region.amazonaws.com
```

Für die Ausgabe sollte ein erfolgloser Ping mit einem Paketverlust von 100% angezeigt werden, wie im folgenden Beispiel gezeigt.

```
ping -c 1 s3.us-west-2.amazonaws.com
PING s3.us-west-2.amazonaws.com (x.x.x.x) 56(84) bytes of data.

--- s3.us-west-2.amazonaws.com ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

Schritt 5: Bereinigen

Wenn Sie die Amazon EC2 EC2-Instance, die Sie für dieses Experiment erstellt haben, oder die AWS FIS-Vorlage nicht mehr benötigen, können Sie sie entfernen.

Um die Amazon EC2 EC2-Instance zu entfernen

1. Öffnen Sie die Amazon-EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die Test-Instance, dann Instance-Status und anschließend Terminate Instance aus.
4. Wählen Sie Terminate (Kündigen) aus, wenn Sie zur Bestätigung aufgefordert werden.

Um die Experimentvorlage mit der AWS FIS-Konsole zu löschen

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Experimentvorlage löschen.
4. Wenn Sie zur Bestätigung aufgefordert werden `delete`, geben Sie die Eingabe ein und wählen Sie dann Experimentvorlage löschen.

Tutorial: Ein wiederkehrendes Experiment planen

Mit dem AWS AWS Fault Injection Service (FIS) können Sie Fault-Injection-Experimente an Ihren AWS Workloads durchführen. Diese Experimente werden auf Vorlagen ausgeführt, die eine oder mehrere Aktionen zur Ausführung auf bestimmten Zielen enthalten. Wenn Sie dies auch verwenden Amazon EventBridge, können Sie Ihre Experimente als einmalige Aufgabe oder als wiederkehrende Aufgaben planen.

Verwenden Sie dieses Tutorial, um einen EventBridge Zeitplan zu erstellen, der alle 5 Minuten AWS eine FIS-Experimentvorlage ausführt.

Aufgaben

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie eine IAM-Rolle und -Richtlinie](#)
- [Schritt 2: Erstellen Sie einen Amazon EventBridge Scheduler](#)
- [Schritt 3: Verifizieren Sie Ihr Experiment](#)
- [Schritt 4: Bereinigen](#)

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, benötigen Sie eine AWS FIS-Experimentvorlage, die Sie nach einem Zeitplan ausführen möchten. Wenn Sie bereits über eine funktionierende Versuchsvorlage verfügen, notieren Sie sich die Vorlagen-ID und AWS-Region. Andernfalls können Sie eine Vorlage erstellen [the section called "Testinstanz beenden und starten"](#), indem Sie den Anweisungen unter folgen und dann zu diesem Tutorial zurückkehren.

Schritt 1: Erstellen Sie eine IAM-Rolle und -Richtlinie

So erstellen Sie eine IAM-Rolle und -Richtlinie

1. Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.
2. Wählen Sie im linken Navigationsbereich Rollen und anschließend Rolle erstellen aus.
3. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie und fügen Sie dann den folgenden Ausschnitt ein, damit Amazon EventBridge Scheduler die Rolle in Ihrem Namen übernehmen kann.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "scheduler.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Wählen Sie Weiter aus.

4. Wählen Sie unter Berechtigungen hinzufügen die Option Richtlinie erstellen aus.
5. Wählen Sie JSON und fügen Sie dann die folgende Richtlinie ein. Ersetzen Sie den *your-experiment-template-id* Wert durch die Vorlagen-ID Ihres Experiments aus den Schritten „Voraussetzungen“.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "fis:StartExperiment",
      "Resource": [
        "arn:aws:fis:*:*:experiment-template/your-experiment-template-id",
        "arn:aws:fis:*:*:experiment/*"
      ]
    }
  ]
}
```

Sie können den Scheduler so einschränken, dass er nur AWS FIS-Experimentvorlagen ausführt, die einen bestimmten Tag-Wert haben. Die folgende Richtlinie gewährt beispielsweise die StartExperiment Erlaubnis für alle AWS FIS-Experimente, schränkt den Scheduler jedoch darauf ein, nur mit Tags versehene Versuchsvorlagen auszuführen. Purpose=Schedule

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "fis:StartExperiment",
      "Resource": "arn:aws:fis:*:*:experiment/*"
    },
    {
      "Effect": "Allow",
      "Action": "fis:StartExperiment",
      "Resource": "arn:aws:fis:*:*:experiment-template/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Schedule"
        }
      }
    }
  ]
}
```

```
}  
  }  
}  
}
```

Wählen Sie Next: Tags (Weiter: Tags) aus.

6. Klicken Sie auf Weiter: Prüfen.
7. Geben Sie unter Richtlinie überprüfen einen Namen für Ihre Richtlinie **FIS_RecurringExperiment** ein und wählen Sie dann Richtlinie erstellen aus.
8. Fügen Sie unter Berechtigungen hinzufügen die neue FIS_RecurringExperiment Richtlinie zu Ihrer Rolle hinzu und wählen Sie dann Weiter aus.
9. Geben Sie der Rolle FIS_RecurringExperiment_role unter Name, Überprüfung und Erstellung einen Namen und wählen Sie dann Rolle erstellen aus.

Schritt 2: Erstellen Sie einen Amazon EventBridge Scheduler

Um einen Scheduler zu erstellen Amazon EventBridge

1. Öffnen Sie die EventBridge Amazon-Konsole unter <https://console.aws.amazon.com/events/>.
2. Wählen Sie im linken Navigationsbereich Zeitpläne aus.
3. Vergewissern Sie sich, dass Sie sich in derselben Vorlage AWS-Region wie in Ihrer AWS FIS-Experimentvorlage befinden.
4. Wählen Sie Zeitplan erstellen und geben Sie Folgendes ein:
 - Fügen Sie unter Name des Zeitplans Folgendes ein FIS_recurring_experiment_tutorial.
 - Wählen Sie unter Zeitplanmuster die Option Wiederkehrender Zeitplan aus.
 - Wählen Sie unter Zeitplantyp die Option Ratenbasierter Zeitplan aus.
 - Wählen Sie unter Preisausdruck die Option 5 Minuten aus.
 - Wählen Sie unter Flexibles Zeitfenster die Option Aus aus.
 - (Optional) Wählen Sie unter Zeitrahmen Ihre Zeitzone aus.
 - Wählen Sie Weiter aus.

5. Wählen Sie unter Ziel auswählen die Option Alle aus APIs, und suchen Sie dann nach AWS FIS.
6. Wählen Sie AWS FIS und dann aus. StartExperiment
7. Fügen Sie unter Eingabe die folgende JSON-Payload ein. Ersetzen Sie den *your-experiment-template-id* Wert durch die Vorlagen-ID Ihres Experiments. Das ClientToken ist eine eindeutige Kennung für den Scheduler. In diesem Tutorial verwenden wir ein vom Amazon EventBridge Scheduler zugelassenes Kontext-Schlüsselwort. Weitere Informationen finden Sie unter [Hinzufügen von Kontextattributen](#) im EventBridge Amazon-Benutzerhandbuch.

```
{
  "ClientToken": "<aws.scheduler.execution-id>",
  "ExperimentTemplateId": "your-experiment-template-id"
}
```

Wählen Sie Weiter aus.

8. (Optional) Unter Einstellungen können Sie die Wiederholungsrichtlinie, die Dead-Letter-Warteschlange (DLQ) und die Verschlüsselungseinstellungen festlegen. Alternativ können Sie die Standardwerte beibehalten.
9. Wählen Sie unter Berechtigungen die Option Bestehende Rolle verwenden aus, und suchen Sie dann nach FIS_RecurringExperiment_role.
10. Wählen Sie Weiter aus.
11. Überprüfen Sie unter Zeitplan überprüfen und erstellen die Details Ihres Terminplaners und wählen Sie dann Zeitplan erstellen aus.

Schritt 3: Verifizieren Sie Ihr Experiment

Um zu überprüfen, ob Ihr AWS FIS-Experiment planmäßig verlief

1. Öffnen Sie die AWS FIS-Konsole unter. <https://console.aws.amazon.com/fis/>
2. Wählen Sie im linken Navigationsbereich Experimente aus.
3. Fünf Minuten, nachdem Sie Ihren Zeitplan erstellt haben, sollten Sie sehen, dass Ihr Experiment läuft.

Schritt 4: Bereinigen

Um Ihren Amazon EventBridge Scheduler zu deaktivieren

1. Öffnen Sie die EventBridge Amazon-Konsole unter <https://console.aws.amazon.com/events/>.
2. Wählen Sie im linken Navigationsbereich Zeitpläne aus.
3. Wählen Sie Ihren neu erstellten Scheduler aus und klicken Sie dann auf Deaktivieren.

Arbeiten mit der AWS FIS Szenario-Bibliothek

Szenarien definieren Ereignisse oder Bedingungen, anhand derer Kunden die Resilienz ihrer Anwendungen testen können, z. B. die Unterbrechung der Rechenressourcen, auf denen die Anwendung ausgeführt wird. Die Szenarien werden von AWS erstellt und verwaltet und minimieren undifferenzierten Aufwand, indem sie Ihnen eine Gruppe von vordefinierten Zielen und Fehleraktionen (z. B. das Stoppen von 30% der Instances in einer Autoscaling-Gruppe) für häufig auftretende Anwendungsbeeinträchtigungen zur Verfügung stellen.

Szenarien werden über eine nur für Konsolen verfügbare Szenariobibliothek bereitgestellt und mithilfe einer Experimentvorlage ausgeführt. AWS FIS Um ein Experiment mit einem Szenario durchzuführen, wählen Sie das Szenario aus der Bibliothek aus, geben Parameter an, die Ihren Workload-Details entsprechen, und speichern es als Experimentvorlage in Ihrem Konto.

Topics

- [Ein Szenario ansehen](#)
- [Anhand eines Szenarios](#)
- [Exportieren eines Szenarios](#)
- [Referenz zu den Szenarien](#)

Ein Szenario ansehen

So zeigen Sie ein Szenario mit der Konsole an:

1. Öffnen Sie die AWS FIS Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich die Option Szenario-Bibliothek aus.
3. Um Informationen zu einem bestimmten Szenario anzuzeigen, wählen Sie die Szenariokarte aus, um ein geteiltes Fenster aufzurufen.
 - Auf der Registerkarte Beschreibung im geteilten Bereich unten auf der Seite können Sie eine kurze Beschreibung des Szenarios einsehen. Sie finden auch eine kurze Zusammenfassung der Voraussetzungen, die eine Zusammenfassung der erforderlichen Zielressourcen und aller Maßnahmen enthält, die Sie ergreifen müssen, um die Ressourcen für die Verwendung mit dem Szenario vorzubereiten. Schließlich finden Sie auch zusätzliche Informationen zu den Zielen und Aktionen im Szenario sowie zur voraussichtlichen Dauer, wenn das Experiment mit den Standardeinstellungen erfolgreich ausgeführt wird.

- Auf der Registerkarte Inhalt im geteilten Bereich unten auf der Seite können Sie eine Vorschau einer teilweise ausgefüllten Version der Experimentvorlage anzeigen, die anhand des Szenarios erstellt wird.
- Auf der Registerkarte „Details“ im geteilten Bereich unten auf der Seite finden Sie eine detaillierte Erklärung, wie das Szenario implementiert wird. Dies kann detaillierte Informationen darüber enthalten, wie einzelne Aspekte des Szenarios approximiert werden. Falls zutreffend, können Sie auch nachlesen, welche Metriken als Stoppbedingungen und zur Gewährleistung der Beobachtbarkeit verwendet werden sollten, damit Sie aus dem Experiment lernen können. Schließlich finden Sie Empfehlungen, wie Sie die resultierende Versuchsvorlage erweitern können.

Anhand eines Szenarios

Um ein Szenario mit der Konsole zu verwenden:

1. Öffnen Sie die AWS FIS Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich die Option Szenario-Bibliothek aus.
3. Um Informationen zu einem bestimmten Szenario anzuzeigen, wählen Sie die Szenariokarte aus, um ein geteiltes Fenster aufzurufen
4. Um das Szenario zu verwenden, wählen Sie die Szenariokarte aus und wählen Sie Vorlage mit Szenario erstellen aus.
5. Füllen Sie in der Ansicht „Experimentvorlage erstellen“ alle fehlenden Elemente aus.
 - a. In einigen Szenarien können Sie Parameter bearbeiten, die von mehreren Aktionen oder Zielen gemeinsam genutzt werden. Diese Funktion wird deaktiviert, sobald Sie Änderungen am Szenario vorgenommen haben, einschließlich Änderungen durch die Bearbeitung gemeinsam genutzter Parameter. Um diese Funktion zu verwenden, klicken Sie auf die Schaltfläche Gemeinsam genutzte Parameter bearbeiten. Bearbeiten Sie die Parameter im Modal und klicken Sie auf die Schaltfläche Speichern.
 - b. Bei einigen Experimentvorlagen fehlen möglicherweise Aktions- oder Zielparameter, die auf jeder Aktions- und Zielkarte hervorgehoben sind. Wählen Sie für jede Karte die Schaltfläche Bearbeiten, fügen Sie die fehlenden Informationen hinzu und klicken Sie auf der Karte auf die Schaltfläche Speichern.
 - c. Für alle Vorlagen ist eine Ausführungsrolle für den Dienstzugriff erforderlich. Sie können eine vorhandene Rolle auswählen oder eine neue Rolle für diese Experimentvorlage erstellen.

- d. Wir empfehlen, eine oder mehrere optionale Stoppbedingungen zu definieren, indem Sie einen vorhandenen CloudWatch AWS-Alarm auswählen. Weitere Informationen zu [Stoppbedingungen für AWS FIS](#). Wenn Sie noch keinen Alarm konfiguriert haben, können Sie den Anweisungen unter [Amazon CloudWatch Alarms verwenden](#) folgen und die Versuchsvorlage später aktualisieren.
 - e. Wir empfehlen, optionale Experimentprotokolle in CloudWatch Amazon-Protokollen oder in einem Amazon S3-Bucket zu aktivieren. Weitere Informationen zu [Protokollierung von Experimenten für AWS FIS](#). Wenn Sie noch keine entsprechenden Ressourcen konfiguriert haben, können Sie die Versuchsvorlage später aktualisieren.
6. Wählen Sie unter Experimentvorlage erstellen die Option Experimentvorlage erstellen aus.
 7. Wählen Sie in der Ansicht „Experimentvorlagen“ der AWS FIS Konsole die Option Experiment starten aus. Weitere Informationen zu [Verwaltung von AWS FIS-Experimentvorlagen](#).

Exportieren eines Szenarios

Szenarien sind nur für Konsolen verfügbar. Szenarien ähneln zwar Experimentvorlagen, sind aber keine vollständigen Experimentvorlagen und können nicht direkt importiert werden. AWS FIS Wenn Sie Szenarien als Teil Ihrer eigenen Automatisierung verwenden möchten, können Sie einen von zwei Pfaden verwenden:

1. Folgen Sie den Schritten unter [Anhand eines Szenarios](#), um eine gültige AWS FIS Experimentvorlage zu erstellen und diese Vorlage zu exportieren.
2. Folgen Sie den Schritten in [Ein Szenario ansehen](#) und in Schritt 3, kopieren und speichern Sie auf der Registerkarte Inhalt den Inhalt des Szenarios und fügen Sie dann fehlende Parameter manuell hinzu, um eine gültige Experimentvorlage zu erstellen.

Referenz zu den Szenarien

In der Szenariobibliothek enthaltene Szenarien sind so konzipiert, dass sie nach Möglichkeit [Tags](#) verwenden. Jedes Szenario beschreibt die erforderlichen Tags in den Abschnitten Voraussetzungen und Funktionsweise der Szenariobeschreibung. Sie können Ihre Ressourcen mit diesen vordefinierten Tags kennzeichnen oder Sie können mithilfe der gemeinsamen Parameterbearbeitung Ihre eigenen Tags festlegen (siehe [Anhand eines Szenarios](#)).

In dieser Referenz werden die gängigen Szenarien in der AWS FIS-Szenariobibliothek beschrieben. Sie können die unterstützten Szenarien auch mithilfe der AWS FIS-Konsole auflisten.

Weitere Informationen finden Sie unter [Arbeiten mit der AWS FIS Szenario-Bibliothek](#).

AWS FIS unterstützt die folgenden Amazon EC2-Szenarien. [Diese Szenarien zielen auf Instances ab, die Tags verwenden](#). Sie können Ihre eigenen Tags oder die im Szenario enthaltenen Standard-Tags verwenden. In einigen dieser Szenarien werden [SSM-Dokumente verwendet](#).

- **EC2-Stress: Instance-Ausfall** — Untersuchen Sie die Auswirkungen eines Instance-Ausfalls, indem Sie eine oder mehrere EC2-Instances stoppen.

Ziel-Instances in der aktuellen Region, denen ein bestimmtes Tag zugeordnet ist. In diesem Szenario stoppen wir diese Instances und starten sie am Ende der Aktionsdauer neu, standardmäßig 5 Minuten.

- **EC2-Stress: Disk** — Erkunden Sie die Auswirkungen einer erhöhten Festplattenauslastung auf Ihre EC2-basierte Anwendung.

In diesem Szenario zielen wir auf EC2-Instances in der aktuellen Region ab, denen ein bestimmtes Tag zugewiesen ist. In diesem Szenario können Sie für die Aktionsdauer eine zunehmende Festplattenauslastung festlegen, die in die Ziel-Instances von EC2 injiziert wird, standardmäßig 5 Minuten für jede Festplattenbelastungsaktion.

- **EC2-Stress: CPU** — Erkunden Sie die Auswirkungen einer erhöhten CPU-Auslastung auf Ihre EC2-basierte Anwendung.

In diesem Szenario zielen wir auf EC2-Instances in der aktuellen Region ab, denen ein bestimmtes Tag zugewiesen ist. In diesem Szenario können Sie eine zunehmende Menge an CPU-Stress, der auf bestimmte EC2-Instances injiziert wird, für die Aktionsdauer anpassen, standardmäßig 5 Minuten für jede CPU-Belastungsaktion.

- **EC2-Stress: Arbeitsspeicher** — Erkunden Sie die Auswirkungen einer erhöhten Speicherauslastung auf Ihre EC2-basierte Anwendung.

In diesem Szenario zielen wir auf EC2-Instances in der aktuellen Region ab, denen ein bestimmtes Tag zugewiesen ist. In diesem Szenario können Sie eine zunehmende Menge an Speicherbelastung, die auf gezielte EC2-Instances injiziert wird, für die Aktionsdauer anpassen, standardmäßig 5 Minuten für jede Speicherbelastungsaktion.

- **EC2-Stress: Netzwerklatenz** — Erkunden Sie die Auswirkungen einer erhöhten Netzwerklatenz auf Ihre EC2-basierte Anwendung.

In diesem Szenario zielen wir auf EC2-Instances in der aktuellen Region ab, denen ein bestimmtes Tag zugewiesen ist. In diesem Szenario können Sie eine zunehmende Menge an Netzwerklatenz,

die bestimmten EC2-Instances zugewiesen wird, für die Aktionsdauer anpassen, standardmäßig 5 Minuten für jede Latenzaktion.

AWS FIS unterstützt die folgenden Amazon EKS-Szenarien. Diese Szenarien zielen auf EKS-Pods ab, die Labels einer Kubernetes-Anwendung verwenden. Sie können Ihre eigenen Labels oder die im Szenario enthaltenen Standardlabels verwenden. Weitere Informationen zu EKS mit FIS finden Sie unter [EKS-Pod-Aktionen](#).

- **EKS-Stress: Pod Delete** — Erkunden Sie die Auswirkungen eines EKS-Pod-Ausfalls, indem Sie einen oder mehrere Pods löschen.

In diesem Szenario zielen wir auf Pods in der aktuellen Region ab, die einem Anwendungslabel zugeordnet sind. In diesem Szenario werden wir alle übereinstimmenden Pods beenden. Die Neuerstellung von Pods wird durch die Kubernetes-Konfiguration gesteuert.

- **EKS-Stress: CPU** — Erkunden Sie die Auswirkungen einer erhöhten CPU-Auslastung auf Ihre EKS-basierte Anwendung.

In diesem Szenario zielen wir auf Pods in der aktuellen Region ab, die einem Anwendungslabel zugeordnet sind. In diesem Szenario können Sie für die Aktionsdauer eine zunehmende Menge an CPU-Belastung, die auf gezielte EKS-Pods injiziert wird, anpassen, standardmäßig 5 Minuten für jede CPU-Stressaktion.

- **EKS-Stress: Disk** — Erkunden Sie die Auswirkungen einer erhöhten Festplattenauslastung auf Ihre EKS-basierte Anwendung.

In diesem Szenario zielen wir auf Pods in der aktuellen Region ab, die einem Anwendungslabel zugeordnet sind. In diesem Szenario können Sie für die Aktionsdauer eine zunehmende Menge an Festplattenbelastung anpassen, die auf gezielte EKS-Pods injiziert wird, standardmäßig 5 Minuten für jede CPU-Belastungsaktion.

- **EKS-Stress: Arbeitsspeicher** — Erkunden Sie die Auswirkungen einer erhöhten Speicherauslastung auf Ihre EKS-basierte Anwendung.

In diesem Szenario zielen wir auf Pods in der aktuellen Region ab, die einem Anwendungslabel zugeordnet sind. In diesem Szenario können Sie eine zunehmende Menge an Speicherstress, der auf gezielte EKS-Pods injiziert wird, für die Aktionsdauer anpassen, standardmäßig 5 Minuten für jede Speicherstressaktion.

- **EKS-Stress: Netzwerklatenz** — Erkunden Sie die Auswirkungen einer erhöhten Netzwerklatenz auf Ihre EKS-basierte Anwendung.

In diesem Szenario zielen wir auf Pods in der aktuellen Region ab, die einem Anwendungslabel zugeordnet sind. In diesem Szenario können Sie eine zunehmende Menge an Netzwerklatenz, die auf gezielte EKS-Pods übertragen wird, für die Aktionsdauer anpassen, standardmäßig 5 Minuten für jede Latenzaktion.

AWS FIS unterstützt die folgenden Szenarien für Single-AZ-, Multi-AZ- und Multi-Region-Anwendungen. Diese Szenarien zielen auf mehrere Ressourcentypen ab.

- **AZ Availability: Power Interruption-** Injizieren Sie die zu erwartenden Symptome einer vollständigen Stromunterbrechung in einer Availability Zone (AZ). Weitere Informationen zu [AZ Availability: Power Interruption](#).
- **AZ: Application Slowdown-** Erhöhen Sie die Latenz zwischen Ressourcen innerhalb einer einzigen Availability Zone (AZ), um eine Anwendung zu verlangsamen. Weitere Informationen zu [AZ: Application Slowdown](#).
- **Cross-AZ: Traffic Slowdown—** Fügen Sie Paketverlust ein, um den Verkehr zwischen Availability Zones () AZs zu unterbrechen und zu verlangsamen. Weitere Informationen zu [Cross-AZ: Traffic Slowdown](#).
- **Cross-Region: Connectivity-** Blockieren Sie den Netzwerkverkehr der Anwendung von der Versuchsregion zur Zielregion und unterbrechen Sie die regionsübergreifende Datenreplikation. Erfahren Sie mehr über die Verwendung von [Cross-Region: Connectivity](#).

AWS FIS unterstützt die folgenden Szenarien für Amazon EBS-Volumes. Diese Szenarien zielen mithilfe von Tags auf Volumes ab. Sie können Ihre eigenen Tags oder die im Szenario enthaltenen Standardtags verwenden. Die Zielvolumes müssen sich in derselben Availability Zone befinden. Weitere Informationen finden Sie unter [Fehlertests auf Amazon EBS](#).

- **EBS: Sustained Latency—** Untersuchen Sie die Auswirkungen anhaltender I/O Latenz auf Ihre Anwendung.

In diesem Szenario zielen wir auf Volumes in der aktuellen Availability Zone ab, denen ein bestimmtes Tag zugewiesen ist. In diesem Szenario wird eine konstante Latenz von 500 ms bei 50 Prozent der Lese- und 100 Prozent der Schreibvorgänge für ein Volume erreicht, wobei eine einzige Latenzaktion über einen Zeitraum von 15 Minuten verwendet wird. In diesem Szenario können Sie die Höhe der eingespeisten Latenz, den Prozentsatz der I/O injizierten Latenz und die Dauer der Aktion anpassen.

- **EBS: Increasing Latency**— Untersuchen Sie, wie sich die Erhöhung der I/O Latenz auf Ihre Anwendung auswirkt.

In diesem Szenario zielen wir auf Volumes in der aktuellen Availability Zone ab, denen ein bestimmtes Tag zugewiesen ist. In diesem Szenario wird bei 10 Prozent der Lese- und 25 Prozent der Schreibvorgänge für ein Volume mit fünf Latenzaktionen über einen Zeitraum von 15 Minuten eine zunehmende Latenz von 50 ms, 200 ms, 700 ms, 1 Sekunde und 15 Sekunden erreicht. In diesem Szenario können Sie für jede Latenzaktion die Menge der injizierten Latenz, den Prozentsatz der I/O injizierten Latenz und die Aktionsdauer anpassen.

- **EBS: Intermittent Latency**— Untersuchen Sie die Auswirkungen intermittierender I/O Latenzspitzen auf Ihre Anwendung.

In diesem Szenario zielen wir auf Volumes in der aktuellen Availability Zone ab, denen ein bestimmtes Tag zugewiesen ist. In diesem Szenario werden bei 0,1 Prozent der Lese- und I/O Schreibvorgänge für ein Volume drei starke, intermittierende Latenzspitzen von 30 Sekunden, 10 Sekunden und 20 Sekunden ausgelöst. Dabei werden drei Latenzaktionen verwendet, wobei zwischen jedem Anstieg über einen Zeitraum von 15 Minuten Wiederherstellungsintervalle liegen. In diesem Szenario können Sie für jede Latenzaktion den Umfang der injizierten Latenz, den Prozentsatz der I/O injizierten Latenz und die Aktionsdauer anpassen.

- **EBS: Decreasing Latency**— Untersuchen Sie die Auswirkungen abnehmender I/O Latenz auf Ihre Anwendung.

In diesem Szenario zielen wir auf Volumes in der aktuellen Availability Zone ab, denen ein bestimmtes Tag zugewiesen ist. In diesem Szenario wird bei 10 Prozent der Lese- und Schreibvorgänge für ein Volume eine verringerte Latenz von 20 Sekunden, 5 Sekunden, 900 ms, 300 ms und 40 ms erreicht, wobei fünf Latenzaktionen über einen Zeitraum von 15 Minuten verwendet werden. In diesem Szenario können Sie für jede Latenzaktion den Umfang der zugeführten Latenz, den Prozentsatz der I/O injizierten Latenz und die Aktionsdauer anpassen.

AZ Availability: Power Interruption

Sie können das AZ Availability: Power Interruption Szenario verwenden, um die erwarteten Symptome einer vollständigen Stromunterbrechung in einer Availability Zone (AZ) auszulösen.

Dieses Szenario kann verwendet werden, um zu demonstrieren, dass Multi-AZ-Anwendungen während einer einzigen, vollständigen AZ-Stromunterbrechung erwartungsgemäß funktionieren. Dazu gehören der Verlust von zonaler Rechenleistung (Amazon EC2, EKS und ECS), keine

erneute Skalierung der Rechenleistung in der AZ, Verlust der Subnetzkonnektivität, RDS-Failover, ElastiCache Failover, eingeschränkter Zugriff auf S3 Express One Zone-Verzeichnisbuckets und nicht reagierende EBS-Volumes. Standardmäßig werden Aktionen, für die keine Ziele gefunden wurden, übersprungen.

Aktionen

Zusammen führen die folgenden Aktionen zu vielen der zu erwartenden Symptome einer vollständigen Stromunterbrechung in einer einzigen AZ. AZ-Verfügbarkeit: Die Stromunterbrechung wirkt sich nur auf Dienste aus, bei denen davon ausgegangen wird, dass sie während einer einzigen AZ-Stromunterbrechung beeinträchtigt werden. Standardmäßig werden in diesem Szenario 30 Minuten lang Symptome einer Stromunterbrechung und dann weitere 30 Minuten lang Symptome auftreten, die während der Wiederherstellung auftreten können.

Stopp-Instanzen

Während einer AZ-Stromunterbrechung werden die EC2-Instances in der betroffenen AZ heruntergefahren. Nach der Wiederherstellung der Stromversorgung werden die Instances neu gestartet. AZ Availability: Power Interruption beinhaltet [aws:ec2:stop-instances, um alle Instances](#) in der betroffenen AZ für die Dauer der Unterbrechung zu stoppen. Nach Ablauf der Dauer werden die Instanzen neu gestartet. Das Stoppen von EC2-Instances, die von Amazon EKS verwaltet werden, führt dazu, dass abhängige EKS-Pods gelöscht werden. Das Stoppen von EC2-Instances, die von Amazon ECS verwaltet werden, führt dazu, dass abhängige ECS-Aufgaben gestoppt werden.

Diese Aktion zielt auf EC2-Instances ab, die in der betroffenen AZ ausgeführt werden. Standardmäßig zielt sie auf Instances ab, deren Tag `AzImpairmentPower` den Wert hat. `StopInstances` Sie können dieses Tag zu Ihren Instanzen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Instanzen gefunden werden, wird diese Aktion standardmäßig übersprungen.

Stop-ASG-Instanzen

Während einer AZ-Stromunterbrechung werden EC2-Instances, die von einer Auto Scaling Scaling-Gruppe in der betroffenen AZ verwaltet werden, heruntergefahren. Nach der Wiederherstellung der Stromversorgung werden die Instances neu gestartet. AZ Availability: Power Interruption beinhaltet [aws:ec2:stop-instances, um alle Instances](#), einschließlich der von Auto Scaling verwalteten, in der betroffenen AZ für die Dauer der Unterbrechung zu stoppen. Nach Ablauf der Dauer werden die Instances neu gestartet.

Diese Aktion zielt auf EC2-Instances ab, die in der betroffenen AZ ausgeführt werden. Standardmäßig zielt sie auf Instances ab, deren Tag `AzImpairmentPower` den Wert `IceAsg` hat. Sie können dieses Tag zu Ihren Instanzen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Instanzen gefunden werden, wird diese Aktion standardmäßig übersprungen.

Instanzstarts pausieren

Während einer AZ-Stromunterbrechung schlagen EC2-API-Aufrufe zur Bereitstellung von Kapazität in der AZ fehl. Insbesondere APIs wird Folgendes betroffen sein: `ec2:StartInstances` und `ec2:CreateFleet`, und `ec2:RunInstances`. AZ Availability: Power Interruption beinhaltet [aws:ec2: api-insufficient-instance-capacity -error](#), um zu verhindern, dass neue Instanzen in der betroffenen AZ bereitgestellt werden.

Diese Aktion zielt auf IAM-Rollen ab, die zur Bereitstellung von Instanzen verwendet werden. Diese müssen mit einem ARN gezielt werden. Wenn keine gültigen IAM-Rollen gefunden werden, wird diese Aktion standardmäßig übersprungen.

Unterbrechen Sie die ASG-Skalierung

Während einer AZ-Stromunterbrechung schlagen EC2-API-Aufrufe der Auto Scaling-Steuerebene zur Wiederherstellung verlorener Kapazität in der AZ fehl. Insbesondere APIs wird Folgendes betroffen sein: `ec2:StartInstances` und `ec2:CreateFleet`, und `ec2:RunInstances`. AZ Availability: Power Interruption beinhaltet [aws:ec2: asg-insufficient-instance-capacity -error](#), um zu verhindern, dass neue Instanzen in der betroffenen AZ bereitgestellt werden. Dadurch wird auch verhindert, dass Amazon EKS und Amazon ECS in der betroffenen AZ skaliert werden.

Diese Aktion zielt auf Auto Scaling Scaling-Gruppen ab. Standardmäßig zielt es auf Auto Scaling Scaling-Gruppen ab, deren Tag `AzImpairmentPower` den Wert `IceAsg` hat. Sie können dieses Tag zu Ihren Auto Scaling Scaling-Gruppen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Auto Scaling Scaling-Gruppen gefunden werden, wird diese Aktion standardmäßig übersprungen.

Netzwerkonnektivität unterbrechen

Während einer Stromunterbrechung ist das Netzwerk in der AZ nicht verfügbar. In diesem Fall kann es bei einigen AWS-Services bis zu ein paar Minuten dauern, bis das DNS aktualisiert wird, um widerzuspiegeln, dass private Endpunkte in der betroffenen AZ nicht verfügbar sind. Während dieser Zeit können DNS-Abfragen IP-Adressen zurückgeben, auf die nicht zugegriffen werden kann.

AZ Availability: Power Interruption beinhaltet [aws:network:disrupt-connectivity](#), um die gesamte [Netzwerk-konnektivität](#) für alle Subnetze in der betroffenen AZ für 2 Minuten zu blockieren. Dadurch werden Timeouts und DNS-Aktualisierungen für die meisten Anwendungen erzwungen. Wenn die Aktion nach 2 Minuten beendet wird, kann das DNS des regionalen Dienstes anschließend wiederhergestellt werden, solange die AZ weiterhin nicht verfügbar ist.

Diese Aktion zielt auf Subnetze ab. Standardmäßig zielt sie auf Cluster ab, deren Tag `AzImpairmentPower` den Wert `DisruptSubnet` hat. Sie können dieses Tag zu Ihren Subnetzen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Subnetze gefunden werden, wird diese Aktion standardmäßig übersprungen.

Failover RDS

Während einer AZ-Stromunterbrechung werden die RDS-Knoten in der betroffenen AZ heruntergefahren. Einzelne AZ-RDS-Knoten in der betroffenen AZ werden vollständig nicht verfügbar sein. Bei Multi-AZ-Clustern führt der Writer-Knoten ein Failover in eine nicht betroffene AZ durch und Leseknoten in der betroffenen AZ sind nicht verfügbar. Bei Multi-AZ-Clustern umfasst AZ Availability: Power Interruption dies [aws:rds](#): für ein Failover, wenn failover-db-cluster sich der Writer in der betroffenen AZ befindet.

Diese Aktion zielt auf RDS-Cluster ab. Standardmäßig zielt sie auf Cluster ab, deren Tag `AzImpairmentPower` den Wert `DisruptRds` hat. Sie können dieses Tag zu Ihren Clustern hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Cluster gefunden werden, wird diese Aktion standardmäßig übersprungen.

ElastiCache Replizierungsgruppe anhalten

Während einer AZ-Stromunterbrechung sind die ElastiCache Knoten in der AZ nicht verfügbar. AZ Availability: Power Interruption beinhaltet [aws:elasticache](#): replicationgroup-interrupt-az-power um ElastiCache Knoten in der betroffenen AZ zu beenden. Für die Dauer der Unterbrechung werden keine neuen Instances in der betroffenen AZ bereitgestellt, sodass die Kapazität der Replikationsgruppe weiterhin reduziert bleibt.

Diese Aktion zielt auf ElastiCache Replikationsgruppen ab. Standardmäßig zielt sie auf Replikationsgruppen ab, deren Tag `AzImpairmentPower` den Wert `ElasticacheImpact` hat. Sie können dieses Tag zu Ihren Replikationsgruppen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Replikationsgruppen gefunden werden, wird diese Aktion standardmäßig übersprungen. Beachten Sie, dass nur Replikationsgruppen mit Knoten in der betroffenen AZ als gültige Ziele betrachtet werden.

Starten Sie ARC Zonal Autoshift

Fünf Minuten nach Beginn der AZ-Stromunterbrechung verlagert die Wiederherstellungsaktion den Ressourcenverkehr für die verbleibenden 25 Minuten der Stromunterbrechung `aws:arc:start-zonal-autoshift` automatisch von der angegebenen AZ weg. Nach Ablauf dieser Dauer wird der Verkehr wieder auf die ursprüngliche AZ umgestellt. Beachten Sie, dass bei einer realen AZ-Stromunterbrechung die Beeinträchtigung erkannt und der Ressourcenverkehr verlagert AWS wird, sofern Autoshift aktiviert ist. Der Zeitpunkt dieser Schicht ist zwar unterschiedlich, es wird jedoch geschätzt, dass sie fünf Minuten nach Beginn der Beeinträchtigung eintritt.

Diese Aktion zielt auf Autoshift-fähige Ressourcen des Amazon Application Recovery Controller (ARC) ab. Standardmäßig zielt sie auf Ressourcen mit dem Tag key `AzImpairmentPower` und value `ab.RecoverAutoshiftResources`. Sie können dieses Tag zu Ihren Ressourcen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Möglicherweise möchten Sie beispielsweise ein anwendungsspezifisches Tag verwenden. Wenn keine gültigen Ressourcen gefunden werden, wird diese Aktion standardmäßig übersprungen.

EBS-I/O anhalten

Nach einer AZ-Stromunterbrechung kann es bei einem sehr kleinen Prozentsatz der Instances vorkommen, dass EBS-Volumes nicht mehr reagieren, sobald die Stromversorgung wiederhergestellt ist. AZ Availability: Power Interruption beinhaltet [aws:ebs:pause-io](#), sodass ein EBS-Volume nicht mehr reagiert.

Standardmäßig werden nur Volumes als Ziel ausgewählt, die so eingestellt sind, dass sie auch nach dem Beenden der Instance bestehen bleiben. Diese Aktion zielt auf Volumes ab, `AzImpairmentPower` deren Tag den Wert hat. `APIPauseVolume`. Sie können dieses Tag zu Ihren Bänden hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Volumes gefunden werden, wird diese Aktion standardmäßig übersprungen.

Unterbrechen Sie die Konnektivität zu S3 Express One Zone-Verzeichnis-Buckets

Während einer AZ-Stromunterbrechung ist kein Zugriff auf Daten möglich, die in den S3 Express One Zone-Verzeichnisbuckets in der AZ gespeichert sind. AZ-Verfügbarkeit: Die Stromunterbrechung umfasst [aws:network:disrupt-connectivity](#), um die Konnektivität zwischen Subnetzen und One Zone-Verzeichnis-Buckets in der betroffenen AZ für die Dauer des Experiments zu unterbrechen, was zu Timeouts bei API-Vorgängen auf der Datenebene auf zentraler Endpunkt-Datenebene führt. Verwenden Sie diese Aktion, um Störungen zu testen, wenn Rechenleistung und Speicher in einer AZ gemeinsam genutzt werden.

Diese Aktion zielt auf Subnetze ab. Standardmäßig zielt sie auf Subnetze ab, deren Tag `AzImpairmentPower` den Wert `DisruptSubnet` hat. Sie können dieses Tag zu Ihren Subnetzen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Subnetze gefunden werden, wird diese Aktion standardmäßig übersprungen.

Einschränkungen

- Dieses Szenario beinhaltet keine [Stoppbedingungen](#). Die richtigen Stoppbedingungen für Ihre Anwendung sollten der Versuchsvorlage hinzugefügt werden.
- In der Ziel-AZ werden Amazon EKS-Pods, die auf EC2 laufen, mit EC2-Worker-Knoten beendet und das Starten neuer EC2-Knoten wird blockiert. Amazon EKS-Pods, die auf AWS Fargate ausgeführt werden, werden jedoch nicht unterstützt.
- In der Ziel-AZ werden Amazon ECS-Aufgaben, die auf EC2 ausgeführt werden, mit EC2-Worker-Knoten beendet und das Starten neuer EC2-Knoten wird blockiert. Amazon ECS-Aufgaben, die auf AWS Fargate ausgeführt werden, werden jedoch nicht unterstützt.
- [Amazon RDS Multi-AZ](#) mit zwei lesbaren Standby-DB-Instances wird nicht unterstützt. In diesem Fall werden die Instances beendet, ein RDS-Failover durchgeführt und die Kapazität wird sofort wieder in der betroffenen AZ bereitgestellt. Der lesbare Standby-Modus in der betroffenen AZ bleibt verfügbar.

Voraussetzungen

- Fügen Sie der AWS FIS [Experimentrolle](#) die erforderliche Berechtigung hinzu.
- Ressourcen-Tags müssen auf Ressourcen angewendet werden, auf die das Experiment abzielen soll. Diese können Ihre eigene Tagging-Konvention oder die im Szenario definierten Standard-Tags verwenden.

Berechtigungen

ARC Zonal Autoshift verwendet eine mit dem IAM-Dienst verknüpfte Rolle, `AWSServiceRoleForZonalAutoshiftPracticeRun` um Zonal Shift in Ihrem Namen durchzuführen. Diese Rolle verwendet die von IAM verwaltete Richtlinie. [AWSZonalAutoshiftPracticeRunSLRPolicy](#) Sie müssen die Rolle nicht manuell erstellen. Wenn Sie eine Versuchsvorlage aus dem Szenario AZ Power Interruption im AWS-Managementkonsole, dem oder einem AWS SDK erstellen AWS CLI, erstellt ARC

die serviceverknüpfte Rolle für Sie. Weitere Informationen finden Sie unter [Verwenden der serviceverknüpften Rolle für zonales Autoshift](#) in ARC.

Die folgende Richtlinie gewährt AWS FIS die erforderlichen Berechtigungen, um ein Experiment mit dem AZ Availability: Power Interruption Szenario durchzuführen. Diese Richtlinie muss der [Rolle des Experiments](#) zugeordnet werden.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowFISExperimentLoggingActionsCloudwatch",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:PutResourcePolicy",
        "logs:DescribeResourcePolicies",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:*:*:network-acl/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkAcl",
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkAcl",
      "Resource": "arn:aws:ec2:*:*:network-acl/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkAclEntry",
      "ec2>DeleteNetworkAcl"
    ],
    "Resource": [
      "arn:aws:ec2::*:network-acl/*",
      "arn:aws:ec2::*:vpc/*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/managedByFIS": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:CreateNetworkAcl",
    "Resource": "arn:aws:ec2::*:vpc/*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVpcs",
      "ec2:DescribeManagedPrefixLists",
      "ec2:DescribeSubnets",
      "ec2:DescribeNetworkAcls"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "ec2:ReplaceNetworkAclAssociation",
    "Resource": [
      "arn:aws:ec2::*:subnet/*",
      "arn:aws:ec2::*:network-acl/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [

```



```

        "rds:FailoverDBCluster"
    ],
    "Resource": [
        "arn:aws:rds:*:*:cluster:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "rds:RebootDBInstance"
    ],
    "Resource": [
        "arn:aws:rds:*:*:db:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:DescribeReplicationGroups",
        "elasticache:InterruptClusterAzPower"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ]
},
{
    "Sid": "TargetResolutionByTags",
    "Effect": "Allow",
    "Action": [
        "tag:GetResources"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*"
},
{
    "Effect": "Allow",
    "Action": [

```

```

        "ec2:DescribeInstances"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "kms:CreateGrant"
    ],
    "Resource": [
        "arn:aws:kms:*:*:key/*"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": "ec2.*.amazonaws.com"
        },
        "Bool": {
            "kms:GrantIsForAWSResource": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeVolumes"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:PauseVolumeIO"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*"
},
{
    "Sid": "AllowInjectAPI",
    "Effect": "Allow",
    "Action": [
        "ec2:InjectApiError"
    ],
    "Resource": [
        "*"
    ]
},

```

```

        "Condition": {
            "ForAnyValue:StringEquals": {
                "ec2:FisActionId": [
                    "aws:ec2:api-insufficient-instance-capacity-error",
                    "aws:ec2:asg-insufficient-instance-capacity-error"
                ]
            }
        },
        {
            "Sid": "DescribeAsg",
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}

```

Inhalt des Szenarios

Der folgende Inhalt definiert das Szenario. Diese JSON-Datei kann gespeichert und verwendet werden, um mithilfe des [create-experiment-template](#) Befehls von der AWS-Befehlszeilenschnittstelle (AWS CLI) eine [Versuchsvorlage](#) zu erstellen. Die neueste Version des Szenarios finden Sie in der Szenariobibliothek in der FIS-Konsole.

```

{
  "targets": {
    "IAM-role": {
      "resourceType": "aws:iam:role",
      "resourceArns": [],
      "selectionMode": "ALL"
    },
    "EBS-Volumes": {
      "resourceType": "aws:ec2:ebs-volume",
      "resourceTags": {
        "AzImpairmentPower": "ApiPauseVolume"
      }
    },

```

```

    "selectionMode": "COUNT(1)",
    "parameters": {
      "availabilityZoneIdentifier": "us-east-1a"
    },
    "filters": [
      {
        "path": "Attachments.DeleteOnTermination",
        "values": [
          "false"
        ]
      }
    ],
  },
  "EC2-Instances": {
    "resourceType": "aws:ec2:instance",
    "resourceTags": {
      "AzImpairmentPower": "StopInstances"
    },
    "filters": [
      {
        "path": "State.Name",
        "values": [
          "running"
        ]
      },
      {
        "path": "Placement.AvailabilityZone",
        "values": [
          "us-east-1a"
        ]
      }
    ],
    "selectionMode": "ALL"
  },
  "ASG": {
    "resourceType": "aws:ec2:autoscaling-group",
    "resourceTags": {
      "AzImpairmentPower": "IceAsg"
    },
    "selectionMode": "ALL"
  },
  "ASG-EC2-Instances": {
    "resourceType": "aws:ec2:instance",
    "resourceTags": {

```

```

        "AzImpairmentPower": "IceAsg"
    },
    "filters": [
        {
            "path": "State.Name",
            "values": [
                "running"
            ]
        },
        {
            "path": "Placement.AvailabilityZone",
            "values": [
                "us-east-1a"
            ]
        }
    ],
    "selectionMode": "ALL"
},
"Subnet": {
    "resourceType": "aws:ec2:subnet",
    "resourceTags": {
        "AzImpairmentPower": "DisruptSubnet"
    },
    "filters": [
        {
            "path": "AvailabilityZone",
            "values": [
                "us-east-1a"
            ]
        }
    ],
    "selectionMode": "ALL",
    "parameters": {}
},
"RDS-Cluster": {
    "resourceType": "aws:rds:cluster",
    "resourceTags": {
        "AzImpairmentPower": "DisruptRds"
    },
    "selectionMode": "ALL",
    "parameters": {
        "writerAvailabilityZoneIdentifiers": "us-east-1a"
    }
},

```

```

    "ElastiCache-Cluster": {
      "resourceType": "aws:elasticache:replicationgroup",
      "resourceTags": {
        "AzImpairmentPower": "DisruptElasticache"
      },
      "selectionMode": "ALL",
      "parameters": {
        "availabilityZoneIdentifier": "us-east-1a"
      }
    },
    "actions": {
      "Pause-Instance-Launches": {
        "actionId": "aws:ec2:api-insufficient-instance-capacity-error",
        "parameters": {
          "availabilityZoneIdentifiers": "us-east-1a",
          "duration": "PT30M",
          "percentage": "100"
        },
        "targets": {
          "Roles": "IAM-role"
        }
      },
      "Pause-EBS-IO": {
        "actionId": "aws:ebs:pause-volume-io",
        "parameters": {
          "duration": "PT30M"
        },
        "targets": {
          "Volumes": "EBS-Volumes"
        },
        "startAfter": [
          "Stop-Instances",
          "Stop-ASG-Instances"
        ]
      },
      "Stop-Instances": {
        "actionId": "aws:ec2:stop-instances",
        "parameters": {
          "completeIfInstancesTerminated": "true",
          "startInstancesAfterDuration": "PT30M"
        },
        "targets": {
          "Instances": "EC2-Instances"
        }
      }
    }
  }

```

```

    }
  },
  "Pause-ASG-Scaling": {
    "actionId": "aws:ec2:asg-insufficient-instance-capacity-error",
    "parameters": {
      "availabilityZoneIdentifiers": "us-east-1a",
      "duration": "PT30M",
      "percentage": "100"
    },
    "targets": {
      "AutoScalingGroups": "ASG"
    }
  },
  "Stop-ASG-Instances": {
    "actionId": "aws:ec2:stop-instances",
    "parameters": {
      "completeIfInstancesTerminated": "true",
      "startInstancesAfterDuration": "PT30M"
    },
    "targets": {
      "Instances": "ASG-EC2-Instances"
    }
  },
  "Pause-network-connectivity": {
    "actionId": "aws:network:disrupt-connectivity",
    "parameters": {
      "duration": "PT2M",
      "scope": "all"
    },
    "targets": {
      "Subnets": "Subnet"
    }
  },
  "Failover-RDS": {
    "actionId": "aws:rds:failover-db-cluster",
    "parameters": {},
    "targets": {
      "Clusters": "RDS-Cluster"
    }
  },
  "Pause-ElastiCache": {
    "actionId": "aws:elasticache:replicationgroup-interrupt-az-power",
    "parameters": {
      "duration": "PT30M"
    }
  }
}

```

```
    },
    "targets": {
      "ReplicationGroups": "ElastiCache-Cluster"
    }
  },
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": ""
    }
  ],
  "roleArn": "",
  "tags": {
    "Name": "AZ Impairment: Power Interruption"
  },
  "logConfiguration": {
    "logSchemaVersion": 2
  },
  "experimentOptions": {
    "accountTargeting": "single-account",
    "emptyTargetResolutionMode": "skip"
  },
  "description": "Affect multiple resource types in a single AZ, targeting by tags and explicit ARNs, to approximate power interruption in one AZ."
}
```

AZ: Application Slowdown

Sie können das Szenario AZ: Application Slowdown verwenden, um zusätzliche Latenz zwischen Ressourcen innerhalb einer einzigen Availability Zone (AZ) einzuführen. Diese Latenz führt zu vielen Symptomen einer Anwendungsverlangsamung, einer teilweisen Unterbrechung, die manchmal auch als grauer Ausfall bezeichnet wird. Sie erhöht die Latenz der Netzwerkflüsse zwischen den Zielressourcen. Netzwerkflüsse stellen den Verkehr zwischen Rechenressourcen dar — Datenpakete, die Anfragen, Antworten und andere Kommunikationswege zwischen Ihren Servern, Containern und Diensten übertragen. Das Szenario kann dabei helfen, Beobachtbarkeitseinstellungen zu validieren, Alarmschwellenwerte zu optimieren, die Empfindlichkeit von Anwendungen gegenüber Verlangsamungen zu ermitteln und wichtige betriebliche Entscheidungen wie die Evakuierung von AZ zu treffen.

Standardmäßig erhöht das Szenario für eine Dauer von 30 Minuten eine Latenz von 200 ms auf 100% der Netzwerkflüsse zwischen Zielressourcen innerhalb der ausgewählten AZ. Sie können das Dialogfeld Gemeinsame Parameter bearbeiten in der AWS FIS-Konsole verwenden, um die folgenden Parameter auf Szenarioebene anzupassen, die dann für die zugrunde liegenden Aktionen gelten:

- **Availability Zone** — Sie können die AZ auswählen, die im Szenario beeinträchtigt werden soll.
- **Latenz in Millisekunden (ms)** — passen Sie diese an die Empfindlichkeit und die Bedürfnisse Ihrer Anwendung an. Sie können die Latenz für empfindlichere Anwendungen niedriger oder höher einstellen, um beispielsweise die Behandlung von Timeouts zu testen. Erwägen Sie, ein Vielfaches Ihrer aktuellen Anwendungslatenz als Ausgangswert zu verwenden.
- **Prozentsatz der Datenflüsse** — reduzieren Sie, um einen Teil des Datenverkehrs zu beeinträchtigen. Für noch subtilere Tests können Sie beispielsweise eine Latenz von 200 ms hinzufügen, was 25% der Netzwerkflüsse betrifft.
- **Dauer** — legen Sie fest, wie lange das Experiment läuft. Sie können die Dauer für schnellere Tests verkürzen oder längere Dauertests ausführen. Legen Sie die Dauer beispielsweise auf 2 Stunden fest, um Ihre Wiederherstellungsmechanismen unter eingeschränkten Bedingungen zu testen.
- **Ressourcenausrichtung** — Sie können Zielressourcen für das Gesamtszenario mithilfe von Tags (für EC2-Instances oder ECS-Aufgaben auf EC2 oder Fargate) oder Labels (für EKS-Pods auf EC2) definieren. Sie können Ihre eigenen Tags und Labels angeben oder die im Szenario bereitgestellten Standardwerte verwenden. Wenn Sie keine Tags oder Labels verwenden möchten, können Sie die Aktion so bearbeiten, dass sie auf Ressourcen abzielt, indem Sie andere Parameter angeben.
- **Anpassung** — Wenn Sie nicht auf EC2- oder ECS-Ressourcen abzielen möchten, können Sie die Aktionen mit Standard-Tags belassen. Im Experiment werden keine Ressourcen gefunden, auf die Sie abzielen könnten, und die Aktion wird übersprungen. Wenn Sie jedoch nicht auf EKS-Ressourcen abzielen möchten, sollten Sie die EKS-Aktion und das EKS-Ziel vollständig aus dem Szenario entfernen, da dafür eine EKS-Cluster-ID angegeben werden muss. Für eine noch detailliertere Anpassung können Sie einzelne Aktionen direkt in der Experimentvorlage ändern.

Aktionen

Zusammengenommen tragen die folgenden Aktionen dazu bei, viele der Symptome einer Anwendungsverlangsamung in einer einzigen AZ hervorzurufen, indem sie zu zusätzlicher Latenz bei den Netzwerkabläufen führen, die sich dann in der gesamten Anwendung ausbreiten. Diese Aktionen werden parallel ausgeführt, wobei jeweils standardmäßig eine Latenz von 200 ms für 30 Minuten

hinzugefügt wird. Nach diesem Zeitraum kehrt die Latenz auf ein normales Niveau zurück. Für die Ausführung des Szenarios ist mindestens einer der folgenden Ressourcentypen erforderlich: EC2-Instance, ECS-Task oder EKS-Pod.

ECS-Netzwerklatenz

AZ: Application Slowdown beinhaltet [aws:ecs](#), task-network-latency um Latenz für ECS-Aufgaben einzuführen. Die Aktion zielt auf Aufgaben in der ausgewählten AZ ab. Standardmäßig zielt sie auf Aufgaben ab, [deren Tag](#) AZApplicationSlowdown den Wert hatLatencyForECS. Sie können das Standard-Tag durch Ihr eigenes ersetzen oder das Szenario-Tag zu Ihren Aufgaben hinzufügen. Wenn keine gültigen Aufgaben gefunden werden, wird diese Aktion übersprungen. Bevor Sie ein Experiment auf ECS durchführen, sollten Sie die [Einrichtungsschritte für ECS-Aufgabenaktionen](#) befolgen.

EKS-Netzwerklatenz

AZ: Application Slowdown beinhaltet [aws:eks](#), pod-network-latency um Latenz für EKS-Pods einzuführen. Die Aktion zielt auf Pods in der ausgewählten AZ ab. Standardmäßig zielt sie auf Pods innerhalb eines Clusters ab, die Labels im Format key=value haben. Das bereitgestellte Standardlabel ist. AZApplicationSlowdown=LatencyForEKS Sie können das Standardlabel durch Ihr eigenes ersetzen oder dieses Label zu Ihren Pods hinzufügen. Wenn keine gültigen Pods gefunden werden, wird diese Aktion übersprungen. Bevor Sie ein Experiment mit EKS durchführen, sollten Sie die [Einrichtungsschritte für EKS-Pod-Aktionen](#) befolgen.

EC2-Netzwerklatenz

[AZ: Application Slowdown verwendet die Aktion aws:ssm:send-command, um das Dokument - Network-Latency-SOURCES auszuführen, um Latenz für EC2-Instances einzuführenAWSFIS-Run.](#) Die Aktion zielt auf Instances in der ausgewählten AZ ab. Standardmäßig zielt sie auf Instances ab, AZApplicationSlowdown deren [Tag](#) den Wert hatLatencyForEC2. Sie können das Standard-Tag durch Ihr eigenes ersetzen oder dieses Tag zu Ihren Instances hinzufügen. Wenn keine gültigen Instanzen gefunden werden, wird diese Aktion übersprungen. Bevor Sie mit SSM ein Experiment auf EC2 ausführen, sollten Sie [den AWS Systems Manager Manager-Agenten konfigurieren](#).

Einschränkungen

- Dieses Szenario beinhaltet keine [Stoppbedingungen](#). Die richtigen Stoppbedingungen für Ihre Anwendung sollten der Versuchsvorlage hinzugefügt werden.

Voraussetzungen

- Fügen Sie der AWS [FIS-Experimentrolle](#) die erforderlichen Berechtigungen hinzu.
- Sie müssen innerhalb der ausgewählten AZ auf eine oder mehrere Ressourcen aus einem der folgenden drei Typen abzielen: EC2-Instances, ECS-Aufgaben oder EKS-Pods.
- Alle Ziele des Szenarios müssen sich in derselben VPC befinden.

Berechtigungen

Um dieses Szenario auszuführen, benötigen Sie eine IAM-Rolle mit einer Vertrauensrichtlinie, die es FIS ermöglicht, die Rolle und die verwalteten Richtlinien für die Ressourcentypen zu übernehmen, auf die Sie im Experiment abzielen: EC2, ECS und EKS. Wenn Sie eine Versuchsvorlage aus dem Szenario AZ: Application Slowdown erstellen, erstellt FIS die Rolle für Sie mit der Vertrauensrichtlinie und den folgenden von AWS verwalteten Richtlinien:

- [AWSFaultInjectionSimulatorEC2Access](#)
- [AWSFaultInjectionSimulatorECSAccess](#)
- [AWSFaultInjectionSimulatorEKSAccess](#)

Wenn Sie eine bestehende [IAM-Rolle](#) verwenden, um das AZ: Application Slowdown-Szenario auszuführen, können Sie die folgende Richtlinie anhängen, um AWS FIS die erforderlichen Berechtigungen zu gewähren:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DescribeTasks",
      "Effect": "Allow",
      "Action": "ecs:DescribeTasks",
      "Resource": "*"
    },
    {
      "Sid": "DescribeContainerInstances",
      "Effect": "Allow",
      "Action": "ecs:DescribeContainerInstances",
      "Resource": "arn:aws:ecs:*:*:container-instance/*/*"
    },
    {
```

```

        "Sid": "DescribeInstances",
        "Effect": "Allow",
        "Action": "ec2:DescribeInstances",
        "Resource": "*"
    },
    {
        "Sid": "DescribeSubnets",
        "Effect": "Allow",
        "Action": "ec2:DescribeSubnets",
        "Resource": "*"
    },
    {
        "Sid": "DescribeCluster",
        "Effect": "Allow",
        "Action": "eks:DescribeCluster",
        "Resource": "arn:aws:eks:*:*:cluster/*"
    },
    {
        "Sid": "TargetResolutionByTags",
        "Effect": "Allow",
        "Action": "tag:GetResources",
        "Resource": "*"
    },
    {
        "Sid": "SendCommand",
        "Effect": "Allow",
        "Action": [
            "ssm:SendCommand"
        ],
        "Resource": [
            "arn:aws:ec2:*:*:instance/*",
            "arn:aws:ssm:*:*:managed-instance/*",
            "arn:aws:ssm:*:*:document/*"
        ]
    },
    {
        "Sid": "ListCommands",
        "Effect": "Allow",
        "Action": [
            "ssm:ListCommands"
        ],
        "Resource": "*"
    },
    {

```

```

        "Sid": "CancelCommand",
        "Effect": "Allow",
        "Action": [
            "ssm:CancelCommand"
        ],
        "Resource": "*"
    }
}

```

Inhalt des Szenarios

Der folgende Inhalt definiert das Szenario. Diese JSON-Datei kann gespeichert und verwendet werden, um mithilfe des [create-experiment-template](#) Befehls von der AWS-Befehlszeilenschnittstelle (AWS CLI) eine [Versuchsvorlage](#) zu erstellen. Die neueste Version des Szenarios finden Sie in der Szenario-Bibliothek in der FIS-Konsole und wechseln Sie zur Registerkarte Inhalt.

```

{
  "tags": {
    "Name": "AZ: Application Slowdown"
  },
  "description": "Add latency between resources within a single AZ.",
  "actions": {
    "LatencyForEKS": {
      "actionId": "aws:eks:pod-network-latency",
      "parameters": {
        "delayMilliseconds": "200",
        "duration": "PT30M",
        "flowsPercent": "100",
        "interface": "DEFAULT",
        "kubernetesServiceAccount": "fis-service-account",
        "sources": "us-east-1a"
      },
      "targets": {
        "Pods": "TargetsForEKS"
      }
    },
    "LatencyForEC2": {
      "actionId": "aws:ssm:send-command",
      "parameters": {
        "duration": "PT30M",
        "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-Network-
Latency-Sources",

```

```

        "documentParameters": "{\\\"DelayMilliseconds\\\":\\\"200\\\",\\\"Sources\\\":\\\"us-east-1a\\\",\\\"Interface\\\":\\\"DEFAULT\\\",\\\"TrafficType\\\":\\\"egress\\\",\\\"DurationSeconds\\\":\\\"1800\\\",\\\"FlowsPercent\\\":\\\"100\\\",\\\"InstallDependencies\\\":\\\"True\\\"}"
    },
    "targets": {
        "Instances": "TargetsForEC2"
    }
},
"LatencyForECS": {
    "actionId": "aws:ecs:task-network-latency",
    "parameters": {
        "delayMilliseconds": "200",
        "duration": "PT30M",
        "flowsPercent": "100",
        "installDependencies": "true",
        "sources": "us-east-1a",
        "useEcsFaultInjectionEndpoints": "true"
    },
    "targets": {
        "Tasks": "TargetsForECS"
    },
    "startAfter": []
}
},
"targets": {
    "TargetsForEKS": {
        "parameters": {
            "availabilityZoneIdentifier": "us-east-1a",
            "clusterIdentifier": "",
            "namespace": "default",
            "selectorType": "labelSelector",
            "selectorValue": "AZApplicationSlowdown=LatencyForEKS"
        },
        "resourceType": "aws:eks:pod",
        "selectionMode": "ALL"
    },
    "TargetsForEC2": {
        "filters": [
            {
                "path": "Placement.AvailabilityZone",
                "values": [
                    "us-east-1a"
                ]
            }
        ]
    }
}

```

```

    ],
    "resourceTags": {
        "AZApplicationSlowdown": "LatencyForEC2"
    },
    "resourceType": "aws:ec2:instance",
    "selectionMode": "ALL"
},
"TargetsForECS": {
    "filters": [
        {
            "path": "AvailabilityZone",
            "values": [
                "us-east-1a"
            ]
        }
    ],
    "resourceTags": {
        "AZApplicationSlowdown": "LatencyForECS"
    },
    "resourceType": "aws:ecs:task",
    "selectionMode": "ALL"
}
},
"experimentOptions": {
    "accountTargeting": "single-account",
    "emptyTargetResolutionMode": "skip"
},
"stopConditions": [
    {
        "source": "none"
    }
]
}

```

Cross-AZ: Traffic Slowdown

Sie können das Szenario Cross-AZ: Traffic Slowdown verwenden, um den Datenverkehr zwischen Availability Zones () durch Paketverlust zu unterbrechen und zu verlangsamen. AZs Der Paketverlust beeinträchtigt die AZ-übergreifende Kommunikation, was zu einer teilweisen Unterbrechung führt, die manchmal auch als grauer Ausfall bezeichnet wird. Er führt zu Paketverlusten bei Netzwerkströmen zwischen Zielressourcen. Netzwerkflüsse stellen den Verkehr zwischen Rechenressourcen dar — Datenpakete, die Anfragen, Antworten und andere Kommunikationswege

zwischen Ihren Servern, Containern und Diensten übertragen. Das Szenario kann dabei helfen, Beobachtbarkeitseinstellungen zu validieren, Alarmschwellenwerte zu optimieren, Anwendungsempfindlichkeiten und Abhängigkeiten bei der AZ-übergreifenden Kommunikation zu ermitteln und wichtige betriebliche Entscheidungen wie die Evakuierung von AZ zu treffen.

Standardmäßig führt das Szenario für eine Dauer von 30 Minuten zu einem Paketverlust von 15% bei 100% der ausgehenden Netzwerkflüsse für Zielressourcen aus der ausgewählten AZ. Sie können das Dialogfeld Gemeinsame Parameter bearbeiten in der AWS FIS-Konsole verwenden, um die folgenden Parameter auf Szenarioebene anzupassen, die dann für die zugrunde liegenden Aktionen gelten:

- **Availability Zone** — Sie können die AZ auswählen, die beeinträchtigt werden soll, und Paketverluste werden von dieser AZ zur anderen AZs innerhalb der Region übertragen.
- **Paketverlust** — Passen Sie den Paketverlust für Tests auf subtile Unterbrechungen niedriger an, z. B. 5% oder höher, um schwerwiegende Kommunikationsbeeinträchtigungen und Wiederherstellungsmechanismen zu testen, z. B. 50% oder sogar 100%, um die gesamte Konnektivität zu beeinträchtigen.
- **Prozentsatz der Datenflüsse** — reduzieren Sie, um einen Teil des Datenverkehrs zu beeinträchtigen. Für noch subtilere Tests können Sie beispielsweise einen Paketverlust von 15% einbauen, der 25% der Netzwerkflüsse betrifft.
- **Dauer** — legen Sie fest, wie lange das Experiment läuft. Sie können die Dauer für schnellere Tests verkürzen oder längere Dauertests ausführen. Legen Sie die Dauer beispielsweise auf 2 Stunden fest, um die Wiederherstellungsmechanismen unter beeinträchtigten Bedingungen zu testen.
- **Ressourcenausrichtung** — Sie können Zielressourcen für das Gesamtszenario mithilfe von Tags (für EC2-Instances oder ECS-Aufgaben auf EC2 oder Fargate) oder Labels (für EKS-Pods auf EC2) definieren. Sie können Ihre eigenen Tags und Labels angeben oder die im Szenario bereitgestellten Standardwerte verwenden. Wenn Sie keine Tags oder Labels verwenden möchten, können Sie die Aktion so bearbeiten, dass sie auf Ressourcen abzielt, indem Sie andere Parameter angeben.
- **Anpassung** — Wenn Sie nicht auf EC2- oder ECS-Ressourcen abzielen möchten, können Sie die Aktionen mit Standard-Tags belassen. Im Experiment werden keine Ressourcen gefunden, auf die Sie abzielen könnten, und die Aktion wird übersprungen. Wenn Sie jedoch nicht auf EKS-Ressourcen abzielen möchten, sollten Sie die EKS-Aktion und das EKS-Ziel vollständig aus dem Szenario entfernen, da dafür eine EKS-Cluster-ID angegeben werden muss. Für eine noch detailliertere Anpassung können Sie einzelne Aktionen direkt in der Experimentvorlage ändern.

Aktionen

Zusammen tragen die folgenden Aktionen dazu bei, die Symptome einer Verlangsamung des Datenverkehrs zwischen AZs zu erzeugen, indem sie zu Paketverlusten bei der ausgehenden Kommunikation von der Ziel-AZ zu anderen AZs in der Region auf Netzwerkebene führen. Diese Aktionen werden parallel ausgeführt, wobei jede Aktion standardmäßig einen Paketverlust von 15% für 30 Minuten verursacht. Nach diesem Zeitraum kehrt die Kommunikation zur Normalität zurück. Für die Ausführung des Szenarios ist mindestens einer der folgenden Ressourcentypen in der ausgewählten AZ erforderlich: EC2-Instance, ECS-Task oder EKS-Pod.

Verlust von ECS-Netzwerkpaketen

Cross-AZ: Traffic Slowdown beinhaltet [aws:ecs:](#), um Paketverlust bei ECS-Aufgaben task-network-packet-loss einzuschleusen. Die Aktion zielt auf Aufgaben in der ausgewählten AZ ab und beeinträchtigt deren ausgehende Kommunikation mit allen anderen in der Region. AZs Sie können den Umfang der Auswirkungen weiter anpassen, indem Sie die Aktion bearbeiten und dem Feld etwas hinzufügen oder AZs daraus entfernen. Sources Standardmäßig zielt es auf Aufgaben ab, [deren Tag](#) CrossAZTrafficSlowdown den Wert hatPacketLossForECS. Sie können das Standard-Tag durch Ihr eigenes ersetzen oder das Szenario-Tag zu Ihren Aufgaben hinzufügen. Wenn keine gültigen Aufgaben gefunden werden, wird diese Aktion übersprungen. Bevor Sie ein Experiment auf ECS durchführen, sollten Sie die [Einrichtungsschritte für ECS-Aufgabenaktionen](#) befolgen.

Verlust von EKS-Netzwerkpaketen

Cross-AZ: Traffic Slowdown beinhaltet [aws:eks:](#), um Paketverlust für EKS-Pods pod-network-packet-loss zu verhindern. Die Aktion zielt auf Pods in der ausgewählten AZ ab und beeinträchtigt deren ausgehende Kommunikation mit allen anderen Pods in der Region. AZs Sie können den Umfang der Auswirkungen weiter anpassen, indem Sie die Aktion bearbeiten und dem Feld etwas hinzufügen oder AZs daraus entfernen. Sources Standardmäßig zielt es auf Pods innerhalb eines Clusters ab, die Labels im Format key=value haben. Das bereitgestellte Standardlabel ist. CrossAZTraffic=PacketLossForEKS Sie können das Standardlabel durch Ihr eigenes ersetzen oder dieses Label zu Ihren Pods hinzufügen. Wenn keine gültigen Pods gefunden werden, wird diese Aktion übersprungen. Bevor Sie ein Experiment mit EKS durchführen, sollten Sie die [Einrichtungsschritte für EKS-Pod-Aktionen](#) befolgen.

Verlust von EC2-Netzwerkpaketen

Cross-AZ: Traffic Slowdown verwendet die Aktion [aws:ssm:send-command](#), um das Dokument [AWSFIS-Run-Network-Packet-Loss-Sources](#) auszuführen, um Paketverlust für EC2-Instances zu

[verursachen, und beeinträchtigt deren](#) ausgehende Kommunikation mit allen anderen in der Region. AZs Sie können den Umfang der Auswirkung weiter anpassen, indem Sie die Aktion bearbeiten und dem Feld etwas hinzufügen oder daraus entfernen. AZs Sources Die Aktion zielt auf Instanzen in der ausgewählten AZ ab. Standardmäßig zielt sie auf Instances ab, CrossAZTrafficSlowdown deren [Tag](#) den Wert hatPacketLossForEC2. Sie können das Standard-Tag durch Ihr eigenes ersetzen oder dieses Tag zu Ihren Instances hinzufügen. Wenn keine gültigen Instanzen gefunden werden, wird diese Aktion übersprungen. Bevor Sie mit SSM ein Experiment auf EC2 ausführen, sollten Sie [den AWS Systems Manager Manager-Agenten konfigurieren](#).

Einschränkungen

- Dieses Szenario beinhaltet keine [Stoppbedingungen](#). Die richtigen Stoppbedingungen für Ihre Anwendung sollten der Versuchsvorlage hinzugefügt werden.

Voraussetzungen

- Fügen Sie der AWS [FIS-Experimentrolle](#) die erforderlichen Berechtigungen hinzu.
- Sie müssen innerhalb der ausgewählten AZ auf eine oder mehrere Ressourcen aus einem der folgenden drei Typen abzielen: EC2-Instances, ECS-Aufgaben oder EKS-Pods.
- Alle Ziele des Szenarios müssen sich in derselben VPC befinden.

Berechtigungen

Um dieses Szenario auszuführen, benötigen Sie eine IAM-Rolle mit einer Vertrauensrichtlinie, die es FIS ermöglicht, die Rolle und die verwalteten Richtlinien für die Ressourcentypen zu übernehmen, auf die Sie im Experiment abzielen: EC2, ECS und EKS. Wenn Sie eine Versuchsvorlage aus dem Szenario Cross-AZ: Traffic Slowdown erstellen, erstellt FIS die Rolle für Sie mit der Vertrauensrichtlinie und den folgenden von AWS verwalteten Richtlinien:

- [AWSFaultInjectionSimulatorEC2Access](#)
- [AWSFaultInjectionSimulatorECSAccess](#)
- [AWSFaultInjectionSimulatorEKSAccess](#)

Wenn Sie eine bestehende [IAM-Rolle](#) verwenden, um das Szenario Cross-AZ: Traffic Slowdown auszuführen, können Sie die folgende Richtlinie anhängen, um AWS FIS die erforderlichen Berechtigungen zu gewähren:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DescribeTasks",
      "Effect": "Allow",
      "Action": "ecs:DescribeTasks",
      "Resource": "*"
    },
    {
      "Sid": "DescribeContainerInstances",
      "Effect": "Allow",
      "Action": "ecs:DescribeContainerInstances",
      "Resource": "arn:aws:ecs:*:*:container-instance/*/*"
    },
    {
      "Sid": "DescribeInstances",
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    },
    {
      "Sid": "DescribeSubnets",
      "Effect": "Allow",
      "Action": "ec2:DescribeSubnets",
      "Resource": "*"
    },
    {
      "Sid": "DescribeCluster",
      "Effect": "Allow",
      "Action": "eks:DescribeCluster",
      "Resource": "arn:aws:eks:*:*:cluster/*"
    },
    {
      "Sid": "TargetResolutionByTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    },
    {
      "Sid": "SendCommand",
      "Effect": "Allow",
      "Action": [

```

```

        "ssm:SendCommand"
    ],
    "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:managed-instance/*",
        "arn:aws:ssm:*:*:document/*"
    ]
},
{
    "Sid": "ListCommands",
    "Effect": "Allow",
    "Action": [
        "ssm:ListCommands"
    ],
    "Resource": "*"
},
{
    "Sid": "CancelCommand",
    "Effect": "Allow",
    "Action": [
        "ssm:CancelCommand"
    ],
    "Resource": "*"
}
]
}

```

Inhalt des Szenarios

Der folgende Inhalt definiert das Szenario. Diese JSON-Datei kann gespeichert und verwendet werden, um mithilfe des [create-experiment-template](#) Befehls von der AWS-Befehlszeilenschnittstelle (AWS CLI) eine [Versuchsvorlage](#) zu erstellen. Die neueste Version des Szenarios finden Sie in der Szenario-Bibliothek in der FIS-Konsole und wechseln Sie zur Registerkarte Inhalt.

```

{
  "tags": {
    "Name": "Cross-AZ: Traffic Slowdown"
  },
  "description": "Inject packet loss to disrupt and slow down traffic between AZs.",
  "actions": {
    "PacketLossForEC2": {
      "actionId": "aws:ssm:send-command",
      "parameters": {

```

```

        "duration": "PT30M",
        "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-Network-
Packet-Loss-Sources",
        "documentParameters": "{\"Sources\":\"us-east-1b,us-east-1c,us-
east-1d,us-east-1e,us-east-1f\", \"LossPercent\":\"15\", \"Interface\":\"DEFAULT\",
\"TrafficType\":\"egress\", \"DurationSeconds\":\"1800\", \"FlowsPercent\":\"100\",
\"InstallDependencies\":\"True\"}"
    },
    "targets": {
        "Instances": "TargetsForEC2"
    }
},
"PacketLossForECS": {
    "actionId": "aws:ecs:task-network-packet-loss",
    "parameters": {
        "sources": "us-east-1b,us-east-1c,us-east-1d,us-east-1e,us-east-1f",
        "lossPercent": "15",
        "duration": "PT30M",
        "flowsPercent": "100",
        "installDependencies": "true",
        "useEcsFaultInjectionEndpoints": "true"
    },
    "targets": {
        "Tasks": "TargetsForECS"
    }
},
"PacketLossForEKS": {
    "actionId": "aws:eks:pod-network-packet-loss",
    "parameters": {
        "sources": "us-east-1b,us-east-1c,us-east-1d,us-east-1e,us-east-1f",
        "lossPercent": "15",
        "duration": "PT30M",
        "flowsPercent": "100",
        "interface": "DEFAULT",
        "kubernetesServiceAccount": "fis-service-account"
    },
    "targets": {
        "Pods": "TargetsForEKS"
    }
}
},
"targets": {
    "TargetsForEC2": {
        "filters": [

```

```

        {
            "path": "Placement.AvailabilityZone",
            "values": [
                "us-east-1a"
            ]
        }
    ],
    "resourceTags": {
        "CrossAZTrafficSlowdown": "PacketLossForEC2"
    },
    "resourceType": "aws:ec2:instance",
    "selectionMode": "ALL"
},
"TargetsForECS": {
    "filters": [
        {
            "path": "AvailabilityZone",
            "values": [
                "us-east-1a"
            ]
        }
    ],
    "resourceTags": {
        "CrossAZTrafficSlowdown": "PacketLossForECS"
    },
    "resourceType": "aws:ecs:task",
    "selectionMode": "ALL"
},
"TargetsForEKS": {
    "parameters": {
        "availabilityZoneIdentifier": "us-east-1a",
        "clusterIdentifier": "",
        "namespace": "default",
        "selectorType": "labelSelector",
        "selectorValue": "CrossAZTrafficSlowdown=PacketLossForEKS"
    },
    "resourceType": "aws:eks:pod",
    "selectionMode": "ALL"
}
},
"experimentOptions": {
    "accountTargeting": "single-account",
    "emptyTargetResolutionMode": "skip"
},

```

```
"stopConditions": [  
  {  
    "source": "none"  
  }  
]  
}
```

Cross-Region: Connectivity

Sie können das Cross-Region: Connectivity Szenario verwenden, um den Netzwerkverkehr der Anwendung von der Versuchsregion zur Zielregion zu blockieren und die regionsübergreifende Replikation für globale Amazon S3- und Amazon DynamoDB-Tabellen mit mehreren Regionen anzuhalten. Regionsübergreifend: Die Konnektivität wirkt sich auf den ausgehenden Anwendungsdatenverkehr aus der Region aus, in der Sie das Experiment ausführen (Versuchsregion). Zustandsloser eingehender Verkehr aus der Region, die Sie von der Versuchsregion isolieren möchten (Zielregion), wird möglicherweise nicht blockiert. Der Datenverkehr von AWS Managed Services darf nicht blockiert werden.

Dieses Szenario kann verwendet werden, um zu demonstrieren, dass Anwendungen mit mehreren Regionen erwartungsgemäß funktionieren, wenn auf Ressourcen in der Zielregion von der Versuchsregion aus nicht zugegriffen werden kann. Es beinhaltet das Blockieren des Netzwerkverkehrs von der Versuchsregion zur Zielregion, indem auf Transit-Gateways und Routentabellen gezielt wird. Außerdem wird die regionsübergreifende Replikation für globale S3- und DynamoDB-Tabellen angehalten. Standardmäßig werden Aktionen, für die keine Ziele gefunden wurden, übersprungen.

Aktionen

Zusammen blockieren die folgenden Aktionen die regionsübergreifende Konnektivität für die enthaltenen AWS-Services. Die Aktionen werden parallel ausgeführt. Standardmäßig blockiert das Szenario den Verkehr für 3 Stunden, die Sie auf eine maximale Dauer von 12 Stunden erhöhen können.

Unterbrechen Sie die Transit-Gateway-Konnektivität

Cross Region: Connectivity beinhaltet [aws:network: transit-gateway-disrupt-cross -region-connectivity](#), um den regionsübergreifenden Netzwerkverkehr von VPCs der Versuchsregion zur Zielregion zu blockieren, die durch ein VPCs Transit-Gateway verbunden ist. Dies hat keinen Einfluss auf den Zugriff auf VPC-Endpunkte innerhalb der Versuchsregion, blockiert jedoch den Datenverkehr aus der Versuchsregion, der für einen VPC-Endpunkt in der Zielregion bestimmt ist.

Diese Aktion zielt auf Transit-Gateways ab, die die Versuchsregion mit der Zielregion verbinden. Standardmäßig zielt sie auf Transit-Gateways ab, [deren Tag DisruptTransitGateway](#) den Wert hat. Allowed Sie können dieses Tag zu Ihren Transit-Gateways hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Transit-Gateways gefunden werden, wird diese Aktion standardmäßig übersprungen.

Unterbrechen Sie die Subnetzkonnektivität

Cross Region: Connectivitybeinhaltet [aws:network: route-table-disrupt-cross -region-connectivity](#), um regionsübergreifenden Netzwerkverkehr von der Versuchsregion zu öffentlichen AWS-IP-Blöcken VPCs in der Zielregion zu blockieren. Zu diesen öffentlichen IP-Blöcken gehören AWS-Serviceendpunkte in der Zielregion, z. B. der regionale S3-Endpunkt, und AWS-IP-Blöcke für verwaltete Services, z. B. die IP-Adressen, die für Load Balancer und Amazon API Gateway verwendet werden. Diese Aktion blockiert auch die Netzwerkkonnektivität über regionsübergreifende VPC-Peering-Verbindungen von der Versuchsregion zur Zielregion. Es wirkt sich nicht auf den Zugriff auf VPC-Endpunkte in der Versuchsregion aus, blockiert jedoch den Datenverkehr aus der Versuchsregion, der für einen VPC-Endpunkt in der Zielregion bestimmt ist.

Diese Aktion zielt auf Subnetze in der Versuchsregion ab. Standardmäßig zielt sie auf Subnetze ab, deren [Tag DisruptSubnet](#) den Wert hat. Allowed Sie können dieses Tag zu Ihren Subnetzen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Subnetze gefunden werden, wird diese Aktion standardmäßig übersprungen.

Unterbrechen Sie die VPC-Endpunktkonnektivität

Cross Region: Connectivitybeinhaltet [aws:network: disrupt-vpc-endpoint](#) unterbricht die Konnektivität zu einem Dienst, der den Ziel-VPC-Endpunkten zugeordnet ist. Wenn ein VPC-Endpunkt beispielsweise einen privaten Link zu com.amazonaws.us-east-1.ec2 erstellt, wird die Konnektivität zu diesem Service unterbrochen.

Diese Aktion zielt VPC VPC-Endpunkte in der Versuchsregion ab. Standardmäßig zielt es auf Schnittstellen-VPC-Endpunkte mit einem [Tag](#) ab, der mit einem Wert benannt DisruptVpcEndpoint ist. Allowed Sie können dieses Tag zu Ihren VPC-Endpunkten hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen VPC-Endpoints gefunden werden, wird diese Aktion standardmäßig übersprungen.

S3-Replikation unterbrechen

Cross Region: Connectivitybeinhaltet [aws:s3: bucket-pause-replication](#), um die S3-Replikation von der Versuchsregion zur Zielregion für die Ziel-Buckets anzuhalten. Die Replikation von der Zielregion

zur Experimentregion bleibt davon unberührt. Nach dem Ende des Szenarios wird die Bucket-Replikation an dem Punkt fortgesetzt, an dem sie angehalten wurde. Beachten Sie, dass die Zeit, die für die Replikation benötigt wird, um alle Objekte synchron zu halten, je nach Dauer des Experiments und der Geschwindigkeit, mit der Objekte in den Bucket hochgeladen werden, variiert.

Diese Aktion zielt auf S3-Buckets in der Versuchsregion ab, wobei die [regionsübergreifende Replikation \(CRR\) für einen S3-Bucket in der Zielregion](#) aktiviert ist. Standardmäßig zielt sie auf Buckets ab, deren [Tag](#) den Wert `DisruptS3.Allowed` hat. Sie können dieses Tag zu Ihren Buckets hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Buckets gefunden werden, wird diese Aktion standardmäßig übersprungen.

DynamoDB-Replikation anhalten

Cross-Region: Connectivity beinhaltet [aws:dynamodb: global-table-pause-replication](#) um die Replikation zwischen der Versuchsregion und allen anderen Regionen, einschließlich der Zielregion, anzuhalten. Dies verhindert die Replikation in und aus der Versuchsregion, hat aber keinen Einfluss auf die Replikation zwischen anderen Regionen. Nach dem Ende des Szenarios wird die Tabellenreplikation an dem Punkt fortgesetzt, an dem sie angehalten wurde. Beachten Sie, dass die Zeit, die für die Replikation benötigt wird, um alle Daten synchron zu halten, von der Dauer des Experiments und der Geschwindigkeit der Änderungen an der Tabelle abhängt.

Diese Aktion zielt sowohl stark auf mehrere Regionen von DynamoDB als auch auf konsistente globale Tabellen in der Versuchsregion ab. Standardmäßig zielt sie auf Tabellen ab, deren [Tag](#) den Wert `DisruptDynamoDb` hat. Sie können dieses Tag zu Ihren Tabellen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen globalen Tabellen gefunden werden, wird diese Aktion standardmäßig übersprungen.

Halten Sie die MemoryDB-Replikation für mehrere Regionen an

Cross-Region: Connectivity beinhaltet [aws:memorydb: multi-region-cluster-pause -replication, um die Replikation](#) vom regionalen Mitgliedscluster in der Versuchsregion zu den übrigen Clustern im Zielcluster mit mehreren Regionen anzuhalten. Die Replikation zwischen anderen regionalen Mitgliedsclustern wird nicht beeinträchtigt. Nach dem Ende des Szenarios wird die Replikation an dem Punkt fortgesetzt, an dem sie angehalten wurde. Beachten Sie, dass die Zeit, bis die Replikation Daten zwischen den Mitgliedsclustern synchronisiert, je nach Dauer des Experiments und der Geschwindigkeit, mit der Daten in die Cluster geschrieben werden, variiert.

Diese Aktion zielt auf MemoryDB-Cluster mit mehreren Regionen ab, bei denen sich ein regionales Mitglied in der Versuchsregion befindet. Standardmäßig zielt sie auf Cluster mit mehreren Regionen ab, deren [Tag](#) den Wert `DisruptMemoryDB` hat. Sie können dieses Tag zu Ihren

Clustern mit mehreren Regionen hinzufügen oder das Standard-Tag in der Experimentvorlage durch Ihr eigenes Tag ersetzen. Wenn keine gültigen Cluster gefunden werden, wird diese Aktion standardmäßig übersprungen.

Einschränkungen

- Dieses Szenario beinhaltet keine [Stoppbedingungen](#). Die richtigen Stoppbedingungen für Ihre Anwendung sollten der Versuchsvorlage hinzugefügt werden.

Voraussetzungen

- Fügen Sie der AWS [FIS-Experimentrolle](#) die erforderliche Berechtigung hinzu.
- Ressourcen-Tags müssen auf Ressourcen angewendet werden, auf die das Experiment abzielen soll. Diese können Ihre eigene Tagging-Konvention oder die im Szenario definierten Standard-Tags verwenden.

Berechtigungen

Die folgende Richtlinie gewährt AWS FIS die erforderlichen Berechtigungen, um ein Experiment mit dem Cross-Region: Connectivity Szenario durchzuführen. Diese Richtlinie muss der [Rolle des Experiments](#) zugeordnet werden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RouteTableDisruptConnectivity1",
      "Effect": "Allow",
      "Action": "ec2:CreateRouteTable",
      "Resource": "arn:aws:ec2:*:*:route-table/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "RouteTableDisruptConnectivity2",
      "Effect": "Allow",
      "Action": "ec2:CreateRouteTable",
```

```

    "Resource": "arn:aws:ec2:*:*:vpc/*"
  },
  {
    "Sid": "RouteTableDisruptConnectivity21",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:route-table/*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": "CreateRouteTable",
        "aws:RequestTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "RouteTableDisruptConnectivity3",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": "CreateNetworkInterface",
        "aws:RequestTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "RouteTableDisruptConnectivity4",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:prefix-list/*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": "CreateManagedPrefixList",
        "aws:RequestTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "RouteTableDisruptConnectivity5",
    "Effect": "Allow",
    "Action": "ec2>DeleteRouteTable",
    "Resource": [
      "arn:aws:ec2:*:*:route-table/*",

```

```

        "arn:aws:ec2:*:*:vpc/*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/managedByFIS": "true"
        }
    }
},
{
    "Sid": "RouteTableDisruptConnectivity6",
    "Effect": "Allow",
    "Action": "ec2:CreateRoute",
    "Resource": "arn:aws:ec2:*:*:route-table/*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/managedByFIS": "true"
        }
    }
},
{
    "Sid": "RouteTableDisruptConnectivity7",
    "Effect": "Allow",
    "Action": "ec2:CreateNetworkInterface",
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/managedByFIS": "true"
        }
    }
},
{
    "Sid": "RouteTableDisruptConnectivity8",
    "Effect": "Allow",
    "Action": "ec2:CreateNetworkInterface",
    "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
    ]
},
{
    "Sid": "RouteTableDisruptConnectivity9",
    "Effect": "Allow",
    "Action": "ec2>DeleteNetworkInterface",
    "Resource": "arn:aws:ec2:*:*:network-interface/*",

```

```

        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/managedByFIS": "true"
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity10",
            "Effect": "Allow",
            "Action": "ec2:CreateManagedPrefixList",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity11",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteManagedPrefixList",
                "ec2:ModifyManagedPrefixList"
            ],
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "EC2DescribeResources",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeManagedPrefixLists",
                "ec2:DescribeSubnets",
                "ec2:DescribeRouteTables",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeTransitGatewayPeeringAttachments",
                "ec2:DescribeTransitGatewayAttachments",
            ]
        }
    ]
}

```

```

        "ec2:DescribeTransitGateways",
        "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
},
{
    "Sid": "RouteTableDisruptConnectivity14",
    "Effect": "Allow",
    "Action": "ec2:ReplaceRouteTableAssociation",
    "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:route-table/*"
    ]
},
{
    "Sid": "RouteTableDisruptConnectivity15",
    "Effect": "Allow",
    "Action": "ec2:GetManagedPrefixListEntries",
    "Resource": "arn:aws:ec2:*:*:prefix-list/*"
},
{
    "Sid": "RouteTableDisruptConnectivity16",
    "Effect": "Allow",
    "Action": "ec2:AssociateRouteTable",
    "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:route-table/*"
    ]
},
{
    "Sid": "RouteTableDisruptConnectivity17",
    "Effect": "Allow",
    "Action": "ec2:DisassociateRouteTable",
    "Resource": "arn:aws:ec2:*:*:route-table/*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/managedByFIS": "true"
        }
    }
},
{
    "Sid": "RouteTableDisruptConnectivity18",
    "Effect": "Allow",
    "Action": "ec2:DisassociateRouteTable",

```

```

        "Resource": "arn:aws:ec2:*:*:subnet/*"
    },
    {
        "Sid": "RouteTableDisruptConnectivity19",
        "Effect": "Allow",
        "Action": "ec2:ModifyVpcEndpoint",
        "Resource": "arn:aws:ec2:*:*:route-table/*",
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/managedByFIS": "true"
            }
        }
    },
    {
        "Sid": "TransitGatewayDisruptConnectivity1",
        "Effect": "Allow",
        "Action": [
            "ec2:DisassociateTransitGatewayRouteTable",
            "ec2:AssociateTransitGatewayRouteTable"
        ],
        "Resource": [
            "arn:aws:ec2:*:*:transit-gateway-route-table/*",
            "arn:aws:ec2:*:*:transit-gateway-attachment/*"
        ]
    },
    {
        "Sid": "S3CrossRegion1",
        "Effect": "Allow",
        "Action": "s3:ListAllMyBuckets",
        "Resource": "*"
    },
    {
        "Sid": "S3CrossRegion3",
        "Effect": "Allow",
        "Action": "s3:PauseReplication",
        "Resource": "arn:aws:s3::*:*",
        "Condition": {
            "StringLike": {
                "s3:DestinationRegion": "*"
            }
        }
    },
    {
        "Sid": "S3CrossRegion4",

```

```

    "Effect": "Allow",
    "Action": [
        "s3:GetReplicationConfiguration",
        "s3:PutReplicationConfiguration"
    ],
    "Resource": "arn:aws:s3:::*",
    "Condition": {
        "BoolIfExists": {
            "s3:isReplicationPauseRequest": "true"
        }
    }
},
{
    "Sid": "DynamoDbPauseReplication",
    "Effect": "Allow",
    "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:PutResourcePolicy",
        "dynamodb:GetResourcePolicy",
        "dynamodb>DeleteResourcePolicy"
    ],
    "Resource": [
        "arn:aws:dynamodb:*:*:table/*"
    ]
},
{
    "Sid": "DynamoDbMrscPauseReplication",
    "Effect": "Allow",
    "Action": [
        "dynamodb:InjectError"
    ],
    "Resource": ["*"]
},
{
    "Sid": "ResolveResourcesViaTags",
    "Effect": "Allow",
    "Action": "tag:GetResources",
    "Resource": "*"
},
{
    "Sid": "MemDbCrossRegion",
    "Effect": "Allow",
    "Action": [
        "memorydb:DescribeMultiRegionClusters",

```



```

        "memorydb:PauseMultiRegionClusterReplication"
    ],
    "Resource": [
        "arn:aws:memorydb:*:*:multiregioncluster/*"
    ]
},
{
    "Sid": "DisruptVPCE1",
    "Effect": "Allow",
    "Action": "ec2:CreateSecurityGroup",
    "Resource": [
        "arn:aws:ec2:*:*:vpc/*",
        "arn:aws:ec2:*:*:security-group/*"
    ]
},
{
    "Sid": "DisruptVPCE2",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:security-group/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateSecurityGroup",
            "aws:RequestTag/managedByFIS": "true"
        }
    }
},
{
    "Sid": "DisruptVPCE3",
    "Effect": "Allow",
    "Action": [
        "ec2:DeleteSecurityGroup",
        "ec2:RevokeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2:*:*:security-group/*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/managedByFIS": "true"
        }
    }
},
{
    "Sid": "DisruptVPCE4",
    "Effect": "Allow",

```

```

        "Action": "vpce:AllowMultiRegion",
        "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*"
    },
    {
        "Sid": "ModifyVPCE",
        "Effect": "Allow",
        "Action": "ec2:ModifyVpcEndpoint",
        "Resource": [
            "arn:aws:ec2:*:*:vpc-endpoint/*",
            "arn:aws:ec2:*:*:security-group/*"
        ]
    }
]
}

```

Inhalt des Szenarios

Der folgende Inhalt definiert das Szenario. Diese JSON-Datei kann gespeichert und verwendet werden, um mithilfe des [create-experiment-template](#) Befehls von der AWS-Befehlszeilenschnittstelle (AWS CLI) eine [Versuchsvorlage](#) zu erstellen. Die neueste Version des Szenarios finden Sie in der Szenariobibliothek in der FIS-Konsole.

```

{
    "targets": {
        "Transit-Gateway": {
            "resourceType": "aws:ec2:transit-gateway",
            "resourceTags": {
                "TgwTag": "TgwValue"
            },
            "selectionMode": "ALL"
        },
        "Subnet": {
            "resourceType": "aws:ec2:subnet",
            "resourceTags": {
                "SubnetKey": "SubnetValue"
            },
            "selectionMode": "ALL",
            "parameters": {}
        },
        "VPC-Endpoint": {
            "resourceType": "aws:ec2:vpc-endpoint",
            "resourceTags": {
                "DisruptPrivateLink": "Allowed"
            }
        }
    }
}

```

```

        },
        "selectionMode": "ALL"
    },
    "S3-Bucket": {
        "resourceType": "aws:s3:bucket",
        "resourceTags": {
            "S3Impact": "Allowed"
        },
        "selectionMode": "ALL"
    },
    "DynamoDB-Global-Table": {
        "resourceType": "aws:dynamodb:global-table",
        "resourceTags": {
            "DisruptDynamoDb": "Allowed"
        },
        "selectionMode": "ALL"
    },
    "MemoryDB-Multi-Region-Cluster": {
        "resourceType": "aws:memorydb:multi-region-cluster",
        "resourceTags": {
            "DisruptMemoryDb": "Allowed"
        },
        "selectionMode": "ALL"
    }
},
"actions": {
    "Disrupt-Transit-Gateway-Connectivity": {
        "actionId": "aws:network:transit-gateway-disrupt-cross-region-
connectivity",
        "parameters": {
            "duration": "PT3H",
            "region": "eu-west-1"
        },
        "targets": {
            "TransitGateways": "Transit-Gateway"
        }
    },
    "Disrupt-Subnet-Connectivity": {
        "actionId": "aws:network:route-table-disrupt-cross-region-
connectivity",
        "parameters": {
            "duration": "PT3H",
            "region": "eu-west-1"
        },
    },

```

```

        "targets": {
            "Subnets": "Subnet"
        }
    },
    "Disrupt-Vpc-Endpoint": {
        "actionId": "aws:network:disrupt-vpc-endpoint",
        "parameters": {
            "duration": "PT3H"
        },
        "targets": {
            "VPCEndpoints": "VPC-Endpoint"
        }
    },
    "Pause-S3-Replication": {
        "actionId": "aws:s3:bucket-pause-replication",
        "parameters": {
            "duration": "PT3H",
            "region": "eu-west-1"
        },
        "targets": {
            "Buckets": "S3-Bucket"
        }
    },
    "Pause-DynamoDB-Replication": {
        "actionId": "aws:dynamodb:global-table-pause-replication",
        "parameters": {
            "duration": "PT3H"
        },
        "targets": {
            "Tables": "DynamoDB-Global-Table"
        }
    },
    "Pause-MemoryDB-Multi-Region-Cluster-Replication": {
        "actionId": "aws:memorydb:multi-region-cluster-pause-replication",
        "parameters": {
            "duration": "PT3H",
            "region": "eu-west-1"
        },
        "targets": {
            "MultiRegionClusters": "MemoryDB-Multi-Region-Cluster"
        }
    }
},
"stopConditions": [

```

```
        {
            "source": "none"
        }
    ],
    "roleArn": "",
    "logConfiguration": {
        "logSchemaVersion": 2
    },
    "tags": {
        "Name": "Cross-Region: Connectivity"
    },
    "experimentOptions": {
        "accountTargeting": "single-account",
        "emptyTargetResolutionMode": "skip"
    },
    "description": "Block application network traffic from experiment Region to
target Region and pause cross-Region replication"
}
```

Arbeiten mit Multi-Account-Experimenten für AWS FIS

Sie können Versuchsvorlagen für mehrere Konten über die AWS FIS Konsole oder die Befehlszeile erstellen und verwalten. Sie erstellen ein Experiment mit mehreren Konten, indem Sie die Option für das Konten-Targeting-Experiment als "multi-account" angeben und Zielkontenkonfigurationen hinzufügen. Nachdem Sie eine Versuchsvorlage für mehrere Konten erstellt haben, können Sie sie verwenden, um ein Experiment durchzuführen.

Mit einem Experiment mit mehreren Konten können Sie reale Ausfallszenarien für eine Anwendung einrichten und ausführen, die sich über mehrere AWS Konten innerhalb einer Region erstreckt. Sie führen Experimente mit mehreren Konten von einem Orchestrator-Konto aus durch, was sich auf Ressourcen in mehreren Zielkonten auswirkt.

Wenn Sie ein Experiment mit mehreren Konten durchführen, werden Zielkonten mit betroffenen Ressourcen über ihre AWS Health-Dashboards benachrichtigt, sodass die Benutzer in den Zielkonten darüber informiert werden. Mit Experimenten mit mehreren Konten können Sie:

- Führen Sie reale Ausfallszenarien für Anwendungen durch, die sich über mehrere Konten erstrecken, mit den zentralen Kontrollen und Leitplanken, die dies bieten. AWS FIS
- Steuern Sie die Auswirkungen eines Experiments mit mehreren Konten mithilfe von IAM-Rollen mit fein abgestuften Berechtigungen und Tags, um den Umfang der einzelnen Ziele zu definieren.
- Sehen Sie sich anhand der Protokolle zentral an AWS FIS , welche Aktionen in den AWS-Managementkonsole einzelnen Konten ausgeführt werden. AWS FIS
- Überwachen und prüfen Sie API-Aufrufe AWS FIS , die in jedem Konto bei AWS getätigt CloudTrail werden.

Dieser Abschnitt hilft Ihnen bei den ersten Schritten mit Experimenten mit mehreren Konten.

Topics

- [Konzepte für Experimente mit mehreren Konten](#)
- [Bewährte Methoden für Experimente mit mehreren Konten](#)
- [Voraussetzungen für Experimente mit mehreren Konten](#)
- [Erstellen Sie eine Versuchsvorlage für mehrere Konten](#)
- [Aktualisieren Sie die Konfiguration eines Zielkontos](#)
- [Löschen Sie eine Zielkontokonfiguration](#)

Konzepte für Experimente mit mehreren Konten

Im Folgenden sind die wichtigsten Konzepte für Experimente mit mehreren Konten aufgeführt:

- **Orchestrator-Konto** — Das Orchestrator-Konto dient als zentrales Konto für die Konfiguration und Verwaltung des Experiments in der AWS FIS Konsole sowie für die zentrale Protokollierung. Das Orchestrator-Konto besitzt die AWS FIS Versuchsvorlage und das Experiment.
- **Zielkonten** — Ein Zielkonto ist ein einzelnes AWS-Konto mit Ressourcen, die durch ein Experiment mit AWS FIS mehreren Konten beeinträchtigt werden können.
- **Zielkontokonfigurationen** — Sie definieren die Zielkonten, die Teil eines Experiments sind, indem Sie Zielkontenkonfigurationen zur Versuchsvorlage hinzufügen. Eine Zielkontenkonfiguration ist ein Element der Versuchsvorlage, das für Experimente mit mehreren Konten erforderlich ist. Sie definieren eines für jedes Zielkonto, indem Sie eine AWS Konto-ID, eine IAM-Rolle und eine optionale Beschreibung festlegen.

Bewährte Methoden für Experimente mit mehreren Konten

Im Folgenden finden Sie bewährte Methoden für die Verwendung von Experimenten mit mehreren Konten:

- Wenn Sie Ziele für Experimente mit mehreren Konten konfigurieren, empfehlen wir, für alle Zielkonten einheitliche Ressourcen-Tags zu verwenden. Bei einem AWS FIS Experiment werden Ressourcen mit konsistenten Tags in jedem Zielkonto aufgelöst. Eine Aktion muss mindestens eine Zielressource in einem beliebigen Zielkonto auflösen, andernfalls schlägt sie fehl, außer bei Experimenten mit der `emptyTargetResolutionMode` Einstellung aufskip. Aktionskontingente gelten pro Konto. Wenn Sie Ressourcen nach Ressourcen aussuchen möchten ARNs, gilt dasselbe Limit für ein einzelnes Konto pro Aktion.
- Wenn Sie mithilfe von Parametern oder Filtern gezielt Ressourcen in einer oder mehreren Availability Zones verwenden, sollten Sie eine AZ-ID und keinen AZ-Namen angeben. Die AZ-ID ist eine eindeutige und konsistente Kennung für eine Availability Zone für alle Konten. Informationen dazu, wie Sie die AZ-ID für die Availability Zones in Ihrem Konto finden, finden Sie unter [Availability Zone IDs für Ihre AWS-Ressourcen](#).

Voraussetzungen für Experimente mit mehreren Konten

Um Stoppbedingungen für ein Experiment mit mehreren Konten zu verwenden, müssen Sie zunächst Alarme für mehrere Konten konfigurieren. IAM-Rollen werden definiert, wenn Sie eine Versuchsvorlage für mehrere Konten erstellen. Sie können die erforderlichen IAM-Rollen erstellen, bevor Sie die Vorlage erstellen.

Inhalt

- [Berechtigungen für Experimente mit mehreren Konten](#)
- [Stoppen Sie die Bedingungen für Experimente mit mehreren Konten \(optional\)](#)
- [Sicherheitshebel für Experimente mit mehreren Konten \(optional\)](#)

Berechtigungen für Experimente mit mehreren Konten

Bei Experimenten mit mehreren Konten wird die IAM-Rollenverkettung verwendet, AWS FIS um Berechtigungen für Aktionen mit Ressourcen in Zielkonten zu erteilen. Bei Experimenten mit mehreren Konten richten Sie IAM-Rollen für jedes Zielkonto und das Orchestrator-Konto ein. Diese IAM-Rollen erfordern eine Vertrauensbeziehung zwischen den Zielkonten und dem Orchestrator-Konto sowie zwischen dem Orchestrator-Konto und AWS FIS.

Die IAM-Rollen für die Zielkonten enthalten die Berechtigungen, die erforderlich sind, um Maßnahmen mit Ressourcen zu ergreifen. Sie werden für eine Versuchsvorlage erstellt, indem Zielkontenkonfigurationen hinzugefügt werden. Sie erstellen eine IAM-Rolle für das Orchestrator-Konto mit der Berechtigung, die Rollen von Zielkonten zu übernehmen und eine Vertrauensbeziehung mit AWS FIS. Diese IAM-Rolle wird als Vorlage `roleArn` für das Experiment verwendet.

Weitere Informationen zur Rollenverkettung finden Sie unter [Begriffe und Konzepte von Rollen](#) im IAM-Benutzerhandbuch.

Im folgenden Beispiel richten Sie Berechtigungen für ein Orchestrator-Konto A ein, mit dem Sie `aws:ebs:pause-volume-io` im Zielkonto B ein Experiment durchführen können.

1. Erstellen Sie in Konto B eine IAM-Rolle mit den für die Ausführung der Aktion erforderlichen Berechtigungen. Informationen zu den für jede Aktion erforderlichen Berechtigungen finden Sie unter [Referenz zu Aktionen](#). Das folgende Beispiel zeigt die Berechtigungen, die ein Zielkonto für die Ausführung der EBS-Aktion [the section called “aws:ebs:pause-volume-io”](#) Pause Volume IO gewährt.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:PauseVolumeIO"
      ],
      "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": "*"
    }
  ]
}
```

2. Fügen Sie als Nächstes eine Vertrauensrichtlinie zu Konto B hinzu, die eine Vertrauensbeziehung mit Konto A herstellt. Wählen Sie einen Namen für die IAM-Rolle für Konto A, die Sie in Schritt 3 erstellen werden.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "AccountIdA"
      },
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": "*"
    }
  ]
}
```

```

        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "sts:ExternalId":
"arn:aws:fis:region:accountIdA:experiment/*"
            },
            "ArnEquals": {
                "aws:PrincipalArn":
"arn:aws:iam::111122223333:role/role_name"
            }
        }
    }
}

```

- Erstellen Sie in Konto A eine IAM-Rolle. Dieser Rollenname muss mit der Rolle übereinstimmen, die Sie in der Vertrauensrichtlinie in Schritt 2 angegeben haben. Um mehrere Konten ins Visier zu nehmen, gewähren Sie dem Orchestrator die Berechtigungen, jede Rolle zu übernehmen. Das folgende Beispiel zeigt die Berechtigungen für Konto A, Konto B anzunehmen. Wenn Sie zusätzliche Zielkonten haben, fügen Sie dieser Richtlinie eine zusätzliche Rolle ARNs hinzu. Sie können nur einen Rollen-ARN pro Zielkonto haben.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::111122223333:role/role_name"
            ]
        }
    ]
}

```

- Diese IAM-Rolle für Konto A wird als `roleArn` für die Experimentvorlage verwendet. Das folgende Beispiel zeigt die Vertrauensrichtlinie, die für die IAM-Rolle erforderlich ist und die AWS FIS Berechtigungen zur Übernahme von Konto A, dem Orchestrator-Konto, gewährt.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "fis.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Sie können Stacksets auch verwenden, um mehrere IAM-Rollen gleichzeitig bereitzustellen. Für die Nutzung CloudFormation StackSets müssen Sie die erforderlichen StackSet Berechtigungen in Ihren Konten einrichten. AWS Weitere Informationen finden Sie unter [Arbeiten mit AWS CloudFormation StackSets](#).

Stoppen Sie die Bedingungen für Experimente mit mehreren Konten (optional)

Eine Stoppbedingung ist ein Mechanismus, um ein Experiment zu beenden, wenn es einen Schwellenwert erreicht, den Sie als Alarm definieren. Um eine Stoppbedingung für Ihr Experiment mit mehreren Konten einzurichten, können Sie kontenübergreifende Alarmer verwenden. Sie müssen die gemeinsame Nutzung in jedem Zielkonto aktivieren, damit der Alarm dem Orchestrator-Konto mit Leseberechtigungen zur Verfügung steht. Nach der Freigabe können Sie mit Metric Math Metriken aus verschiedenen Zielkonten kombinieren. Anschließend können Sie diesen Alarm als Stoppbedingung für das Experiment hinzufügen.

Weitere Informationen zu kontenübergreifenden Dashboards finden Sie unter [Aktivierung kontenübergreifender Funktionen](#) in CloudWatch

Sicherheitshebel für Experimente mit mehreren Konten (optional)

Sicherheitshebel werden verwendet, um alle laufenden Experimente zu unterbrechen und zu verhindern, dass neue Experimente gestartet werden. Möglicherweise möchten Sie den Sicherheitshebel verwenden, um FIS-Experimente während bestimmter Zeiträume oder als Reaktion auf Betriebsalarme in der Anwendung zu verhindern. Jedes AWS Konto hat einen Sicherheitshebel pro AWS-Region. Wenn ein Sicherheitshebel aktiviert ist, wirkt sich dies auf alle Experimente aus, die auf demselben Konto und in derselben Region wie der Sicherheitshebel durchgeführt werden. Um Experimente mit mehreren Konten zu beenden oder zu verhindern, muss der Sicherheitshebel für dasselbe Konto und dieselbe Region aktiviert werden, in der die Experimente durchgeführt werden.

Erstellen Sie eine Versuchsvorlage für mehrere Konten

Um zu erfahren, wie Sie eine Versuchsvorlage über die AWS-Managementkonsole

Siehe [Erstellen Sie eine Versuchsvorlage](#).

So erstellen Sie eine Experimentvorlage mit der CLI

1. Öffnen Sie das AWS Command Line Interface
2. Um ein Experiment aus einer gespeicherten JSON-Datei zu erstellen, bei der die Option für das Account-Targeting-Experiment auf gesetzt ist "multi-account" (z. B. `my-template.json`), ersetzen Sie die Platzhalterwerte in *italics* durch Ihre eigenen Werte und führen Sie dann den folgenden [create-experiment-template](#) Befehl aus.

```
aws fis create-experiment-template --cli-input-json file://my-template.json
```

Dadurch wird die Versuchsvorlage in der Antwort zurückgegeben. Kopieren Sie das `id` aus der Antwort, das ist die ID der Experimentvorlage.

3. Führen Sie den [create-target-account-configuration](#) Befehl aus, um der Versuchsvorlage eine Zielkontokonfiguration hinzuzufügen. Ersetzen Sie die Platzhalterwerte in *italics* durch Ihre eigenen Werte, verwenden Sie die Werte `id` aus Schritt 2 als Wert für den `--experiment-template-id` Parameter, und führen Sie dann den folgenden Befehl aus. Der Parameter `--description` ist optional. Wiederholen Sie diesen Schritt für jedes Zielkonto.

```
aws fis create-target-account-configuration --experiment-template-id EXTxxxxxxxxx
--account-id 111122223333 --role-arn arn:aws:iam::111122223333:role/role-name --
description "my description"
```

4. Führen Sie den [get-target-account-configuration](#) Befehl aus, um die Details für eine bestimmte Zielkontokonfiguration abzurufen.

```
aws fis get-target-account-configuration --experiment-template-id EXTxxxxxxxxx --
account-id 111122223333
```

5. Nachdem Sie alle Ihre Zielkontokonfigurationen hinzugefügt haben, können Sie den [list-target-account-configurations](#) Befehlsbefehl ausführen, um zu überprüfen, ob Ihre Zielkontokonfigurationen erstellt wurden.

```
aws fis list-target-account-configurations --experiment-template-id EXTxxxxxxxxx
```

Sie können auch überprüfen, ob Sie Zielkontokonfigurationen hinzugefügt haben, indem Sie den [get-experiment-template](#) Befehl ausführen. Die Vorlage gibt ein schreibgeschütztes Feld zurück `targetAccountConfigurationsCount`, das die Anzahl aller Zielkontokonfigurationen in der Experimentvorlage angibt.

6. Wenn Sie bereit sind, können Sie die Experimentvorlage mit dem Befehl [start-experiment](#) ausführen.

```
aws fis start-experiment --experiment-template-id EXTxxxxxxxxx
```

Aktualisieren Sie die Konfiguration eines Zielkontos

Sie können eine bestehende Zielkontokonfiguration aktualisieren, wenn Sie den Rollen-ARN oder die Beschreibung für das Konto ändern möchten. Wenn Sie die Konfiguration eines Zielkontos aktualisieren, wirken sich die Änderungen nicht auf laufende Experimente aus, bei denen die Vorlage verwendet wird.

Um eine Zielkontokonfiguration mit der AWS-Managementkonsole zu aktualisieren:

1. Öffnen Sie die AWS FIS Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich die Option Experimentvorlagen aus.

3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Experimentvorlage aktualisieren.
4. Wählen Sie im Seitenbereich Schritt 3, Servicezugriff konfigurieren aus.
5. Ändern Sie die Konfigurationen des Zielkontos und wählen Sie Experimentvorlage aktualisieren aus.
6. Wählen Sie Schritt 5, Überprüfen und erstellen aus.

So aktualisieren Sie die Konfiguration eines Zielkontos mit der CLI

Führen Sie den [update-target-account-configuration](#) Befehl zu Befehl aus und ersetzen Sie dabei die Platzhalterwerte in *italics* durch Ihre eigenen Werte. Die `--description` Parameter `--role-arn` und sind optional und werden nicht aktualisiert, wenn sie nicht enthalten sind.

```
aws fis update-target-account-configuration --experiment-template-id EXTxxxxxxxxx
--account-id 111122223333 --role-arn arn:aws:iam::111122223333:role/role-name --
description "my description"
```

Löschen Sie eine Zielkontokonfiguration

Wenn Sie eine Zielkontokonfiguration nicht mehr benötigen, können Sie sie löschen. Wenn Sie eine Zielkontokonfiguration löschen, sind alle laufenden Experimente, die die Vorlage verwenden, nicht betroffen. Das Experiment wird so lange ausgeführt, bis es abgeschlossen oder gestoppt wird.

Um eine Zielkontokonfiguration mit dem zu löschen AWS-Managementkonsole

1. Öffnen Sie die AWS FIS Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Aktualisieren.
4. Wählen Sie im Seitenbereich Schritt 3, Dienstzugriff konfigurieren aus.
5. Wählen Sie unter Zielkontokonfigurationen für den Zielkonto-Rollen-ARN, den Sie löschen möchten, die Option Entfernen aus.
6. Wählen Sie Schritt 5, Überprüfen und erstellen aus.
7. Überprüfen Sie die Vorlage und wählen Sie Experimentvorlage aktualisieren. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie die Eingabe ein `update` und wählen Sie „Experimentvorlage aktualisieren“.

So löschen Sie eine Zielkontokonfiguration mit der CLI

Führen Sie den [delete-target-account-configuration](#) Befehl aus und ersetzen Sie die Platzhalterwerte in *italics* durch Ihre eigenen Werte.

```
aws fis update-target-account-configuration --experiment-template-id EXTxxxxxxxxx --  
account-id 111122223333
```

Planung von Experimenten

Mit dem AWS Fault Injection Service (FIS) können Sie Fault Injection-Experimente an Ihren AWS-Workloads durchführen. Diese Experimente werden auf Vorlagen ausgeführt, die eine oder mehrere Aktionen zur Ausführung auf bestimmten Zielen enthalten. Sie können Ihre Experimente jetzt nativ von der FIS-Konsole aus als einmalige Aufgabe oder als wiederkehrende Aufgaben planen. Zusätzlich zu den [geplanten Regeln](#) bietet FIS jetzt eine neue Planungsfunktion. FIS ist jetzt in EventBridge Scheduler integriert und erstellt Regeln in Ihrem Namen. EventBridge Scheduler ist ein serverloser Scheduler, mit dem Sie Aufgaben von einem zentralen, verwalteten Dienst aus erstellen, ausführen und verwalten können.

Important

Experiment Scheduler with AWS Fault Injection Service ist in AWS GovCloud (USA-Ost) und AWS GovCloud (US-West) nicht verfügbar.

Themen

- [Erstellen Sie eine Scheduler-Rolle](#)
- [Erstellen Sie einen Zeitplan für Experimente](#)
- [Aktualisieren Sie einen Experimentplan](#)
- [Deaktivieren oder löschen Sie einen Experimentplan](#)

Erstellen Sie eine Scheduler-Rolle

Eine Ausführungsrolle ist eine IAM-Rolle, die AWS FIS davon ausgeht, dass sie mit dem EventBridge Scheduler interagiert und der Event Bridge-Scheduler das FIS-Experiment startet. Sie fügen dieser Rolle Berechtigungsrichtlinien hinzu, um dem EventBridge Scheduler Zugriff zum Aufrufen des FIS-Experiments zu gewähren. In den folgenden Schritten wird beschrieben, wie Sie eine neue Ausführungsrolle und eine Richtlinie erstellen, mit der Sie ein Experiment starten können EventBridge .

Scheduler-Rolle mit der AWS-CLI erstellen

Diese IAM-Rolle ist erforderlich, damit Event Bridge Experimente im Namen des Kunden planen kann.

1. Kopieren Sie die folgende JSON-Richtlinie zur Übernahme der Rolle und speichern Sie sie lokal unter `fis-execution-role.json`. Diese Vertrauensrichtlinie ermöglicht es EventBridge Scheduler, die Rolle in Ihrem Namen zu übernehmen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "scheduler.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Geben Sie über die AWS-Befehlszeilenschnittstelle (AWS CLI) den folgenden Befehl ein, um eine neue Rolle zu erstellen. `FisSchedulerExecutionRole` Ersetzen Sie ihn durch den Namen, den Sie dieser Rolle geben möchten.

```
aws iam create-role --role-name FisSchedulerExecutionRole --assume-role-policy-document file://fis-execution-role.json
```

Bei Erfolg wird die folgende Ausgabe angezeigt:

```
{
  "Role": {
    "Path": "/",
    "RoleName": "FisSchedulerExecutionRole",
    "RoleId": "AROAZL22PDN5A6WKRBNUN",
    "Arn": "arn:aws:iam::123456789012:role/FisSchedulerExecutionRole",
    "CreateDate": "2023-08-24T17:23:05+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
```

```

        "Service": "scheduler.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
}
}

```

- Um eine neue Richtlinie zu erstellen, die es EventBridge Scheduler ermöglicht, das Experiment aufzurufen, kopieren Sie den folgenden JSON-Code und speichern Sie ihn lokal unter `fis-start-experiment-permissions.json`. Die folgende Richtlinie ermöglicht es EventBridge Scheduler, die `fis:StartExperiment` Aktion für alle Versuchsvorlagen in Ihrem Konto aufzurufen. Ersetzen Sie das `*` am Ende von `"arn:aws:fis:*:*:experiment-template/*"` durch die ID Ihrer Experimentvorlage, wenn Sie die Rolle auf eine einzige Experimentvorlage beschränken möchten.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "fis:StartExperiment",
      "Resource": [
        "arn:aws:fis:*:*:experiment-template/*",
        "arn:aws:fis:*:*:experiment/*"
      ]
    }
  ]
}

```

- Führen Sie den folgenden Befehl aus, um die neue Berechtigungsrichtlinie zu erstellen. Ersetzen Sie `FisSchedulerPolicy` durch den Namen, den Sie dieser Richtlinie geben möchten.

```
aws iam create-policy --policy-name FisSchedulerPolicy --policy-document file://fis-start-experiment-permissions.json
```

Bei Erfolg wird die folgende Ausgabe angezeigt. Beachten Sie den Richtlinien-ARN. Sie verwenden diesen ARN im nächsten Schritt, um die Richtlinie an unsere Ausführungsrolle anzuhängen.

```
{
  "Policy": {
    "PolicyName": "FisSchedulerPolicy",
    "PolicyId": "ANPAZL22PDN5ESVUWXLBD",
    "Arn": "arn:aws:iam::123456789012:policy/FisSchedulerPolicy",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2023-08-24T17:34:45+00:00",
    "UpdateDate": "2023-08-24T17:34:45+00:00"
  }
}
```

5. Führen Sie den folgenden Befehl aus, um die Richtlinie an Ihre Ausführungsrolle anzuhängen. `your-policy-arn` Ersetzen Sie es durch den ARN der Richtlinie, die Sie im vorherigen Schritt erstellt haben. `FisSchedulerExecutionRole` Ersetzen Sie ihn durch den Namen Ihrer Ausführungsrolle.

```
aws iam attach-role-policy --policy-arn your-policy-arn --role-name
FisSchedulerExecutionRole
```

Der `attach-role-policy` Vorgang gibt keine Antwort in der Befehlszeile zurück.

6. Sie können den Scheduler so einschränken, dass er nur AWS FIS Versuchsvorlagen ausführt, die einen bestimmten Tag-Wert haben. Die folgende Richtlinie gewährt beispielsweise die `fis:StartExperiment` Erlaubnis für alle AWS FIS Experimente, beschränkt den Planer jedoch darauf, nur Versuchsvorlagen auszuführen, die markiert sind. `Purpose=Schedule`

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
        "Effect": "Allow",
        "Action": "fis:StartExperiment",
        "Resource": "arn:aws:fis:*:*:experiment/*"
    },
    {
        "Effect": "Allow",
        "Action": "fis:StartExperiment",
        "Resource": "arn:aws:fis:*:*:experiment-template/*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/Purpose": "Schedule"
            }
        }
    }
}
]
```

Erstellen Sie einen Zeitplan für Experimente

Bevor Sie ein Experiment planen, benötigen Sie eines oder mehrere, [Komponenten der Versuchsvorlage](#) damit Ihr Zeitplan es aufrufen kann. Sie können eine bestehende AWS-Ressource verwenden oder eine neue erstellen.

Sobald die Versuchsvorlage erstellt wurde, klicken Sie auf Aktionen und wählen Sie Experiment planen aus. Sie werden zur Seite „Experiment planen“ weitergeleitet. Der Name des Zeitplans wird für Sie ausgefüllt.

Folgen Sie dem Abschnitt mit dem Zeitplanmuster und wählen Sie entweder einen einmaligen Zeitplan oder einen wiederkehrenden Zeitplan aus. Füllen Sie die erforderlichen Eingabefelder aus und navigieren Sie zu den Berechtigungen.

Schedule pattern

Occurrence [Info](#)
You can define an one-time or recurrent schedule.

☒ One-time schedule ☐ Recurring schedule

Date and time
The date and time to invoke the target.

YYYY/MM/DD (UTC-04:00) America/New_...

YYYY/MM/DD Use 24-hour format timestamp (hh:mm) Timezone

Flexible time window
If you choose a flexible time window, Scheduler invokes your schedule within the time window you specify. For example, if you choose 15 minutes, your schedule runs within 15 minutes after the schedule start time.

Select

Schedule state

Enable schedule
You can choose not to enable the schedule now. You will be able to enable the schedule after it has been created.

☒ Enable

Der Zeitplanstatus ist standardmäßig aktiviert. Hinweis: Wenn Sie den Zeitplanstatus deaktivieren, wird das Experiment auch dann nicht geplant, wenn Sie einen Zeitplan erstellen.

AWS FIS Der Experiment Scheduler basiert auf dem [EventBridge Scheduler](#). Die verschiedenen unterstützten [Zeitplantypen](#) finden Sie in der Dokumentation.

Um den Zeitplan über die Konsole zu aktualisieren

1. Öffnen Sie die [AWS FIS -Konsole](#).
2. Wählen Sie im linken Navigationsbereich die Option Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus, für die Sie den Zeitplan erstellen möchten.
4. Klicken Sie auf Aktionen und wählen Sie im Drop-down-Menü die Option Experiment planen aus.
 - a. Unter Name des Zeitplans wird der Name auto ausgefüllt.
 - b. Wählen Sie unter Zeitplanmuster die Option Wiederkehrender Zeitplan aus.
 - c. [Unter Zeitplantyp können Sie einen tarifbasierten Zeitplan auswählen, siehe Zeitplantypen.](#)
 - d. Wählen Sie unter Rate expression eine Rate aus, die langsamer ist als die Ausführungszeit Ihres Experiments, z. B. 5 Minuten.
 - e. Wählen Sie unter Zeitrahmen Ihre Zeitzone aus.
 - f. Geben Sie unter Startdatum und -zeit ein Startdatum und eine Startzeit an.
 - g. Geben Sie unter Enddatum und -zeit ein Enddatum und eine Endzeit an

- h. Schalten Sie unter Zeitplanstatus die Option Zeitplan aktivieren um.
 - i. Wählen Sie unter Berechtigungen die Option Bestehende Rolle verwenden aus, und suchen `FisSchedulerExecutionRole` Sie dann nach.
 - j. Wählen Sie Weiter aus.
5. Wählen Sie Zeitplan überprüfen und erstellen aus, überprüfen Sie Ihre Terminplanerdetails und wählen Sie dann Zeitplan erstellen aus.

Aktualisieren Sie einen Experimentplan

Sie können einen Experimentplan so aktualisieren, dass er an einem bestimmten Datum und zu einer bestimmten Uhrzeit stattfindet, die für Sie geeignet sind.

So aktualisieren Sie die Ausführung eines Experiments mithilfe der Konsole

1. Öffnen Sie die [Amazon FIS-Konsole](#).
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Ressourcentyp: Experimentvorlage, für die bereits ein Zeitplan erstellt wurde.
4. Klicken Sie auf die Experiment-ID für die Vorlage. Navigieren Sie dann zur Registerkarte „Zeitpläne“.
5. Prüfen Sie, ob dem Experiment ein vorhandener Zeitplan zugeordnet ist. Wählen Sie den zugehörigen Zeitplan aus und klicken Sie auf die Schaltfläche Zeitplan aktualisieren.

Deaktivieren oder löschen Sie einen Experimentplan

Um zu verhindern, dass ein Experiment nach einem Zeitplan ausgeführt oder ausgeführt wird, können Sie die Regel löschen oder deaktivieren. In den folgenden Schritten erfahren Sie, wie Sie eine Versuchsausführung mithilfe der AWS Konsole löschen oder deaktivieren können.

Um eine Regel zu löschen oder zu deaktivieren

1. Öffnen Sie die [Amazon FIS-Konsole](#).
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Ressourcentyp: Experimentvorlage, für die bereits ein Zeitplan erstellt wurde.
4. Klicken Sie auf die Experiment-ID für die Vorlage. Navigieren Sie dann zur Registerkarte „Zeitpläne“.

5. Prüfen Sie, ob dem Experiment ein vorhandener Zeitplan zugeordnet ist. Wählen Sie den zugehörigen Zeitplan aus und klicken Sie auf die Schaltfläche Zeitplan aktualisieren.
6. Führen Sie eine der folgenden Aktionen aus:
 - a. Um den Zeitplan zu löschen, klicken Sie auf die Schaltfläche neben der Regel Zeitplan löschen. Geben Sie Zeitplan ein `delete` und klicken Sie auf die Schaltfläche Zeitplan löschen.
 - b. Um den Zeitplan zu deaktivieren, klicken Sie auf die Schaltfläche neben der Regel Zeitplan deaktivieren. Geben Sie den Befehl Zeitplan deaktivieren ein **disable** und klicken Sie auf die Schaltfläche Zeitplan deaktivieren.

Sicherheitshebel für AWS FIS

Sicherheitshebel werden verwendet, um alle laufenden Experimente zu beenden und zu verhindern, dass neue Experimente beginnen. Möglicherweise möchten Sie den Sicherheitshebel verwenden, um FIS-Experimente während bestimmter Zeiträume oder als Reaktion auf Betriebsalarme in der Anwendung zu verhindern. Jedes AWS Konto hat einen Sicherheitshebel pro AWS-Region.

Bei laufenden Experimenten, die durch den Sicherheitshebel gestoppt werden, zahlen Sie nur für die Aktionsdauer, die vor dem Abbruch des Experiments lief. Bei Experimenten, deren Start verhindert wird, fallen keine Kosten an. In den folgenden Abschnitten finden Sie Informationen zu den ersten Schritten mit der Verwendung von Sicherheitshebeln.

Themen

- [Konzepte für Sicherheitshebel](#)
- [Mit Sicherheitshebeln arbeiten](#)

Konzepte für Sicherheitshebel

Ein Sicherheitshebel kann aktiviert oder gelöst werden.

- Wenn der Schalter ausgeschaltet ist, sind FIS-Experimente erlaubt. Standardmäßig sind die Sicherheitshebel deaktiviert.
- Wenn diese Option aktiviert ist, werden laufende Experimente gestoppt und es dürfen keine neuen Experimente gestartet werden.

Ein Experiment, das durch einen Sicherheitshebel beeinflusst wird, endet in einem der folgenden Zustände:

- Abgebrochen, wenn das Experiment lief, als der Sicherheitshebel betätigt war.
- Abgebrochen, wenn das Experiment gestartet wurde, als der Sicherheitshebel bereits betätigt war.

Sie können ein Experiment, das gestoppt oder abgebrochen wurde, nicht fortsetzen oder erneut ausführen. Sie können jedoch ein neues Experiment mit derselben Versuchsvorlage beginnen, sobald der Sicherheitshebel gelöst ist.

Ressource für den Sicherheitshebel

Safety Lever ist eine Ressource, die durch den Amazon Resource Name (ARN) definiert ist. Sicherheitshebel beinhalten die folgenden Parameter:

- Status, der entweder aktiviert oder deaktiviert ist.
- Grund. Dabei handelt es sich um eine vom Benutzer eingegebene Zeichenfolge, um zu protokollieren, warum der Status des Sicherheitshebels geändert wurde.

Mit Sicherheitshebeln arbeiten

In diesem Abschnitt wird detailliert beschrieben, wie Sie Sicherheitshebel über die AWS FIS Konsole oder die Befehlszeile anzeigen, ein- und ausschalten können.

Sicherheitshebel anzeigen

Sie können den Status Ihres Sicherheitshebels für Ihr Konto und Ihre Region überprüfen, indem Sie die folgenden Schritte ausführen.

So können Sie sich einen Sicherheitshebel über die Konsole ansehen

1. [Öffnen Sie die AWS FIS Konsole](#)
2. Wählen Sie im Navigationsbereich Experimente aus.
3. Wenn der Sicherheitshebel betätigt ist, wird oben auf der Seite ein Warnbanner angezeigt. Wenn kein Warnbanner angezeigt wird, ist der Sicherheitshebel deaktiviert.

So zeigen Sie einen Sicherheitshebel mit der CLI an

- Verwenden Sie den folgenden Befehl:

```
aws fis get-safety-lever --id "default"
```

Ein Sicherheitshebel kann sich in einem der folgenden Zustände befinden:

- Deaktiviert - Der Sicherheitshebel beeinträchtigt keine Experimente. Experimente können ungehindert ablaufen. Die Sicherheitshebel sind standardmäßig deaktiviert.

- **Einkuppeln** - Der Sicherheitshebel wechselt von gelöst zu aktiviert. Möglicherweise gibt es immer noch Experimente, die noch nicht eingestellt wurden. In diesem Zustand kann der Sicherheitshebel nicht verändert werden.
- **Aktiviert** - Der Sicherheitshebel ist aktiv und es laufen keine Experimente. Alle neuen Experimente, bei denen versucht wird, bei betätigtem Sicherheitshebel zu starten, werden abgebrochen.

Betätigen eines Sicherheitshebels

Um einen Sicherheitshebel über die Konsole zu betätigen

1. [Öffnen Sie die AWS FIS Konsole](#)
2. Wählen Sie im Navigationsbereich Experimente aus.
3. Wählen Sie die Schaltfläche Alle Experimente beenden.
4. Geben Sie den Grund für das Betätigen des Sicherheitshebels ein.
5. Wählen Sie Bestätigen aus.

Um einen Sicherheitshebel mit der CLI zu aktivieren

- Verwenden Sie den folgenden -Befehl. Füllen Sie das Feld „Grund“ mit Ihrer eigenen Antwort aus.

```
aws fis update-safety-lever-state --id "default" --state  
"status=engaged,reason=xxxxx"
```

Einen Sicherheitshebel lösen

Um einen Sicherheitshebel mit der Konsole zu lösen

1. [Öffnen Sie die Konsole AWS FIS](#)
2. Wählen Sie im Navigationsbereich Experimente aus.
3. Wählen Sie die Taste „Sicherheitshebel abschalten“.
4. Geben Sie den Grund für das Lösen des Sicherheitshebels ein.
5. Wählen Sie Bestätigen aus.

Um einen Sicherheitshebel mit der CLI zu deaktivieren

- Verwenden Sie den folgenden Befehl:

```
aws fis update-safety-lever-state --id "default" --state  
"status=disengaged,reason=recovered"
```

Überwachung von AWS FIS-Experimenten

Sie können die folgenden Tools verwenden, um den Fortschritt und die Auswirkungen Ihrer Experimente mit dem AWS AWS Fault Injection Service (FIS) zu überwachen.

AWS FIS-Konsole und AWS CLI

Verwenden Sie die AWS FIS-Konsole oder die AWS CLI , um den Fortschritt eines laufenden Experiments zu überwachen. Sie können den Status jeder Aktion im Experiment und die Ergebnisse jeder Aktion einsehen. Weitere Informationen finden Sie unter [the section called “Sehen Sie sich Ihre Experimente an”](#).

CloudWatch Nutzungsmetriken und Alarmer

Verwenden Sie CloudWatch Nutzungsmetriken, um einen Überblick über die Ressourcennutzung Ihres Kontos zu erhalten. AWS Die FIS-Nutzungsmetriken entsprechen den AWS Servicekontingenten. Sie können Alarmer konfigurieren, mit denen Sie benachrichtigt werden, wenn sich Ihre Nutzung einem Servicekontingent nähert. Weitere Informationen finden Sie unter [Überwachen Sie mit CloudWatch](#).

Sie können auch Stoppbedingungen für Ihre AWS FIS-Experimente festlegen, indem Sie CloudWatch Alarmer einrichten, die definieren, wann ein Experiment die Grenzwerte überschreitet. Wenn der Alarm ausgelöst wird, stoppt das Experiment. Weitere Informationen finden Sie unter [Bedingungen beenden](#). Weitere Informationen zum Erstellen von CloudWatch Alarmen finden Sie unter [Erstellen eines CloudWatch Alarms auf der Grundlage eines statischen Schwellenwerts](#) und [Erstellen eines CloudWatch Alarms auf der Grundlage von Anomalieerkennung](#) im CloudWatch Amazon-Benutzerhandbuch.

AWS Protokollierung von FIS-Experimenten

Aktivieren Sie die Protokollierung von Experimenten, um während der Ausführung detaillierte Informationen über Ihr Experiment zu erfassen. Weitere Informationen finden Sie unter [Protokollierung von Experimenten](#).

Ereignisse zur Änderung des Versuchsstatus

EventBridge Mit Amazon können Sie automatisch auf Systemereignisse oder Ressourcenänderungen reagieren. AWS FIS sendet eine Benachrichtigung, wenn sich der Status eines Experiments ändert. Sie können Regeln für die Ereignisse erstellen, die Sie interessieren und die die automatische Aktion festlegen, die ergriffen werden soll, wenn ein Ereignis einer Regel

entspricht. Zum Beispiel das Senden einer Benachrichtigung an ein Amazon SNS SNS-Thema oder das Aufrufen einer Lambda-Funktion. Weitere Informationen finden Sie unter [Überwachen Sie mit EventBridge](#).

CloudTrail Logs

Wird verwendet AWS CloudTrail , um detaillierte Informationen zu den Aufrufen der AWS FIS-API zu erfassen und sie als Protokolldateien in Amazon S3 zu speichern. CloudTrail protokolliert auch Aufrufe an den Service APIs für die Ressourcen, mit denen Sie Experimente durchführen. Anhand dieser CloudTrail Protokolle können Sie ermitteln, welche Anrufe getätigt wurden, von welcher Quell-IP-Adresse der Anruf kam, wer den Anruf getätigt hat, wann der Anruf getätigt wurde usw.

AWS Benachrichtigungen im Health Dashboard

AWS Health bietet fortlaufenden Einblick in die Leistung Ihrer Ressourcen und die Verfügbarkeit Ihrer AWS Services und Konten. Wenn Sie ein Experiment starten, sendet AWS FIS eine Benachrichtigung an Ihr AWS Health Dashboard. Die Benachrichtigung ist für die Dauer des Experiments in jedem Konto vorhanden, das Ressourcen enthält, auf die ein Experiment abzielt, einschließlich Experimenten mit mehreren Konten. Bei Experimenten mit mehreren Konten, die nur Aktionen beinhalten, die keine Ziele beinhalten, wie z. B. `aws:ssm:start-automation-execution` und `aws:fis:wait`, wird keine Benachrichtigung ausgegeben. Informationen über die Rolle, die für die Durchführung des Experiments verwendet wurde, werden unter Betroffene Ressourcen aufgeführt. Weitere Informationen zum AWS Health Dashboard finden Sie unter [AWS Health Dashboard](#) im AWS Health Health-Benutzerhandbuch.

Note

AWS Health bietet Veranstaltungen nach bestem Wissen und Gewissen an.

Überwachen Sie die AWS FIS-Nutzungsmetriken mit Amazon CloudWatch

Sie können Amazon verwenden CloudWatch , um die Auswirkungen von AWS FIS-Experimenten auf Ziele zu überwachen. Sie können auch Ihre AWS FIS-Nutzung überwachen.

Weitere Informationen zum Anzeigen des Status eines Experiments finden Sie unter [Sehen Sie sich Ihre Experimente an](#).

Überwachen Sie AWS FIS-Experimente

Identifizieren Sie bei der Planung Ihrer AWS FIS-Experimente die CloudWatch Metriken, anhand derer Sie den Ausgangswert oder den „stabilen Zustand“ für die Zielressourcentypen für das Experiment ermitteln können. Nachdem Sie ein Experiment gestartet haben, können Sie diese CloudWatch Metriken für die in der Experimentvorlage ausgewählten Ziele überwachen.

Weitere Informationen zu den verfügbaren CloudWatch Metriken für einen von AWS FIS unterstützten Zielressourcentyp finden Sie im Folgenden:

- [Überwachen Sie Ihre Instances mit CloudWatch](#)
- [Amazon CloudWatch ECS-Metriken](#)
- [Überwachung von Amazon RDS-Metriken mit CloudWatch](#)
- [Überwachung von Run Command-Metriken mithilfe von CloudWatch](#)

AWS FIS-Nutzungsmetriken

Sie können CloudWatch Nutzungsmetriken verwenden, um einen Überblick über die Ressourcennutzung Ihres Kontos zu erhalten. Verwenden Sie diese Metriken, um Ihre aktuelle Servicenutzung in CloudWatch Diagrammen und Dashboards zu visualisieren.

AWS Die FIS-Nutzungsmetriken entsprechen den AWS Servicekontingenten. Sie können Alarmer konfigurieren, mit denen Sie benachrichtigt werden, wenn sich Ihre Nutzung einem Servicekontingent nähert. Weitere Informationen zu CloudWatch Alarmen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

AWS FIS veröffentlicht die folgende Metrik im AWS/Usage-Namespace.

Metrik	Beschreibung
ResourceCount	Die Gesamtanzahl der angegebenen Ressourcen, die in Ihrem Konto ausgeführt werden. Die Ressource wird durch die Dimensionen definiert, die der Metrik zugeordnet sind.

Die folgenden Dimensionen werden verwendet, um die von FIS veröffentlichten Nutzungsmetriken zu verfeinern. AWS

Dimension	Beschreibung
Service	Der Name des AWS Dienstes, der die Ressource enthält. Für AWS FIS-Nutzungsmetriken lautet der Wert für diese Dimension. FIS
Type	Der Typ von Entität, die gemeldet wird. Derzeit AWS ist der einzig gültige Wert für FIS-Nutzungsmetriken. Resource
Resource	Der Typ der Ressource, die ausgeführt wird. Die möglichen Werte gelten ExperimentTemplates für Experimentvorlagen und ActiveExperiments für aktive Experimente.
Class	Diese Dimension ist für die future Verwendung reserviert.

Überwachen Sie AWS FIS-Experimente mit Amazon EventBridge

Wenn sich der Status eines Experiments ändert, sendet AWS FIS eine Benachrichtigung. Diese Benachrichtigungen werden als Ereignisse über Amazon zur Verfügung gestellt EventBridge (früher als CloudWatch Ereignisse bezeichnet). AWS FIS sendet diese Ereignisse nach bestem Wissen und Gewissen aus. Ereignisse werden nahezu EventBridge in Echtzeit übertragen.

Mit EventBridge können Sie Regeln erstellen, die als Reaktion auf ein Ereignis programmgesteuerte Aktionen auslösen. Sie können beispielsweise eine Regel konfigurieren, die ein SNS-Thema aufruft, um eine E-Mail-Benachrichtigung zu senden, oder eine Lambda-Funktion aufruft, um eine Aktion auszuführen.

Weitere Informationen zu EventBridge finden Sie unter [Erste Schritte mit Amazon EventBridge](#) im EventBridge Amazon-Benutzerhandbuch.

Im Folgenden wird die Syntax eines Ereignisses zur Änderung des Versuchsstatus beschrieben:

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "FIS Experiment State Change",
  "source": "aws.fis",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "region",
  "resources": [
    "arn:aws:fis:region:account_id:experiment/experiment-id"
  ],
  "detail": {
    "experiment-id": "EXPaBCD1efg2HIJkL3",
    "experiment-template-id": "EXTa1b2c3de5f6g7h",
    "new-state": {
      "status": "new_value",
      "reason": "reason_string"
    },
    "old-state": {
      "status": "old_value",
      "reason": "reason_string"
    }
  }
}
```

experiment-id

Die ID des Experiments, dessen Status sich geändert hat.

experiment-template-id

Die ID der vom Experiment verwendeten Experimentvorlage.

new_value

Der neue Status des Experiments. Die möglichen Werte sind:

- completed
- failed
- initiating
- running

- stopped
- stopping

old_value

Der vorherige Stand des Experiments. Die möglichen Werte sind:

- initiating
- pending
- running
- stopping

Protokollierung von Experimenten für AWS FIS

Sie können die Protokollierung von Experimenten verwenden, um während der Durchführung detaillierte Informationen über Ihr Experiment zu erfassen.

Die Protokollierung von Experimenten wird auf der Grundlage der Kosten berechnet, die mit den einzelnen Protokollzieltypen verbunden sind. Weitere Informationen finden Sie unter [CloudWatch Amazon-Preise](#) (unter Bezahltes Kontingent, Logs, Vended Logs) und [Amazon S3-Preise](#).

Berechtigungen

Sie müssen AWS FIS-Berechtigungen zum Senden von Protokollen an jedes von Ihnen konfigurierte Protokollziel gewähren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch im Folgenden:

- [An Logs gesendete CloudWatch Protokolle](#)
- [An Amazon S3 gesendete Protokolle](#)

Protokollschema

Das Folgende ist das Schema, das bei der Protokollierung von Experimenten verwendet wird. Die aktuelle Schemaversion ist 2. Die Felder für details hängen vom Wert von ablog_type. Die Felder für resolved_targets hängen vom Wert von abtarget_type. Weitere Informationen finden Sie unter [the section called “Beispiel für Protokolldatensätze”](#).

```
{
  "id": "EXP123abc456def789",
```

```

    "log_type": "experiment-start | target-resolution-start | target-resolution-detail
| target-resolution-end | action-start | action-error | action-end | experiment-end",
    "event_timestamp": "yyyy-mm-ddThh:mm:ssZ",
    "version": "2",
    "details": {
        "account_id": "123456789012",
        "action_end_time": "yyyy-mm-ddThh:mm:ssZ",
        "action_id": "String",
        "action_name": "String",
        "action_start_time": "yyyy-mm-ddThh:mm:ssZ",
        "action_state": {
            "status": "pending | initiating | running | completed | cancelled |
stopping | stopped | failed",
            "reason": "String"
        },
        "action_targets": "String to string map",
        "error_information": "String",
        "experiment_end_time": "yyyy-mm-ddThh:mm:ssZ",
        "experiment_state": {
            "status": "pending | initiating | running | completed | stopping | stopped
| failed",
            "reason": "String"
        },
        "experiment_start_time": "yyyy-mm-ddThh:mm:ssZ",
        "experiment_template_id": "String",
        "page": Number,
        "parameters": "String to string map",
        "resolved_targets": [
            {
                "field": "value"
            }
        ],
        "resolved_targets_count": Number,
        "status": "failed | completed",
        "target_name": "String",
        "target_resolution_end_time": "yyyy-mm-ddThh:mm:ssZ",
        "target_resolution_start_time": "yyyy-mm-ddThh:mm:ssZ",
        "target_type": "String",
        "total_pages": Number,
        "total_resolved_targets_count": Number
    }
}

```

Versionshinweise

- Version 2 führt ein:
 - Das `target_type` Feld und ändert das `resolved_targets` Feld von einer Liste von ARNs zu einer Liste von Objekten. Die gültigen Felder für das `resolved_targets` Objekt hängen vom Wert von `abtarget_type`, der dem [Ressourcentyp](#) der Ziele entspricht.
 - Die `target-resolution-detail` Ereignistypen `action-error` und die Ereignistypen, die das `account_id` Feld hinzufügen.
- Version 1 ist die erste Version.

Ziele protokollieren

AWS FIS unterstützt die Protokollzustellung an die folgenden Ziele:

- Ein Amazon-S3-Bucket
- Eine Amazon CloudWatch Logs-Protokollgruppe

Lieferung von S3-Protokollen

Die Protokolle werden an den folgenden Speicherort übermittelt.

```
bucket-and-optional-prefix/AWSLogs/account-id/fis/region/experiment-id/YYYY/MM/DD/account-id_awsfislogs_region_experiment-id_YYYYMMDDHHMMZ_hash.log
```

Es kann mehrere Minuten dauern, bis die Protokolle an den Bucket geliefert werden.

CloudWatch Logs, Protokollzustellung

Die Protokolle werden in einen Protokollstreamnamed `/aws/fis/übertragen`*experiment-id*.

Protokolle werden in weniger als einer Minute an die Protokollgruppe übermittelt.

Beispiel für Protokolldatensätze

Im Folgenden finden Sie Beispielprotokolldatensätze für ein Experiment, bei dem die `aws:ec2:reboot-instances` Aktion auf einer zufällig ausgewählten EC2 Instanz ausgeführt wird.

Datensätze

- [Start des Experiments](#)
- [target-resolution-start](#)
- [target-resolution-detail](#)
- [target-resolution-end](#)
- [Aktion-Start](#)
- [Aktion-Ende](#)
- [Aktionsfehler](#)
- [Ende des Experiments](#)

Experiment-Start

Im Folgenden finden Sie einen Beispieldatensatz für das `experiment-start` Ereignis.

```
{
  "id": "EXPhjAXCGY78HV2a4A",
  "log_type": "experiment-start",
  "event_timestamp": "2023-05-31T18:50:45Z",
  "version": "2",
  "details": {
    "experiment_template_id": "EXTCDh1M8HHkhxoaQ",
    "experiment_start_time": "2023-05-31T18:50:43Z"
  }
}
```

target-resolution-start

Im Folgenden finden Sie einen Beispieldatensatz für das `target-resolution-start` Ereignis.

```
{
  "id": "EXPhjAXCGY78HV2a4A",
  "log_type": "target-resolution-start",
  "event_timestamp": "2023-05-31T18:50:45Z",
  "version": "2",
  "details": {
    "target_resolution_start_time": "2023-05-31T18:50:45Z",
  }
}
```

```
    "target_name": "EC2InstancesToReboot"
  }
}
```

target-resolution-detail

Im Folgenden finden Sie einen Beispieldatensatz für das `target-resolution-detail` Ereignis. Wenn die Zielauflösung fehlschlägt, enthält der Datensatz auch das `error_information` Feld.

```
{
  "id": "EXPhjAXCGY78HV2a4A",
  "log_type": "target-resolution-detail",
  "event_timestamp": "2023-05-31T18:50:45Z",
  "version": "2",
  "details": {
    "target_resolution_end_time": "2023-05-31T18:50:45Z",
    "target_name": "EC2InstancesToReboot",
    "target_type": "aws:ec2:instance",
    "account_id": "123456789012",
    "resolved_targets_count": 2,
    "status": "completed"
  }
}
```

target-resolution-end

Wenn die Zielauflösung fehlschlägt, enthält der Datensatz auch das `error_information` Feld. Wenn `total_pages` es größer als 1 ist, hat die Anzahl der aufgelösten Ziele die Größenbeschränkung für einen Datensatz überschritten. Es gibt zusätzliche `target-resolution-end` Datensätze, die die verbleibenden aufgelösten Ziele enthalten.

Im Folgenden finden Sie einen Beispieldatensatz für das `target-resolution-end` Ereignis einer EC2 Aktion.

```
{
  "id": "EXPhjAXCGY78HV2a4A",
  "log_type": "target-resolution-end",
  "event_timestamp": "2023-05-31T18:50:45Z",
  "version": "2",
```

```

"details": {
  "target_resolution_end_time": "2023-05-31T18:50:46Z",
  "target_name": "EC2InstanceToReboot",
  "target_type": "aws:ec2:instance",
  "resolved_targets": [
    {
      "arn": "arn:aws:ec2:us-east-1:123456789012:instance/
i-0f7ee2abfffc330de5"
    }
  ],
  "page": 1,
  "total_pages": 1
}
}

```

Im Folgenden finden Sie einen Beispieldatensatz für das `target-resolution-end` Ereignis für eine EKS-Aktion.

```

{
  "id": "EXP24YfiucfyVPJpEJn",
  "log_type": "target-resolution-end",
  "event_timestamp": "2023-05-31T18:50:45Z",
  "version": "2",
  "details": {
    "target_resolution_end_time": "2023-05-31T18:50:46Z",
    "target_name": "myPods",
    "target_type": "aws:eks:pod",
    "resolved_targets": [
      {
        "pod_name": "example-696fb6498b-sxhw5",
        "namespace": "default",
        "cluster_arn": "arn:aws:eks:us-east-1:123456789012:cluster/fis-demo-
cluster",
        "target_container_name": "example"
      }
    ],
    "page": 1,
    "total_pages": 1
  }
}

```

Aktion-Start

Im Folgenden finden Sie einen Beispieldatensatz für das `action-start` Ereignis. Wenn in der Versuchsvorlage Parameter für die Aktion angegeben sind, enthält der Datensatz auch das `parameters` Feld.

```
{
  "id": "EXPhjAXCGY78HV2a4A",
  "log_type": "action-start",
  "event_timestamp": "2023-05-31T18:50:56Z",
  "version": "2",
  "details": {
    "action_name": "Reboot",
    "action_id": "aws:ec2:reboot-instances",
    "action_start_time": "2023-05-31T18:50:56Z",
    "action_targets": {"Instances":"EC2InstancesToReboot"}
  }
}
```

Aktion-Fehler

Im Folgenden finden Sie einen Beispieldatensatz für das `action-error` Ereignis. Dieses Ereignis wird nur zurückgegeben, wenn eine Aktion fehlschlägt. Es wird für jedes Konto zurückgegeben, bei dem die Aktion fehlschlägt.

```
{
  "id": "EXPhjAXCGY78HV2a4A",
  "log_type": "action-error",
  "event_timestamp": "2023-05-31T18:50:56Z",
  "version": "2",
  "details": {
    "action_name": "pause-io",
    "action_id": "aws:ebs:pause-volume-io",
    "account_id": "123456789012",
    "action_state": {
      "status": "failed",
      "reason": "Unable to start Pause Volume IO. Target volumes must be attached to an instance type based on the Nitro system. VolumeId(s): [vol-1234567890abcdef0]:"
    }
  }
}
```

Ende der Aktion

Im Folgenden finden Sie einen Beispieldatensatz für das action-end Ereignis.

```
{
  "id": "EXPhjAXCGY78HV2a4A",
  "log_type": "action-end",
  "event_timestamp": "2023-05-31T18:50:56Z",
  "version": "2",
  "details": {
    "action_name": "Reboot",
    "action_id": "aws:ec2:reboot-instances",
    "action_end_time": "2023-05-31T18:50:56Z",
    "action_state": {
      "status": "completed",
      "reason": "Action was completed."
    }
  }
}
```

Ende des Experiments

Im Folgenden finden Sie einen Beispieldatensatz für das experiment-end Ereignis.

```
{
  "id": "EXPhjAXCGY78HV2a4A",
  "log_type": "experiment-end",
  "event_timestamp": "2023-05-31T18:50:57Z",
  "version": "2",
  "details": {
    "experiment_end_time": "2023-05-31T18:50:57Z",
    "experiment_state": {
      "status": "completed",
      "reason": "Experiment completed"
    }
  }
}
```

Aktivieren Sie die Protokollierung der Experimente

Die Protokollierung von Experimenten ist standardmäßig deaktiviert. Um Versuchsprotokolle für ein Experiment zu erhalten, müssen Sie das Experiment anhand einer Experimentvorlage mit aktivierter Protokollierung erstellen. Wenn Sie zum ersten Mal ein Experiment ausführen, das so konfiguriert ist,

dass es ein Ziel verwendet, das zuvor nicht für die Protokollierung verwendet wurde, verzögern wir das Experiment, um die Protokollzustellung an dieses Ziel zu konfigurieren, was etwa 15 Sekunden dauert.

Um die Protokollierung von Experimenten über die Konsole zu aktivieren

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Experimentvorlage aktualisieren.
4. Konfigurieren Sie für Logs die Zieloptionen. Um Logs an einen S3-Bucket zu senden, wählen Sie An einen Amazon S3 S3-Bucket senden und geben Sie den Bucket-Namen und das Präfix ein. Um Logs an Logs zu CloudWatch senden, wählen Sie Send to CloudWatch Logs und geben Sie die Log-Gruppe ein.
5. Wählen Sie „Experimentvorlage aktualisieren“.

Um die Protokollierung von Experimenten zu aktivieren, verwenden Sie AWS CLI

Verwenden Sie den [update-experiment-template](#)Befehl und geben Sie eine Protokollkonfiguration an.

Deaktivieren Sie die Protokollierung der Experimente

Wenn Sie keine Protokolle für Ihre Experimente mehr erhalten möchten, können Sie die Protokollierung der Experimente deaktivieren.

Um die Protokollierung von Experimenten über die Konsole zu deaktivieren

1. Öffnen Sie die AWS FIS-Konsole unter <https://console.aws.amazon.com/fis/>.
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie die Experimentvorlage aus und klicken Sie dann auf Aktionen, Experimentvorlage aktualisieren.
4. Deaktivieren Sie für Logs die Optionen An einen Amazon S3 S3-Bucket senden und An CloudWatch Logs senden.
5. Wählen Sie Experimentvorlage aktualisieren.

Um die Protokollierung von Experimenten zu deaktivieren, verwenden Sie AWS CLI

Verwenden Sie den [update-experiment-template](#) Befehl und geben Sie eine leere Protokollkonfiguration an.

API-Aufrufe protokollieren mit AWS CloudTrail

AWS Der AWS Fault Injection Service (FIS) ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in AWS FIS ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für AWS FIS als Ereignisse. Zu den aufgezeichneten Aufrufen gehören Aufrufe von der AWS FIS-Konsole und Codeaufrufen für die AWS FIS-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für AWS FIS. 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 AWS FIS gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

Benutzen CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto , wenn Sie das Konto erstellen. Wenn in AWS FIS Aktivitäten auftreten, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS-Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem System AWS-Konto, einschließlich der Ereignisse für AWS FIS, 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 erstellen, gilt der Trail standardmäßig für alle AWS 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:

- [Erstellen Sie einen Trail für Ihr AWS Konto](#)
- [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 AWS FIS-Aktionen werden von der [AWS Fault Injection Service API-Referenz](#) protokolliert CloudTrail und sind dort dokumentiert. Informationen zu den Versuchsaktionen, die auf einer Zielressource ausgeführt werden, finden Sie in der API-Referenzdokumentation für den Dienst, dem die Ressource gehört. Aktionen, die auf einer EC2 Amazon-Instance ausgeführt werden, finden Sie beispielsweise in der [Amazon EC2 API-Referenz](#).

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Root- oder -Benutzeranmeldeinformationen 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 Service gestellt wurde.

Weitere Informationen finden Sie unter [CloudTrail userIdentity-Element](#).

Verstehen Sie die Einträge in AWS der FIS-Protokolldatei

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Im Folgenden finden Sie ein Beispiel für einen CloudTrail Protokolleintrag für einen Aufruf der AWS StopExperiment FIS-Aktion.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAIOSFODNN7EXAMPLE:jdoe",
    "arn": "arn:aws:sts::111122223333:assumed-role/example/jdoe",
```

```

"accountId": "111122223333",
"accessKeyId": "AKIAI44QH8DHBEXAMPLE",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::111122223333:role/example",
    "accountId": "111122223333",
    "userName": "example"
  },
  "webIdFederationData": {},
  "attributes": {
    "creationDate": "2020-12-03T09:40:42Z",
    "mfaAuthenticated": "false"
  }
},
"eventTime": "2020-12-03T09:44:20Z",
"eventSource": "fis.amazonaws.com",
"eventName": "StopExperiment",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.51.100.25",
"userAgent": "Boto3/1.22.9 Python/3.8.13 Linux/5.4.186-113.361.amzn2int.x86_64
Botocore/1.25.9",
"requestParameters": {
  "clientToken": "1234abc5-6def-789g-012h-ijklm34no56p",
  "experimentTemplateId": "ABCDE1fgHIjKlMnOp",
  "tags": {}
},
"responseElements": {
  "experiment": {
    "actions": {
      "exampleAction1": {
        "actionId": "aws:ec2:stop-instances",
        "duration": "PT10M",
        "state": {
          "reason": "Initial state",
          "status": "pending"
        },
        "targets": {
          "Instances": "exampleTag1"
        }
      },
      "exampleAction2": {

```

```

    "actionId": "aws:ec2:stop-instances",
    "duration": "PT10M",
    "state": {
      "reason": "Initial state",
      "status": "pending"
    },
    "targets": {
      "Instances": "exampleTag2"
    }
  },
  "creationTime": 1605788649.95,
  "endTime": 1606988660.846,
  "experimentTemplateId": "ABCDE1fgHIJkLmNop",
  "id": "ABCDE1fgHIJkLmNop",
  "roleArn": "arn:aws:iam::111122223333:role/AllowFISActions",
  "startTime": 1605788650.109,
  "state": {
    "reason": "Experiment stopped",
    "status": "stopping"
  },
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:example"
    }
  ],
  "tags": {},
  "targets": {
    "ExampleTag1": {
      "resourceTags": {
        "Example": "tag1"
      },
      "resourceType": "aws:ec2:instance",
      "selectionMode": "RANDOM(1)"
    },
    "ExampleTag2": {
      "resourceTags": {
        "Example": "tag2"
      },
      "resourceType": "aws:ec2:instance",
      "selectionMode": "RANDOM(1)"
    }
  }
}

```

```

    }
  },
  "requestID": "1abcd23e-f4gh-567j-klm8-9np01q234r56",
  "eventID": "1234a56b-c78d-9e0f-g1h2-34jk56m7n890",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

Im Folgenden finden Sie ein Beispiel für einen CloudTrail Protokolleintrag für eine API-Aktion, die AWS FIS im Rahmen eines Experiments aufgerufen hat, das die FIS-Aktion beinhaltet. `aws:ssm:send-command` AWS Das `userIdentity` Element spiegelt eine Anfrage wider, die mit temporären Anmeldeinformationen gestellt wurde, die durch die Übernahme einer Rolle abgerufen wurden. Der Name der angenommenen Rolle erscheint in `userName`. Die ID des Experiments, `EXP21nT17WMzA6dnGz`, erscheint in `principalId` und als Teil des ARN der angenommenen Rolle.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AR0ATZZZ4JPIXUEXAMPLE:EXP21nT17WMzA6dnUgz",
    "arn": "arn:aws:sts::111122223333:assumed-role/AllowActions/EXP21nT17WMzA6dnUgz",
    "accountId": "111122223333",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AR0ATZZZ4JPIXUEXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/AllowActions",
        "accountId": "111122223333",
        "userName": "AllowActions"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-05-30T13:23:19Z",
        "mfaAuthenticated": "false"
      }
    }
  }
}

```

```
    },
    "invokedBy": "fis.amazonaws.com"
  },
  "eventTime": "2022-05-30T13:23:19Z",
  "eventSource": "ssm.amazonaws.com",
  "eventName": "ListCommands",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "fis.amazonaws.com",
  "userAgent": "fis.amazonaws.com",
  "requestParameters": {
    "commandId": "51dab97f-489b-41a8-a8a9-c9854955dc65"
  },
  "responseElements": null,
  "requestID": "23709ced-c19e-471a-9d95-cf1a06b50ee6",
  "eventID": "145fe5a6-e9d5-45cc-be25-b7923b950c83",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```

Problembehebung AWS FIS

AWS FIS Gibt detaillierte Fehler aus den `GetExperiment` API- und FIS-Experimentprotokollen zurück, um Fehler zu beheben. Fehler werden als Teil des Versuchsstatus zurückgegeben, wenn der Status des Experiments fehlgeschlagen ist. Wenn mehrere Aktionen fehlschlagen, wird die erste fehlgeschlagene Aktion als Experimentfehler zurückgegeben. Sie können Ihre FIS-Versuchsprotokolle auf weitere Fehler überprüfen. Informationen zum Protokollieren und Überwachen von AWS FIS Experimenten finden Sie unter [Überwachung von AWS FIS-Experimenten](#).

Je nach Art des Fehlers wird möglicherweise einer der folgenden Fehler angezeigt:

- Grund: Detaillierte Beschreibung des spezifischen Fehlers. Ursachenwerte sollten nicht für die Automatisierung verwendet werden, da sie sich ändern können.
- Code: Die Art des Fehlers. Codewerte sollten nicht für die Automatisierung verwendet werden, da sie sich ändern können, sofern in der Tabelle unten nichts anderes angegeben ist.
- Ort: Kontext für den Abschnitt der Experimentvorlage, der fehlgeschlagen ist, z. B. die Aktion oder das Ziel.
- Konto-ID: Das AWS Konto, bei dem der Fehler aufgetreten ist.

Fehlercodes

Fehlercode	Beschreibung des Codes
<code>ConfigurationFailure</code>	Die Aktion, das Ziel, das Experiment oder das Protokoll sind nicht richtig konfiguriert. Überprüfen Sie den Fehler <code>location</code> und stellen Sie sicher, dass die Parameter und Konfigurationen korrekt sind.
<code>DependentServiceFailure</code>	Bei einem anderen AWS Dienst ist ein Fehler aufgetreten. Versuchen Sie erneut, das Experiment auszuführen.
<code>InternalFailure</code>	Bei der Ausführung des Experiments ist ein interner Fehler aufgetreten. Sie können auf der Grundlage dieses Fehlercodes automatisieren.

Fehlercode	Beschreibung des Codes
InvalidTarget	Ein Ziel konnte während der Zielauflösung oder zu Beginn einer Aktion nicht gelöst werden. Dies kann auf einen der folgenden Gründe zurückzuführen sein: • Das Ziel existiert nicht, z. B. wenn es gelöscht wurde oder wenn der ARN falsch ist. • Es gibt ein Tag für Ihr Ziel, das keine Ressourcen auflöst. • Es gibt eine Aktion, die nicht mit einem Ziel verknüpft ist. Um Fehler zu beheben, überprüfen Sie Ihre Protokolle, um festzustellen, welche Ziele nicht behoben wurden. Vergewissern Sie sich, dass alle Aktionen mit Zielen verknüpft sind und dass Ihre Ressourcen-ID oder Tags vorhanden sind und nicht falsch geschrieben wurden.

Fehlercode	Beschreibung des Codes
AuthorizationFailure	Es gibt zwei Hauptursachen für das Scheitern von Experimenten aufgrund von Berechtigungsfehlern: • Die IAM-Rolle, auf die Sie abzielen, verfügt nicht über die erforderlichen Berechtigungen, um Ziele zu lösen oder Maßnahmen für Ihre Ressourcen zu ergreifen. Um diesen Fehler zu beheben, überprüfen Sie die für Ihre Aktionen erforderlichen Berechtigungen in der FIS-Aktionsreferenz und fügen Sie sie Ihrer experimentellen IAM-Rolle hinzu. • Die Erstellung einer AWS Service-Linked Role (SLR) für FIS wurde durch eine Service Control Policy (SCP) in Ihrer Organisation verweigert. FIS verwendet das SLR, um die Überwachung und die Auswahl der Ressourcen für Experimente zu verwalten. Weitere Informationen finden Sie unter Dienstbezogene Rollenberechtigungen für FIS AWS.
QuotaExceededFailure	Das Kontingent für den Ressourcentyp wurde überschritten. Informationen darüber, ob das Kontingent erhöht werden kann, finden Sie unter Kontingente und Einschränkungen für den AWS Fault Injection Service.

Sicherheit im AWS Fault Injection Service

Cloud-Sicherheit hat AWS höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen 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. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#). Weitere Informationen zu den Compliance-Programmen, die für den AWS Fault Injection Service gelten, finden Sie unter [AWS Services im Bereich nach Compliance-Programm AWS](#).
- **Sicherheit in der Cloud** — Ihre Verantwortung richtet sich nach dem AWS Dienst, 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 Verwendung von AWS FIS anwenden können. In den folgenden Themen erfahren Sie, wie Sie AWS FIS konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer AWS FIS-Ressourcen unterstützen.

Inhalt

- [Datenschutz im AWS Fault Injection Service](#)
- [Identitäts- und Zugriffsmanagement für den AWS Fault Injection Service](#)
- [Sicherheit der Infrastruktur im AWS Fault Injection Service](#)
- [Greifen Sie über einen VPC-Endpunkt \(\) mit einer Schnittstelle auf AWS FIS zu AWS PrivateLink](#)

Datenschutz im AWS Fault Injection Service

Das [Modell der AWS gemeinsamen Verantwortung](#) gilt für den Datenschutz im AWS Fault Injection Service. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der

globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS -Modell der geteilten Verantwortung und in der DSGVO](#) im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit AWS FIS oder anderen Geräten arbeiten und dabei die Konsole, die AWS-Services API oder verwenden. AWS CLI AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung im Ruhezustand

AWS FIS verschlüsselt Ihre Daten im Ruhezustand immer. Daten in AWS FIS werden im Ruhezustand mit transparenter serverseitiger Verschlüsselung verschlüsselt. Dieser Service reduziert den Ausführungsaufwand und die Komplexität, die mit dem Schutz sensibler Daten verbunden sind. Mit der Verschlüsselung von Daten im Ruhezustand können Sie sicherheitsrelevante Anwendungen erstellen, die Verschlüsselungsvorschriften und gesetzliche Bestimmungen einhalten.

Verschlüsselung während der Übertragung

AWS FIS verschlüsselt Daten, die zwischen dem Dienst und anderen integrierten Diensten übertragen werden. AWS Alle Daten, die zwischen AWS FIS und integrierten Diensten übertragen werden, werden mit Transport Layer Security (TLS) verschlüsselt. Weitere Informationen zu anderen integrierten AWS Diensten finden Sie unter [Unterstützt AWS-Services](#).

Identitäts- und Zugriffsmanagement für den AWS Fault Injection Service

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um FIS-Ressourcen zu verwenden AWS . IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Inhalt

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So funktioniert der AWS Fault Injection Service mit IAM](#)
- [AWS Beispiele für Richtlinien für den Fault Injection Service](#)
- [Verwenden Sie serviceverknüpfte Rollen für AWS den Fault Injection Service](#)
- [AWS verwaltete Richtlinien für den AWS Fault Injection Service](#)

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in AWS FIS ausführen.

Dienstbenutzer — Wenn Sie den AWS FIS-Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr AWS FIS-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für die AWS FIS-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff AWS auf FIS. Es ist Ihre Aufgabe, zu bestimmen, auf welche AWS FIS-Funktionen und -Ressourcen Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die Berechtigungen der Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen.

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf FIS schreiben können. AWS

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 ein AWS-Konto erstellen, 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 den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. 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 AWS Identitäten oder Ressourcen anhängen. 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 Richtlinienarten

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die sich daraus ergebenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinienarten betroffen sind, finden Sie unter [Bewertungslogik für Richtlinien](#) im IAM-Benutzerhandbuch.

So funktioniert der AWS Fault Injection Service mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf AWS FIS verwenden, sollten Sie sich darüber informieren, welche IAM-Funktionen mit FIS verwendet werden können. AWS

IAM-Funktionen, die Sie mit dem Fault Injection Service verwenden können AWS

IAM-Feature	AWS FIS-Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Ja

Einen allgemeinen Überblick darüber, wie AWS FIS und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im [AWS IAM-Benutzerhandbuch unter Dienste, die mit IAM funktionieren](#).

Identitätsbasierte Richtlinien für FIS AWS

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 Maßnahmen für FIS AWS

Beispiele für identitätsbasierte AWS FIS-Richtlinien finden Sie unter [AWS Beispiele für Richtlinien für den Fault Injection Service](#)

Ressourcenbasierte Richtlinien innerhalb von FIS AWS

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.

Politische Maßnahmen für FIS AWS

Unterstützt Richtlinienaktionen: 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 `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 AWS FIS-Aktionen finden Sie unter Vom [AWS Fault Injection Service definierte Aktionen in der Serviceautorisierungsreferenz](#).

Richtlinienaktionen in AWS FIS verwenden vor der Aktion das folgende Präfix:

```
fis
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
    "fis:action1",  
    "fis:action2"  
]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `List` beginnen, einschließlich der folgenden Aktion:

```
"Action": "fis:List*"
```

Politische Ressourcen für AWS FIS

Unterstützt Richtlinienressourcen: 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 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": "*" 
```

Einige AWS FIS-API-Aktionen unterstützen mehrere Ressourcen. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie sie ARNs durch Kommas.

```
"Resource": [  
    "resource1",
```

```
"resource2"
```

```
]
```

Eine Liste der AWS FIS-Ressourcentypen und ihrer Eigenschaften finden Sie unter Ressourcentypen ARNs, die [durch den AWS Fault Injection Service definiert wurden in der Service](#) Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Vom [AWS Fault Injection Service definierte Aktionen](#).

Schlüssel zur Richtlinienbedingung für AWS FIS

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 AWS FIS-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für den AWS Fault Injection Service in der Service](#) Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Vom [AWS Fault Injection Service definierte Aktionen](#).

Beispiele für identitätsbasierte AWS FIS-Richtlinien finden Sie unter. [AWS Beispiele für Richtlinien für den Fault Injection Service](#)

ACLs AWS in FIS

Unterstützt ACLs: Nein

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.

ABAC mit FIS AWS

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, um Operationen zu ermöglichen, 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.

Ein Beispiel für eine identitätsbasierte Richtlinie zur Beschränkung des Zugriffs auf eine Ressource anhand der Tags für diese Ressource finden Sie unter. [Beispiel: Verwenden Sie Tags, um die Ressourcennutzung zu kontrollieren](#)

Temporäre Anmeldeinformationen mit FIS verwenden AWS

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie den Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Anmeldeinformationen in IAM](#) und [AWS-Services , die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipalberechtigungen für AWS FIS

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für AWS FIS

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.

Servicebezogene Rollen für FIS AWS

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.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen AWS FIS-Rollen finden Sie unter [Verwenden Sie serviceverknüpfte Rollen für AWS den Fault Injection Service](#)

AWS Beispiele für Richtlinien für den Fault Injection Service

Standardmäßig sind Benutzer und Rollen nicht berechtigt, AWS FIS-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 den von AWS FIS definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für den AWS Fault Injection Service in der Serviceautorisierungsreferenz](#).

Inhalt

- [Best Practices für Richtlinien](#)
- [Beispiel: Verwenden Sie die FIS-Konsole AWS](#)
- [Beispiel: Listet die verfügbaren AWS FIS-Aktionen auf](#)

- [Beispiel: Erstellen Sie eine Versuchsvorlage für eine bestimmte Aktion](#)
- [Beispiel: Starten Sie ein Experiment](#)
- [Beispiel: Verwenden Sie Tags, um die Ressourcennutzung zu kontrollieren](#)
- [Beispiel: Löschen Sie eine Experimentvorlage mit einem bestimmten Tag](#)
- [Beispiel: Erteilen der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Beispiel: Verwenden Sie Bedingungsschlüssel für ec2:InjectApiError](#)
- [Beispiel: Verwenden Sie Bedingungsschlüssel für aws:s3:bucket-pause-replication](#)

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand AWS FIS-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese 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.

Beispiel: Verwenden Sie die FIS-Konsole AWS

Um auf die AWS Fault Injection Service-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den AWS FIS-Ressourcen in Ihrem AWS-Konto aufzulisten und einzusehen. 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.

Die folgende Beispielrichtlinie gewährt die Erlaubnis, alle AWS FIS-Ressourcen mithilfe AWS der FIS-Konsole aufzulisten und anzuzeigen, sie jedoch nicht zu erstellen, zu aktualisieren oder zu löschen. Sie gewährt auch Berechtigungen zum Anzeigen der verfügbaren Ressourcen, die von allen AWS FIS-Aktionen verwendet werden, die Sie in einer Experimentvorlage angeben könnten.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FISReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "fis:List*",
        "fis:Get*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AdditionalReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "ssm:Describe*",
        "ssm:Get*",
        "ssm:List*",
        "ec2:DescribeInstances",
        "rds:DescribeDBClusters",
        "ecs:DescribeClusters",
        "ecs:ListContainerInstances",
        "eks:DescribeNodegroup",
        "cloudwatch:DescribeAlarms",
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PermissionsToCreateServiceLinkedRole",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "fis.amazonaws.com"
        }
      }
    }
  ]
}
```

```
}
```

Beispiel: Listet die verfügbaren AWS FIS-Aktionen auf

Die folgende Richtlinie erteilt die Erlaubnis, die verfügbaren AWS FIS-Aktionen aufzulisten.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "fis:ListActions"
      ],
      "Resource": "arn:aws:fis:*:*:action/*"
    }
  ]
}
```

Beispiel: Erstellen Sie eine Versuchsvorlage für eine bestimmte Aktion

Die folgende Richtlinie erteilt die Erlaubnis, eine Experimentvorlage für die Aktion zu erstellen `aws:ec2:stop-instances`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
        "fis:CreateExperimentTemplate"
      ],
      "Resource": [
        "arn:aws:fis:*:*:action/aws:ec2:stop-instances",

```

```

        "arn:aws:fis:*:experiment-template/*"
    ],
    {
        "Sid": "PolicyPassRoleExample",
        "Effect": "Allow",
        "Action": [
            "iam:PassRole"
        ],
        "Resource": [
            "arn:aws:iam::111122223333:role/role-name"
        ]
    }
]
}

```

Beispiel: Starten Sie ein Experiment

Die folgende Richtlinie erteilt die Erlaubnis, ein Experiment mit der angegebenen IAM-Rolle und der angegebenen Experimentvorlage zu starten. Sie ermöglicht es AWS FIS auch, im Namen des Benutzers eine dienstbezogene Rolle zu erstellen. Weitere Informationen finden Sie unter [Verwenden Sie serviceverknüpfte Rollen für AWS den Fault Injection Service](#).

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
        "fis:StartExperiment"
      ],
      "Resource": [
        "arn:aws:fis:*:experiment-template/experiment-template-id",
        "arn:aws:fis:*:experiment/*"
      ]
    },
    {
      "Sid": "PolicyExampleforServiceLinkedRole",

```

```
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:AWSServiceName": "fis.amazonaws.com"
      }
    }
  ]
}
```

Beispiel: Verwenden Sie Tags, um die Ressourcennutzung zu kontrollieren

Die folgende Richtlinie gewährt die Erlaubnis, Experimente anhand von Experimentvorlagen auszuführen, die das Tag enthalten `Purpose=Test`. Sie gewährt keine Erlaubnis, Versuchsvorlagen zu erstellen oder zu ändern oder Experimente mit Vorlagen durchzuführen, die nicht über das angegebene Tag verfügen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "fis:StartExperiment",
      "Resource": "arn:aws:fis:*:*:experiment-template/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Test"
        }
      }
    }
  ]
}
```

Beispiel: Löschen Sie eine Experimentvorlage mit einem bestimmten Tag

Die folgende Richtlinie gewährt die Erlaubnis, eine Experimentvorlage mit Tag zu löschen `Purpose=Test`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "fis:DeleteExperimentTemplate"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Test"
        }
      }
    }
  ]
}
```

Beispiel: Erteilen 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 API AWS CLI oder AWS .

```
{
  "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": "*"
}
]
}

```

Beispiel: Verwenden Sie Bedingungsschlüssel für **ec2:InjectApiError**

In der folgenden Beispielrichtlinie wird der `ec2:FisTargetArns` Bedingungsschlüssel verwendet, um Zielressourcen einzugrenzen. Diese Richtlinie ermöglicht die AWS FIS-Aktionen `aws:ec2:api-insufficient-instance-capacity-error` und `aws:ec2:asg-insufficient-instance-capacity-error`.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:InjectApiError",
      "Resource": "*",

```

```

    "Condition": {
      "StringEquals": {
        "ec2:FisActionId": [
          "aws:ec2:api-insufficient-instance-capacity-error",
          "aws:ec2:asg-insufficient-instance-capacity-error"
        ]
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:InjectApiError",
      "Resource": "*",
      "Condition": {
        "ForAllValues:ArnLike": {
          "ec2:FisTargetArns": [
            "arn:aws:autoscaling:*:*:autoScalingGroup:uuid:autoScalingGroupName/asg-name"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "autoscaling:DescribeAutoScalingGroups",
      "Resource": "*"
    }
  ]
}

```

Beispiel: Verwenden Sie Bedingungsschlüssel für **aws:s3:bucket-pause-replication**

Die folgende Beispielrichtlinie verwendet den S3:IsReplicationPauseRequest Bedingungsschlüssel zum Zulassen PutReplicationConfiguration und GetReplicationConfiguration nur, wenn er von AWS FIS im Kontext der AWS FIS-Aktion verwendet wird. `aws:s3:bucket-pause-replication`

JSON

```
{
```



```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "S3:PauseReplication"
    ],
    "Resource": "arn:aws:s3:::mybucket",
    "Condition": {
      "StringEquals": {
        "s3:DestinationRegion": "region"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "S3:PutReplicationConfiguration",
      "S3:GetReplicationConfiguration"
    ],
    "Resource": "arn:aws:s3:::mybucket",
    "Condition": {
      "BoolIfExists": {
        "s3:IsReplicationPauseRequest": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "S3:ListBucket"
    ],
    "Resource": "arn:aws:s3:::*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "tag:GetResources"
    ],
    "Resource": "*"
  }
]
```

Verwenden Sie serviceverknüpfte Rollen für AWS den Fault Injection Service

AWS [Der Fault Injection Service verwendet AWS Identity and Access Management dienstgebundene Rollen \(IAM\)](#). Eine serviceverknüpfte Rolle ist ein einzigartiger Typ von IAM-Rolle, die direkt mit FIS verknüpft ist. AWS Servicebezogene Rollen sind von AWS FIS vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine dienstbezogene Rolle erleichtert die Einrichtung von AWS FIS, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen, um die Überwachung und die Ressourcenauswahl für Experimente zu verwalten. AWS FIS definiert die Berechtigungen seiner dienstbezogenen Rollen, und sofern nicht anders definiert, kann nur AWS FIS seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Zusätzlich zu der dienstbezogenen Rolle müssen Sie auch eine IAM-Rolle angeben, die die Erlaubnis erteilt, die Ressourcen zu ändern, die Sie in einer Experimentvorlage als Ziele angeben. Weitere Informationen finden Sie unter [IAM-Rollen für AWS FIS-Experimente](#).

Sie können eine serviceverknüpfte Rolle nur löschen, nachdem Sie zuvor die zugehörigen Ressourcen gelöscht haben. Dadurch werden Ihre AWS FIS-Ressourcen geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Dienstbezogene Rollenberechtigungen für FIS AWS

AWS FIS verwendet die dienstbezogene Rolle namens `AWSServiceRoleForFIS`, um die Überwachung und die Ressourcenauswahl für Experimente zu verwalten.

Die dienstbezogene `AWSServiceRoleForFIS`-Rolle vertraut darauf, dass die folgenden Dienste diese Rolle übernehmen:

- `fis.amazonaws.com`

Die serviceverknüpfte `AWSServiceRoleForFIS`-Rolle verwendet die verwaltete Richtlinie `Amazon.FISServiceRolePolicy`. Diese Richtlinie ermöglicht es der AWS FIS, die Überwachung und die Auswahl der Ressourcen für Experimente zu verwalten. Weitere Informationen finden Sie unter [Amazon FISService RolePolicy](#) in der AWS Managed Policy Reference.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Damit die dienstverknüpfte AWSServiceRoleForFIS-Rolle erfolgreich erstellt werden kann, muss die IAM-Identität, mit der Sie AWS FIS verwenden, über die erforderlichen Berechtigungen verfügen. Um die erforderlichen Berechtigungen zu erteilen, fügen Sie die folgende Richtlinie an die IAM-Identität an.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "fis.amazonaws.com"
        }
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

Erstellen Sie eine serviceverknüpfte Rolle für FIS AWS

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie ein AWS FIS-Experiment in der AWS-Managementkonsole, der oder der AWS API starten AWS CLI, erstellt AWS FIS die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie ein FIS-Experiment starten, erstellt AWS FIS die AWS serviceverknüpfte Rolle erneut für Sie.

Bearbeiten Sie eine serviceverknüpfte Rolle für FIS AWS

AWS FIS erlaubt es Ihnen nicht, die dienstbezogene AWSServiceRoleForFIS-Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Löschen Sie eine serviceverknüpfte Rolle für FIS AWS

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Note

Wenn der AWS FIS-Dienst die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu bereinigen, schlägt die Bereinigung möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um die von der FIS verwendeten AWS FIS-Ressourcen zu bereinigen AWSService RoleFor

Stellen Sie sicher, dass derzeit keines Ihrer Experimente läuft. Falls nötig, beenden Sie Ihre Experimente. Weitere Informationen finden Sie unter [Stoppen eines Experiments](#).

So löschen Sie die -servicegebundene Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die mit dem AWSServiceRoleForFIS-Dienst verknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Unterstützte Regionen für dienstverknüpfte FIS-Rollen AWS

AWS FIS unterstützt die Verwendung von dienstbezogenen Rollen in allen Regionen, in denen der Dienst verfügbar ist. Weitere Informationen finden Sie unter [Endpunkte und Kontingente für den AWS Fault Injection-Dienst](#).

AWS verwaltete Richtlinien für den AWS Fault Injection Service

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet AWS wird. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [vom Kunden verwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: Amazon FISService RolePolicy

Diese Richtlinie ist mit der dienstleistungsbezogenen Rolle FIS verknüpft, sodass AWSServiceRoleFor AWS FIS die Überwachung und die Auswahl der Ressourcen für Experimente verwalten kann. Weitere Informationen finden Sie unter [Verwenden Sie serviceverknüpfte Rollen für AWS den Fault Injection Service](#).

AWS verwaltete Richtlinie: Zugriff AWSFault InjectionSimulator EC2

Verwenden Sie diese Richtlinie in einer Experimentierrolle, um AWS FIS die Erlaubnis zu erteilen, Experimente durchzuführen, die die [AWS FIS-Aktionen für Amazon EC2](#) verwenden. Weitere Informationen finden Sie unter [the section called “Die Rolle des Experiments”](#).

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSFaultInjectionSimulatorEC2Zugriff](#) in der Referenz für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSFault InjectionSimulator ECSAccess

Verwenden Sie diese Richtlinie in einer Experimentierrolle, um AWS FIS die Erlaubnis zu erteilen, Experimente durchzuführen, die die [AWS FIS-Aktionen für Amazon ECS](#) verwenden. Weitere Informationen finden Sie unter [the section called “Die Rolle des Experiments”](#).

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSFaultInjectionSimulatorECSAccess](#) in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSFault InjectionSimulator EKSAccess

Verwenden Sie diese Richtlinie in einer Experimentierrolle, um AWS FIS die Erlaubnis zu erteilen, Experimente durchzuführen, die die [AWS FIS-Aktionen für Amazon EKS](#) verwenden. Weitere Informationen finden Sie unter [the section called “Die Rolle des Experiments”](#).

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSFaultInjectionSimulatorEKSAccess](#) in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSFault InjectionSimulatorNetworkAccess

Verwenden Sie diese Richtlinie in einer experimentellen Rolle, um der AWS FIS die Erlaubnis zu erteilen, Experimente durchzuführen, bei denen die [AWS FIS-Netzwerkaktionen](#) verwendet werden. Weitere Informationen finden Sie unter [the section called “Die Rolle des Experiments”](#).

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSFaultInjectionSimulatorNetworkAccess](#) in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSFault InjectionSimulator RDSAccess

Verwenden Sie diese Richtlinie in einer Experimentierrolle, um AWS FIS die Erlaubnis zu erteilen, Experimente durchzuführen, die die [AWS FIS-Aktionen für Amazon RDS](#) verwenden. Weitere Informationen finden Sie unter [the section called “Die Rolle des Experiments”](#).

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSFaultInjectionSimulatorRDSAccess](#) in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSFault InjectionSimulator SSMAccess

Verwenden Sie diese Richtlinie in einer Experimentrolle, um AWS FIS die Erlaubnis zu erteilen, Experimente durchzuführen, die die [AWS FIS-Aktionen für Systems Manager](#) verwenden. Weitere Informationen finden Sie unter [the section called “Die Rolle des Experiments”](#).

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSFaultInjectionSimulatorSSMAccess](#) in der Referenz zu von AWS verwalteten Richtlinien.

AWS FIS aktualisiert verwaltete Richtlinien AWS

Hier finden Sie Informationen zu Aktualisierungen der AWS verwalteten Richtlinien für AWS FIS, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen.

Änderungen	Beschreibung	Date
AWSFaultInjectionSimulatorEC2Zugriff — Aktualisierung einer bestehenden Richtlinie	Für die Szenarien „AZ: Application Slowdown“ und „Cross-AZ: Traffic Slowdown“ wurde die erforderliche Berechtigung hinzugefügt. Die Berechtigungen lauten: ec2: DescribeSubnets	12. November 2025
AWSFaultInjectionSimulatorEC2Access – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die für die Szenarien „AZ: Application Slowdown“ und „Cross-AZ: Traffic Slowdown“ erforderlich sind. Die Berechtigungen sind: ecs:DescribeContainerInstances, ec2: und ec2: DescribeSubnets DescribeInstances	12. November 2025
AWSFaultInjectionSimulatorEC2Access – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, ECS-Ziele zu lösen.	25. Januar 2024
AWSFaultInjectionSimulatorNetworkAccess – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, Experimente mit den Aktionen aws:network:route-table-disrupt-cross-region-connectivity und aws:network:transit-gateway-disrupt-cross-region-connectivity durchzuführen.	25. Januar 2024
AWSFaultInjectionSimulatorEC2Zugriff — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, damit AWS FIS EC2-Instances auflösen kann.	13. November 2023

Änderungen	Beschreibung	Date
AWSFaultInjectionSimulatorEKSAccess – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, EKS-Ziele aufzulösen.	13. November 2023
AWSFaultInjectionSimulatorRDSAccess – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, RDS-Ziele aufzulösen.	13. November 2023
AWSFaultInjectionSimulatorEC2Zugriff — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, SSM-Dokumente auf EC2-Instances auszuführen und EC2-Instances zu beenden.	02. Juni 2023
AWSFaultInjectionSimulatorSSMAccess – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, SSM-Dokumente auf EC2-Instances auszuführen.	02. Juni 2023
AWSFaultInjectionSimulatorECSAccess – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, Experimente mit den neuen Aktionen durchzuführen. <code>aws:ecs:task</code>	01. Juni 2023
AWSFaultInjectionSimulatorEKSAccess – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, Experimente mit den neuen <code>aws:eks:pod</code> Aktionen durchzuführen.	01. Juni 2023
AWSFaultInjectionSimulatorEC2Zugriff — Neue Richtlinie	Es wurde eine Richtlinie hinzugefügt, die es AWS FIS ermöglicht, ein Experiment durchzuführen, das AWS FIS-Aktionen für Amazon EC2 verwendet.	26. Oktober 2022

Änderungen	Beschreibung	Date
AWSFaultInjectionSimulatorECSAccess – Neue Richtlinie	Es wurde eine Richtlinie hinzugefügt, die es AWS FIS ermöglicht, ein Experiment durchzuführen, das AWS FIS-Aktionen für Amazon ECS verwendet.	26. Oktober 2022
AWSFaultInjectionSimulatorEKSAccess – Neue Richtlinie	Es wurde eine Richtlinie hinzugefügt, die es AWS FIS ermöglicht, ein Experiment durchzuführen, das AWS FIS-Aktionen für Amazon EKS verwendet.	26. Oktober 2022
AWSFaultInjectionSimulatorNetworkAccess – Neue Richtlinie	Es wurde eine Richtlinie hinzugefügt, die es AWS FIS ermöglicht, ein Experiment durchzuführen, das AWS FIS-Netzwerkaktionen verwendet.	26. Oktober 2022
AWSFaultInjectionSimulatorRDSAccess – Neue Richtlinie	Es wurde eine Richtlinie hinzugefügt, die es AWS FIS ermöglicht, ein Experiment durchzuführen, das AWS FIS-Aktionen für Amazon RDS verwendet.	26. Oktober 2022
AWSFaultInjectionSimulatorSMSAccess – Neue Richtlinie	Es wurde eine Richtlinie hinzugefügt, die AWS es FIS ermöglicht, ein Experiment durchzuführen, das AWS FIS-Aktionen für Systems Manager verwendet.	26. Oktober 2022
Amazon FISService RolePolicy — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, Subnetze zu beschreiben.	26. Oktober 2022
Amazon FISService RolePolicy — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, EKS-Cluster zu beschreiben.	7. Juli 2022

Änderungen	Beschreibung	Date
Amazon FISService RolePolicy — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, damit AWS FIS die Aufgaben in Ihren Clustern auflisten und beschreiben kann.	7. Februar 2022
Amazon FISService RolePolicy — Aktualisierung einer bestehenden Richtlinie	Die <code>events:ManagedBy</code> Bedingung für die <code>events:DescribeRule</code> Aktion wurde entfernt.	6. Januar 2022
Amazon FISService RolePolicy — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, den Verlauf der CloudWatch Alarmläufe abzurufen, die bei Stopp-Bedingungen verwendet wurden.	30. Juni 2021
AWS FIS hat begonnen, Änderungen zu verfolgen	AWS FIS begann, Änderungen an ihren AWS verwalteten Richtlinien zu verfolgen	1. März 2021

Sicherheit der Infrastruktur im AWS Fault Injection Service

Als verwalteter Dienst ist der AWS Fault Injection Service durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf AWS FIS zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Greifen Sie über einen VPC-Endpunkt () mit einer Schnittstelle auf AWS FIS zu AWS PrivateLink

Sie können eine private Verbindung zwischen Ihrer VPC und dem AWS Fault Injection Service herstellen, indem Sie einen VPC-Schnittstellen-Endpunkt erstellen. VPC-Endgeräte werden mit einer Technologie betrieben [AWS PrivateLink](#), mit der Sie privat auf AWS FIS zugreifen können, APIs ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine AWS Direct Connect-Verbindung zu benötigen. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit AWS FIS APIs zu kommunizieren.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere [Elastic Network-Schnittstellen](#) in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie AWS PrivateLink im Leitfaden unter [Zugriff AWS-Services durch](#).AWS PrivateLink

Überlegungen zu AWS FIS VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für AWS FIS einrichten, lesen Sie im Handbuch [Zugriff und AWS-Service Verwendung eines Schnittstellen-VPC-Endpunkts](#) nach.AWS PrivateLink

AWS FIS unterstützt Aufrufe aller API-Aktionen von Ihrer VPC aus.

Erstellen Sie einen VPC-Schnittstellen-Endpunkt für AWS FIS

Sie können einen VPC-Endpunkt für den AWS FIS-Service entweder mit der Amazon VPC-Konsole oder mit () erstellen. AWS Command Line Interface AWS CLI Weitere Informationen finden Sie unter [Erstellen eines VPC-Endpunkts](#) im AWS PrivateLink -Leitfaden.

Erstellen Sie einen VPC-Endpunkt für AWS FIS mit dem folgenden Dienstenamen:

`com.amazonaws.region.fis`

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen an AWS FIS stellen, indem Sie den Standard-DNS-Namen für die Region verwenden, z. B. `fis.us-east-1.amazonaws.com`

Erstellen Sie eine VPC-Endpunkttrichtlinie für AWS FIS

Sie können Ihrem VPC-Endpunkt eine Endpunkttrichtlinie hinzufügen, die den Zugriff auf AWS FIS steuert. Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter [Kontrolle des Zugriffs auf VPC-Endpunkte mit Endpunktrichtlinien](#) im AWS PrivateLink Leitfaden.

Beispiel: VPC-Endpunktrichtlinie für bestimmte AWS FIS-Aktionen

Die folgende VPC-Endpunktrichtlinie gewährt allen Prinzipalen Zugriff auf die aufgelisteten AWS FIS-Aktionen für alle Ressourcen.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "fis:ListExperimentTemplates",
        "fis:StartExperiment",
        "fis:StopExperiment",
        "fis:GetExperiment"
      ],
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

Beispiel: VPC-Endpunktrichtlinie, die den Zugriff von einem bestimmten AWS-Konto

Die folgende VPC-Endpunktrichtlinie verweigert den angegebenen AWS-Konto Zugriff auf alle Aktionen und Ressourcen, gewährt jedoch allen anderen AWS-Konten Zugriff auf alle Aktionen und Ressourcen.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

```
    },  
    {  
      "Effect": "Deny",  
      "Action": "*",  
      "Resource": "*",  
      "Principal": {  
        "AWS": [ "123456789012" ]  
      }  
    }  
  ]  
}
```

Verschlagworten Sie Ihre AWS FIS-Ressourcen

Ein Tag ist ein Metadaten-Label, das entweder Sie oder Sie einer AWS Ressource AWS zuweisen. Jedes Tag besteht aus einem Schlüssel und einem Wert. Für Tags, die Sie zuweisen, definieren Sie einen Schlüssel und einen Wert. Sie können beispielsweise den Schlüssel als `purpose` und den Wert als `test` für eine Ressource definieren.

Tags sind für folgende Aktivitäten nützlich:

- Identifizieren und organisieren Sie Ihre AWS Ressourcen. Viele AWS Dienste unterstützen Tagging, sodass Sie Ressourcen aus verschiedenen Diensten dasselbe Tag zuweisen können, um anzuzeigen, dass die Ressourcen miteinander verknüpft sind.
- Kontrollieren Sie den Zugriff auf Ihre AWS Ressourcen. Weitere Informationen finden Sie unter [Zugriffssteuerung mit Tags](#) im -IAM-Benutzerhandbuch.

Tagging-Einschränkungen

Die folgenden grundlegenden Einschränkungen gelten für Tags auf AWS FIS-Ressourcen:

- Maximale Anzahl von Tags, die Sie einer Ressource zuweisen können: 50
- Maximale Schlüssellänge: 128 Unicode-Zeichen
- Maximale Wertlänge: 256 Unicode-Zeichen
- Gültige Zeichen für Schlüssel und Werte: a-z, A-Z, 0-9, Leerzeichen und die folgenden Zeichen: `_` `:` `/` `=` `+` `-` und `@`
- Schlüssel und Werte unterscheiden zwischen Groß- und Kleinschreibung.
- Sie können es nicht `aws :` als Präfix für Schlüssel verwenden, da es für die Verwendung reserviert ist AWS

Arbeite mit Tags

Die folgenden Ressourcen des AWS AWS Fault Injection Service (FIS) unterstützen Tagging:

- Aktionen
- Experimente
- Experimentvorlagen

Sie können die Konsole verwenden, um mit Tags für Experimente und Versuchsvorlagen zu arbeiten. Weitere Informationen finden Sie hier:

- [Kennzeichnen Sie ein Experiment](#)
- [Versuchs-Vorlagen mit Tags versehen](#)

Sie können die folgenden AWS CLI Befehle verwenden, um mit Tags für Aktionen, Experimente und Experimentvorlagen zu arbeiten:

- [tag-resource](#) — Fügt einer Ressource Tags hinzu.
- [untag-resource](#) — Entferne Tags aus einer Ressource.
- [list-tags-for-resource](#)— Listet die Tags für eine bestimmte Ressource auf.

Kontingente und Einschränkungen für den AWS Fault Injection Service

Ihr AWS-Konto hat Standardkontingente, früher als Limits bezeichnet, für jeden AWS Dienst. Sofern nicht anders angegeben, ist jedes Kontingent regionsspezifisch. Sie können Erhöhungen für die Kontingente beantragen, die in der folgenden Tabelle als anpassbar gekennzeichnet sind.

Um die Kontingente für AWS FIS in Ihrem Konto einzusehen, öffnen Sie die [Konsole Service Quotas](#). Wählen Sie im Navigationsbereich AWS Dienste und dann AWS Fault Injection Service aus. Werte bis einschließlich der automatisch genehmigten Kontingente werden sofort angewendet. Die automatisch genehmigten Kontingente sind in der Beschreibungsspalte der folgenden Tabelle aufgeführt. Wenn Sie Kontingente benötigen, die die automatisch genehmigten Limits überschreiten, reichen Sie bitte eine Anfrage ein. Werte, die über den automatisch genehmigten Grenzwerten liegen, werden vom Kundensupport überprüft und nach Möglichkeit genehmigt.

Informationen zur Erhöhung eines Kontingents finden Sie unter [Anfordern einer Kontingenterhöhung](#) im Service-Quotas-Benutzerhandbuch.

Ihr AWS-Konto hat die folgenden Kontingente in Bezug auf AWS FIS.

Name	Standard	Anpas	Description
Aktionsdauer in Stunden	Jede unterstützte Region: 12	Nein	Die maximale Anzahl von Stunden, für die das Ausführen einer Aktion in diesem Konto in der aktuellen Region zulässig ist.
Aktionsminuten pro globaler MRSC-Tabelle (Multiregion Strongly Consistent) für aws:dynamodb: action global-table-pause-replication	Jede unterstützte Region: 5.040	Ja	Die maximale Gesamtzahl der Aktionsminuten pro globaler MRSC-Tabelle, auf die aws:dynamodb: global-table-pause-replication innerhalb eines fortlaufenden

Name	Standard	Anpas	Description
			Zeitfensters von 7 Tagen abzielen kann.
Aktionen pro Experimentvorlage	Jede unterstützte Region: 20	Nein	Die maximale Anzahl von Aktionen, die Sie in einer Experimentvorlage in diesem Konto in der aktuellen Region erstellen können.
Aktive Experimente	Jede unterstützte Region: 5	Nein	Die maximale Anzahl von aktiven Experimenten, die Sie in diesem Konto in der aktuellen Region gleichzeitig ausführen können.
Datenaufbewahrung abgeschlossener Experimente in Tagen	Jede unterstützte Region: 120	Nein	Die maximale Anzahl von Tagen, die der AWS FIS zur Aufbewahrung von Daten über abgeschlossene Experimente auf diesem Konto in der aktuellen Region zur Verfügung stehen.
Experimentdauer in Stunden	Jede unterstützte Region: 12	Nein	Die maximale Anzahl von Stunden, für die das Ausführen eines Experiments in diesem Konto in der aktuellen Region zulässig ist.

Name	Standard	Anpas	Description
Experimentvorlagen	Jede unterstützte Region: 500	Nein	Die maximale Anzahl von Experimentvorlagen, die Sie in diesem Konto in der aktuellen Region erstellen können.
Maximale Anzahl verwalteter Präfixlisten in aws:network: -region-connectivity route-table-disrupt-cross	Jede unterstützte Region: 15	Nein	Die maximale Anzahl verwalteter Präfixlisten, die aws:network: route-table-disrupt-cross -region-connectivity pro Aktion zulässt.
Maximale Anzahl von Routing-Tabellen in aws:network: -region-connectivity route-table-disrupt-cross	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von Routing-Tabellen, die aws:network: route-table-disrupt-cross -region-connectivity pro Aktion zulässt.
Maximale Anzahl von Routen in aws:network: -region-connectivity route-table-disrupt-cross	Jede unterstützte Region: 200	Nein	Die maximale Anzahl von Routen, die aws:network: route-table-disrupt-cross -region-connectivity pro Aktion zulässt.
Parallele Aktionen pro Experiment	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von Aktionen, die Sie in einem Experiment in diesem Konto in der aktuellen Region parallel ausführen können.

Name	Standard	Anpas	Description
Stoppbedingungen pro Experimentvorlage	Jede unterstützte Region: 5	Nein	Die maximale Anzahl von Stopp-Bedingungen, die Sie in diesem Konto in der aktuellen Region zu einer Experimentvorlage hinzufügen können.
Ziel-Aurora-DSQL-Cluster für aws:dsql: action. cluster-connection-failure	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl von Aurora-DSQL-Clustern, auf die aws:dsql: abzielen cluster-connection-failure kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Auto Scaling Scaling-Zielgruppen für aws:ec2: -error asg-insufficient-instance-capacity	Jede unterstützte Region: 500	Ja	Die maximale Anzahl von Auto Scaling Scaling-Gruppen, auf die aws:ec2: asg-insufficient-instance-capacity -error abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-Buckets für aws:s3: bucket-pause-replication	Jede unterstützte Region: 25	Ja	Die maximale Anzahl von S3-Buckets, auf die aws:s3: abzielen bucket-pause-replication kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.

Name	Standard	Anpas	Description
Zielcluster für aws:ecs: drain-container-instances	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl von Clustern, auf die aws:ecs: abzielen drain-container-instances kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Zielcluster für aws:rds: failover-db-cluster	Jede unterstützte Region: 160	Ja	Die maximale Anzahl von Clustern, auf die aws:rds: abzielen failover-db-cluster kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel DBInstances für aws:rds: reboot-db-instances	Jede unterstützte Region: 130	Ja	Die maximale Anzahl DBInstances , auf die aws:rds: pro reboot-db-instances Experiment abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren.
Ziel-Instances für aws:ec2:reboot-instances	Jede unterstützte Region: 600	Ja	Die maximale Anzahl von Instances, auf die aws:ec2:reboot-instances abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.

Name	Standard	Anpas	Description
Ziel-Instances für aws:ec2:stop-instances	Jede unterstützte Region: 400	Ja	Die maximale Anzahl von Instances, auf die aws:ec2:stop-instances abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-Instances für aws:ec2:terminate-instances	Jede unterstützte Region: 300	Ja	Die maximale Anzahl von Instances, auf die aws:ec2:terminate-instances abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-Instances für aws:ssm:send-command	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Instances, auf die aws:ssm:send-command abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-Kinesis-Datenstreams für aws:kinesis: action. stream-expired-iterator-exception	Jede unterstützte Region: 280	Ja	Die maximale Anzahl von Kinesis-Datenströmen, auf die aws:kinesis: abzielen stream-expired-iterator-exception kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.

Name	Standard	Anpas	Description
Ziel-Kinesis-Datenstreams für aws:kinesis: action. stream-provisioned-throughput-exception	Jede unterstützte Region: 280	Ja	Die maximale Anzahl von Kinesis-Datenströmen, auf die aws:kinesis: abzielen stream-provisioned-throughput-exception kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel ManagedResources für aws:arc: action. start-zonal-autoshift	Jede unterstützte Region: 200	Ja	Die maximale Anzahl verwalteter Ressourcen, auf die aws:arc: pro Experiment abzielen start-zonal-autoshift kann, wenn Sie Ziele mithilfe von Tags identifizieren.
Zielknotengruppen für aws:eks: terminate-nodegroup-instances	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl von Nodegroups, auf die aws:eks: abzielen terminate-nodegroup-instances kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-Pods für aws:eks: pod-cpu-stress	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Pods, auf die aws:eks: abzielen pod-cpu-stress kann, wenn Sie Ziele anhand von Parametern identifizieren, pro Experiment.

Name	Standard	Anpas	Description
Ziel-Pods für aws:eks:pod-delete	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Pods, auf die aws:eks:pod-delete pro Experiment abzielen kann, wenn Sie Ziele mithilfe von Parametern identifizieren.
Ziel-Pods für aws:eks: pod-io-stress	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Pods, auf die aws:eks: abzielen pod-io-stress kann, wenn Sie Ziele anhand von Parametern identifizieren, pro Experiment.
Ziel-Pods für aws:eks: pod-memory-stress	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Pods, auf die aws:eks: abzielen pod-memory-stress kann, wenn Sie Ziele anhand von Parametern identifizieren, pro Experiment.
Ziel-Pods für aws:eks: pod-network-blackhole-port	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Pods, auf die aws:eks: abzielen pod-network-blackhole-port kann, wenn Sie Ziele anhand von Parametern identifizieren, pro Experiment.

Name	Standard	Anpas	Description
Ziel-Pods für aws:eks: pod-network-latency	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Pods, auf die aws:eks: abzielen pod-network-latency kann, wenn Sie Ziele anhand von Parametern identifizieren, pro Experiment.
Ziel-Pods für aws:eks: pod-network-packet-loss	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Pods, auf die aws:eks: abzielen pod-network-packet-loss kann, wenn Sie Ziele anhand von Parametern identifizieren, pro Experiment.
Ziel ReplicationGroups für aws:elasticache: — Veraltete Version geplant interrupt-cluster-az-power	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl ReplicationGroups , auf die aws:elasticache: pro Experiment abzielen interrupt-cluster-az-power kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren.

Name	Standard	Anpas	Description
Ziel für aws:elasticache: ReplicationGroups replicationgroup-interrupt-az-power	Jede unterstützte Region: 20	Ja	Die maximale Anzahl der aws:elasticache:, auf ReplicationGroups die pro Experiment abgezielt werden kann. replicationgroup-interrupt-az-power Für das Targeting gilt ein Tageslimit. ReplicationGroups Weitere Informationen finden Sie unter: https://docs.aws.amazon.com/fis/latest/userguide/fis-quotas.html .
Ziel SpotInstances für aws:ec2: send-spot-instance-interruptions	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl SpotInstances , auf die aws:ec2: pro Experiment abzielen send-spot-instance-interruptions kann, wenn Sie Ziele mithilfe von Tags identifizieren.

Name	Standard	Anpas	Description
Zielsubnetze für aws:network:disrupt-connectivity	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl von Subnetzen, auf die aws:network:disrupt-connectivity abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment. Kontingente über 5 gelten nur für den Parameter scope:all. Wenn Sie ein höheres Kontingent für einen anderen Bereichstyp benötigen, wenden Sie sich an den Kundensupport.
Zielsubnetze für aws:network: -region-connectivity route-table-disrupt-cross	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Subnetzen, auf die aws:network: route-table-disrupt-cross -region-connectivity abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Zielaufgaben für aws:ecs:stop-task	Jede unterstützte Region: 500	Ja	Die maximale Anzahl von Aufgaben, auf die aws:ecs:stop-task abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.

Name	Standard	Anpas	Description
Zielaufgaben für aws:ecs: task-cpu-stress	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Aufgaben, auf die aws:ecs: abzielen task-cpu-stress kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Zielaufgaben für aws:ecs: task-io-stress	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Aufgaben, auf die aws:ecs: abzielen task-io-stress kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Zielaufgaben für aws:ecs: task-kill-process	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Aufgaben, auf die aws:ecs: abzielen task-kill-process kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Zielaufgaben für aws:ecs: task-network-blackhole-port	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Aufgaben, auf die aws:ecs: abzielen task-network-blackhole-port kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.

Name	Standard	Anpas	Description
Zielaufgaben für aws:ecs: task-network-latency	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Aufgaben, auf die aws:ecs: abzielen task-network-latency kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Zielaufgaben für aws:ecs: task-network-packet-loss	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Aufgaben, auf die aws:ecs: abzielen task-network-packet-loss kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Ziel TransitGateways für aws:network: -region-connectivity transit-gateway-disrupt-cross	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Transit Gateways, auf die aws:network: transit-gateway-disrupt-cross -region-connectivity abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Virtuelle Zielschnittstellen für aws:directconnect:virtual-interface	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl virtueller Schnittstellen pro Experiment, auf die aws:directconnect:virtual-interface abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren.

Name	Standard	Anpas	Description
Ziel-Volumes für aws:ebs: pause-volume-io	Jede unterstützte Region: 160	Ja	Die maximale Anzahl von Volumes, auf die aws:ebs: pro Experiment abzielen pause-volume-io kann, wenn Sie Ziele mithilfe von Tags identifizieren.
Zielkontokonfigurationen pro Experimentvorlage	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von Zielkontokonfigurationen, die Sie für eine Versuchsvorlage in diesem Konto in der aktuellen Region erstellen können.
Zielfunktionen für aws:lambda: action. invocation-add-delay	Jede unterstützte Region: 140	Ja	Die maximale Anzahl von Lambda-Funktionen, auf die aws:lambda: abzielen invocation-add-delay kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Zielfunktionen für die Aktion aws:lambda: invocation-error.	Jede unterstützte Region: 140	Ja	Die maximale Anzahl von Lambda-Funktionen, auf die aws:lambda: invocation-error abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.

Name	Standard	Anpas	Description
invocation-http-integration-response Zielfunktionen für aws:lambda: action.	Jede unterstützte Region: 140	Ja	Die maximale Anzahl von Lambda-Funktionen, auf die aws:lambda: abzielen invocation-http-integration-response kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Zielcluster mit mehreren Regionen für die Aktion aws:memorydb: -replication ins Visier nehmen. multi-region-cluster-pause	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von MemoryDB-Clustern mit mehreren Regionen, auf die aws:memorydb: multi-region-cluster-pause-replication abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment. Wenn Sie ein höheres Kontingent benötigen, wenden Sie sich an den Kundensupport.
Zieltabellen für aws:dynamodb: action global-table-pause-replication	Jede unterstützte Region: 60	Ja	Die maximale Anzahl globaler Tabellen, auf die aws:dynamodb: pro Experiment abzielen kann. global-table-pause-replication

Ihre Nutzung von AWS FIS unterliegt den folgenden zusätzlichen Einschränkungen:

Name	Einschränkung
Tägliches Limit ReplicationGroups für Ziele für aws:elasticache: replicationgroup-interrupt-az-power	Das Limit liegt bei 20 Zielen ReplicationGroups pro Konto, Region und Tag. Sie können eine Erhöhung beantragen, indem Sie in der AWS Support Center-Konsole einen Support-Fall erstellen.

Dokumentverlauf

In der folgenden Tabelle werden wichtige Aktualisierungen der Dokumentation im AWSFault Injection Service-Benutzerhandbuch beschrieben.

Änderung	Beschreibung	Datum
Neue Aktion für AWS Direct Connect	Sie können die Aktion <code>aws:directconnect:virtual-interface</code> verwenden, um die Belastbarkeit der AWS Direct Connect Verbindung zu testen, indem Sie die Border Gateway Protocol-Sitzungen zwischen den lokalen Netzwerken und Peers, die den virtuellen Zielschnittstellen zugeordnet sind, vorübergehend unterbrechen.	18. Dezember 2025
Neue Szenarien	Sie können jetzt die neuen Szenarien „AZ: Application Slowdown“ und „Cross-AZ: Traffic Slowdown“ verwenden.	12. November 2025
Neuer Parameter für SSM-Dokumente	Die AWSFIS-RunSSM-Dokumente <code>-Network-Latency-SOURCES</code> und <code>-Network-Packet-Loss-SOURCE</code> unterstützen jetzt den <code>AWSFIS-RunFlowsPercent</code> -Parameter.	12. November 2025
Neuer Parameter für ECS- und EKS-Aktionen	Die Aktionen <code>aws:ecs:task-network-latency</code> , <code>aws:ecs:</code> , <code>aws:eks: task-network-packet-loss pod-network-latency</code> und	12. November 2025

	aws:eks: unterstützen jetzt den FlowsPercent-Parameter. pod-network-packet-loss	
AWSverwaltete Richtlinienaktualisierungen	Aktualisierter AWSFaultInjectionSimulatorEC2Zugriff auf verwaltete Richtlinien: ec2: DescribeSubnets -Berechtigung hinzugefügt.	12. November 2025
AWSverwaltete Richtlinienaktualisierungen	Aktualisierte verwaltete Richtlinie AWSFaultInjectionSimulatorECSAccess : Die Berechtigungen ecs: DescribeContainerInstances, ec2: DescribeSubnets und ec2: DescribeInstances wurden hinzugefügt.	12. November 2025
Neue Aktionen für Amazon Kinesis Data Streams	Sie können die aws:kinesis:stream-Aktionen verwenden.	15. Oktober 2025
Neues Experiment für Amazon EBS	Sie können die volume-io-latency Aktion aws:ebs: verwenden, um eine erhöhte I/O Latenz auf Ihren Amazon EBS-Volumes zu simulieren.	16. September 2025
Neuer Aktionsbereich AWS FIS	Sie können die Aktion aws:network:disrupt-connectivity verwenden, um die Konnektivität zu Ihren S3 Express One Zone-Verzeichnis-Buckets zu unterbrechen. Dieser Bereich ist jetzt auch im Szenario AZ-Verfügbarkeit: Stromunterbrechung enthalten.	18. August 2025

[MemoryDB-Unterstützung in FIS AWS](#)

Sie können AWS FIS verwenden, um zu testen, wie Ihre Anwendung mit Amazon MemoryDB Multi-Region-Clustern darauf reagiert, dass die Datenreplikation während einer regionsübergreifenden Netzwerkunterbrechung unterbrochen wird.

15. Juli 2025

[ARC-Unterstützung in FIS AWS](#)

Sie können AWS FIS verwenden, um zu testen, wie ARC Zonal Autoshift Ihre Anwendung während einer AZ-Stromunterbrechung automatisch wiederherstellt.

26. März 2025

[Neue Konfiguration des Experimentberichts](#)

Sie können AWS FIS jetzt aktivieren, um Berichte für Experimente zu generieren, die die Aktionen und Antworten der Experimente aus CloudWatch Dashboards zusammenfassen.

12. November 2024

[Neue Lambda-Aktionen](#)

Sie können jetzt `aws:lambda:function` Actions verwenden, um Fehler in Aufrufe Ihrer Lambda-Funktionen einzufügen.

31. Oktober 2024

[Neue Sicherheitshebelfunktion](#)

AWSFIS unterstützt jetzt Sicherheitshebel, mit denen Sie alle laufenden Experimente schnell beenden und verhindern können, dass neue Experimente beginnen.

3. September 2024

[Neues Kapitel zur Fehlerbehebung](#)

AWSFIS hat eine Anleitung zur Fehlerbehebung hinzugefügt, die Fehlercodes und den Kontext für fehlgeschlagene Experimente enthält.

13. August 2024

[Neue Aktion](#)

Sie können die `aws:dynamodb:global-table-pause-replication` Aktion jetzt verwenden, um die Datenreplikation zwischen der globalen Zieltabelle und ihren Replikattabellen anzuhalten. Die `aws:dynamodb:encrypted-global-table-pause-replication` Aktion wird nicht mehr unterstützt.

24. April 2024

[Neue Experimentieroption für den Aktionsmodus](#)

Sie können den Aktionsmodus auf `skip-all` einstellen, um vor der Ausführung eines Experiments eine Zielvorschau zu generieren.

13. März 2024

[AWSverwaltete Richtlinienaktualisierungen](#)

AWSFIS hat bestehende verwaltete Richtlinien aktualisiert.

25. Januar 2024

[Neue Szenarien und Aktionen](#)

Sie können jetzt die AWS FIS-Szenarien `Cross-Region:Connectivity` und `AZ-Verfügbarkeit: Stromunterbrechung` verwenden.

30. November 2023

Neue Aktion	Sie können die aws:ec2:asg-insufficient-instance-capacity-error Aktion jetzt verwenden.	30. November 2023
Neue Aktion	Sie können die aws:ec2:api-insufficient-instance-capacity-error Aktion jetzt verwenden.	30. November 2023
Neue Aktion	Sie können die aws:network:route-table-disrupt-cross-region-connectivity Aktion jetzt verwenden.	30. November 2023
Neue Aktion	Sie können die aws:network:transit-gateway-disrupt-cross-region-connectivity Aktion jetzt verwenden.	30. November 2023
Neue Aktion	Sie können die aws:dynamodb:encrypted-global-table-pause-replication Aktion jetzt verwenden.	30. November 2023
Neue Aktion	Sie können die aws:s3:bucket-pause-replication Aktion jetzt verwenden.	30. November 2023
Neue Aktion	Sie können die aws:elasticache:interrupt-cluster-az-power Aktion jetzt verwenden.	30. November 2023
Neue Experimentiermöglichkeiten	Sie können jetzt die AWS FIS-Experimentieroptionen für das Targeting auf Konten und die Auflösung leerer Ziele verwenden.	8. November 2023

<u>Namensänderung von FIS AWS</u>	Der Dienstname wurde auf AWS Fault Injection Service aktualisiert.	15. November 2023
<u>AWSverwaltete Richtlinienaktualisierungen</u>	AWSFIS hat bestehende verwaltete Richtlinien aktualisiert.	13. November 2023
<u>Neue Szenario-Bibliothek</u>	Sie können jetzt die Funktion der AWS FIS-Szenariobibliothek verwenden.	7. November 2023
<u>Neuer Experimentplaner</u>	Sie können jetzt die AWS FIS-Funktion zur Planung von Experimenten verwenden.	7. November 2023
<u>AWSverwaltete Richtlinienaktualisierungen</u>	AWSFIS hat bestehende verwaltete Richtlinien aktualisiert.	02. Juni 2023
<u>Neue Aktionen</u>	Sie können die neuen <code>aws:ecs:task</code> und <code>aws:eks:pod</code> Aktionen verwenden.	01. Juni 2023
<u>AWSverwaltete Richtlinienaktualisierungen</u>	AWSFIS hat bestehende verwaltete Richtlinien aktualisiert.	01. Juni 2023
<u>Neues vorkonfiguriertes SSM-Dokument</u>	Sie können das folgende vorkonfigurierte SSM-Dokument verwenden: <code>-Disk-Fill.AWSFIS-Run</code>	28. April 2023

[Neue Aktion](#)

Sie können die `aws:ebs:pause-volume-io` Aktion verwenden, um I/O zwischen den Ziel-Volumes und den Instances, an die sie angehängt sind, eine Pause einzulegen.

27. Januar 2023

[Neue Aktion](#)

Sie können die `aws:network:disrupt-connectivity` Aktion verwenden, um bestimmte Arten von Datenverkehr zu den Zielsubnetzen abzulehnen.

26. Oktober 2022

[Neue Aktion](#)

Sie können die `aws:eks:inject-kubernetes-custom-resource` Aktion verwenden, um ein ChaosMesh oder Litmus-Experiment auf einem einzelnen Zielcluster auszuführen.

7. Juli 2022

[Protokollierung von Experimenten](#)

Sie können Ihre Versuchsvorgänge so konfigurieren, dass sie die Protokolle der Experimentaktivitäten an CloudWatch Logs oder an einen S3-Bucket senden.

28. Februar 2022

[Neue Benachrichtigungen](#)

Wenn sich der Status eines Experiments ändert, sendet AWS FIS eine Benachrichtigung aus. Diese Benachrichtigungen werden als Ereignisse über Amazon zur Verfügung gestellt EventBridge.

24. Februar 2022

[Neue Aktion](#)

Sie können die `aws:ecs:stop-task` Aktion verwenden, um die angegebene Aufgabe zu beenden.

9. Februar 2022

[Neue Aktion](#)

Sie können die `aws:cloudwatch:assert-alarm-state` Aktion verwenden, um zu überprüfen, ob sich die angegebenen Alarme in einem der angegebenen Alarmzustände befinden.

5. November 2021

[Neue vorkonfigurierte SSM-Dokumente](#)

Sie können die folgenden vorkonfigurierten SSM-Dokumente verwenden: `AWSFIS-Run -IO-Stress`, `-Network-Blackhold-Port`, `-Network-LatencySources`, `-Network-Packet-Loss` und `AWSFIS-Run -Network-Packet-Loss-SOURCES`. `AWSFIS-Run AWSFIS-Run`

4. November 2021

[Neue Aktion](#)

Sie können die `aws:ec2:send-spot-instance-interruptions` Aktion verwenden, um eine Benachrichtigung über eine Unterbrechung der Spot-Instanz an Ziel-Spot-Instances zu senden und dann die Ziel-Spot-Instances zu unterbrechen.

20. Oktober 2021

[Neue Aktion](#)

Sie können die `aws:ssm:start-automation-execution` Aktion verwenden, um die Ausführung eines Automatisierungs-Runbooks zu initiieren.

17. September 2021

[Erstversion](#)

Die erste Version des AWS Fault Injection Service-Benutzerhandbuchs.

15. März 2021

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.