Benutzerhandbuch

AWS Fehlerinjektionsservice

AWS Fehlerinjektionsservice: Benutzerhandbuch

Table of Contents

Was ist AWS Fault Injection Service?

AWS Fault Injection Service (AWS FIS) ist ein verwalteter Service, mit dem Sie Fehlersimulationsexperimente für Ihre AWSWorkloads durchführen können. Die Fehlersimulation basiert auf den Prinzipien des chaos-Engineerings. Diese Experimente fordern eine Anwendung an, indem sie störende Ereignisse erstellen, sodass Sie beobachten können, wie Ihre Anwendung reagiert. Sie können diese Informationen dann verwenden, um die Leistung und Ausfallsicherheit Ihrer Anwendungen zu verbessern, damit sie sich wie erwartet verhalten.

Um AWS FIS zu verwenden, richten Sie Experimente ein und führen sie aus, die Ihnen helfen, die realen Bedingungen zu erstellen, die erforderlich sind, um Anwendungsprobleme aufzudecken, die sonst möglicherweise schwer zu finden sein könnten. AWS FIS bietet Vorlagen, die Unterbrechungen erzeugen, sowie die Kontrollen und Integritätsschutz, die Sie für die Ausführung von Experimenten in der Produktion benötigen, z. B. automatisches Rollback oder Stoppen des Experiments, wenn bestimmte Bedingungen erfüllt sind.

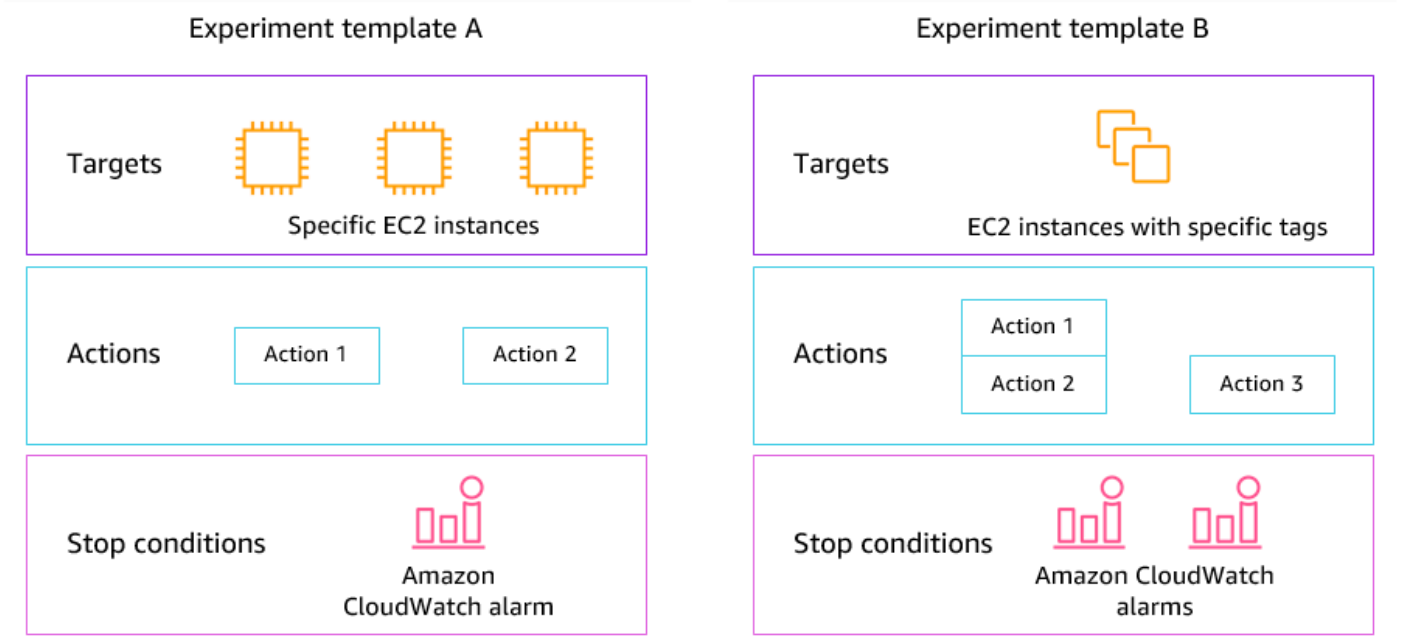
> **Important**
>
> AWS FIS führt echte Aktionen für echte AWS Ressourcen in Ihrem System durch. Bevor Sie mit AWS FIS Experimente in der Produktion ausführen, empfehlen wir Ihnen dringend, eine Planungsphase abzuschließen und die Experimente in einer Vorproduktionsumgebung auszuführen.

Weitere Informationen zur Planung Ihres Experiments finden Sie unter Testzuverlässigkeit und Planen Sie Ihre 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 eine Anwendung oder ein System unter Fehlerbedingungen funktioniert. Um Experimente auszuführen, erstellen Sie zunächst eine Experimentvorlage . Eine Experimentvorlage ist die Vorlage Ihres Experiments. Sie enthält die Aktionen , Ziele und Stoppbedingungen für das Experiment. Nachdem Sie eine Experimentvorlage erstellt haben, können Sie sie verwenden, um ein Experiment auszuführen. Während Ihr Experiment läuft, können Sie seinen Fortschritt verfolgen

und seinen Status anzeigen. Ein Experiment ist abgeschlossen, wenn alle Aktionen im Experiment ausgeführt wurden.



Aktionen

Eine Aktion ist eine Aktivität, die AWS FIS während eines Experiments für eine -AWSRessource ausführt. AWS FIS bietet eine Reihe vorkonfigurierter Aktionen, die auf dem Typ der AWS Ressource basieren. Jede Aktion wird während eines Experiments für eine bestimmte Dauer ausgeführt oder bis Sie das Experiment beenden. Aktionen können sequenziell oder gleichzeitig (parallel) ausgeführt werden.

Targets (Ziele)

Ein Ziel ist eine oder mehrere AWS Ressourcen, für die AWS FIS während eines Experiments eine Aktion ausführt. Sie können bestimmte Ressourcen auswählen oder eine Gruppe von Ressourcen basierend auf bestimmten Kriterien auswählen, z. B. Tags oder Status.

Stopp-Bedingungen

AWS FIS bietet die Kontrollen und Integritätsschutz, die Sie benötigen, um Experimente sicher auf Ihren AWSWorkloads auszuführen. Eine Stoppbedingung ist ein Mechanismus, um ein Experiment zu stoppen, wenn es einen Schwellenwert erreicht, den Sie als Amazon- CloudWatch Alarm definieren. Wenn eine Stoppbedingung ausgelöst wird, während das Experiment ausgeführt wird, stoppt AWS FIS das Experiment.

Unterstützte AWS-Services

AWS FIS bietet vorkonfigurierte Aktionen für bestimmte Arten von Zielen über -AWSServices hinweg. AWS FIS unterstützt Aktionen für Zielressourcen für die folgenden AWS-Services:

- Amazon CloudWatch
- Amazon EBS
- Amazon EC2
- Amazon ECS
- Amazon EKS
- Amazon ElastiCache
- Amazon RDS
- AWS Systems Manager
- Amazon VPC

Für Experimente mit einem Konto müssen sich die Zielressourcen in derselben AWS-Konto wie das Experiment befinden. Sie können AWS FIS-Experimente ausführen, die auf Ressourcen in einem anderen AWS-Konto Konto abzielen, indem Sie AWS FIS-Experimente mit mehreren Konten verwenden.

Weitere Informationen finden Sie unter Aktionen für AWS FIS.

Zugriff auf AWS FIS

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

- AWS Management Console – Bietet eine Weboberfläche, über die Sie auf AWS FIS zugreifen können. Weitere Informationen finden Sie unter Arbeiten mit der AWS Management Console.
- AWS Command Line Interface (AWS CLI) – Bietet Befehle für eine breite Palette von -AWSServices, 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 -AWSRessourcen beschreiben. Mit den Vorlagen können Sie diese Ressourcen als Einheit bereitstellen und verwalten. Weitere Informationen finden Sie in der AWS Ressourcentypreferenz für Fault Injection Service.

- -AWSSDKs – Bietet sprachspezifische APIs und übernimmt viele der Verbindungsdetails, z. B. die Berechnung von Signaturen, die Verarbeitung von Anforderungswiederholungen und die Behandlung von Fehlern. Weitere Informationen finden Sie unter AWS-SDKs.
- HTTPS-API – Bietet Low-Level-API-Aktionen, die Sie mithilfe von HTTPS-Anforderungen aufrufen können. Weitere Informationen finden Sie in der APIAWS-Referenz zum Fault Injection Service .

Preise für AWS FIS

Basierend auf der Anzahl der Zielkonten für Ihr Experiment werden Ihnen pro Minute, die eine Aktion ausführt, von Anfang bis Ende in Rechnung gestellt. Weitere Informationen finden Sie unter AWS FIS-Preise.

Planen Sie Ihre AWS FIS-Experimente

Fehlersimulation ist der Prozess der Belastung einer Anwendung in Test- oder Produktionsumgebungen durch die Erstellung störender Ereignisse, wie Serverausfälle oder API-Drosselung. Wenn Sie beobachten, wie das System reagiert, können Sie dann Verbesserungen implementieren. Wenn Sie Experimente auf Ihrem System ausführen, können Sie dadurch erschöpfte Schwachstellen kontrolliert erkennen, bevor sich diese Schwachstellen auf die Kunden auswirken, die von Ihrem System abhängig sind. Anschließend können Sie die Probleme proaktiv beheben, um unvorhersehbare Ergebnisse zu vermeiden.

Bevor Sie mit der Ausführung von Fehlersimulationsexperimenten mit AWS FIS beginnen, empfehlen wir Ihnen, sich mit den folgenden Prinzipien und Richtlinien vertraut zu machen.

> **Important**
>
> AWS FIS führt echte Aktionen für echte AWS Ressourcen in Ihrem System durch. Bevor Sie mit der Verwendung von AWS FIS beginnen, um Experimente auszuführen, empfehlen wir Ihnen dringend, zunächst eine Planungsphase und einen Test in einer Vorproduktions- oder Testumgebung abzuschließen.

Inhalt

Grundprinzipien und Richtlinien

Bevor Sie mit AWS FIS beginnen, führen Sie die folgenden Schritte aus:

1. Identifizieren der Zielbereitstellung für das Experiment – Identifizieren Sie zunächst die Zielbereitstellung. Wenn dies Ihr erstes Experiment ist, empfehlen wir, in einer Vorproduktions- oder Testumgebung zu beginnen.
2. Überprüfen der Anwendungsarchitektur – Sie müssen sicherstellen, dass Sie alle Anwendungskomponenten, Abhängigkeiten und Wiederherstellungsverfahren für jede Komponente identifiziert haben. Beginnen Sie mit der Überprüfung der Anwendungsarchitektur. Lesen Sie je nach Anwendung das AWS Well-Architected Framework .

3. Steady-State-Verhalten definieren – Definieren Sie das Steady-State-Verhalten Ihres Systems in Bezug auf wichtige technische und geschäftliche Metriken wie Latenz, CPU-Last, fehlgeschlagene Anmeldungen pro Minute, Anzahl der Wiederholungen oder Seitenladegeschwindigkeit.
4. Formieren eines Trichters – Formieren Sie eine Trichter darüber, wie sich das Systemverhalten während des Experiments voraussichtlich ändern wird. Die Definition für eine Backform folgt diesem Format:

   Wenn *eine Fehlersimulationsaktion* durchgeführt wird, sollten die *Auswirkungen auf geschäftliche oder technische Metriken* den *Wert* nicht überschreiten.

   Beispielsweise könnte eine Telefonie für einen Authentifizierungsservice wie folgt lauten: „Wenn die Netzwerklatenz um 10 % zunimmt, gibt es weniger als 1 % mehr Anmeldefehler.“ Nachdem das Experiment abgeschlossen ist, bewerten Sie, ob die Anwendungsausfallsicherheit Ihren geschäftlichen und technischen Erwartungen entspricht.

Wir empfehlen außerdem, diese Richtlinien bei der Arbeit mit AWS FIS zu befolgen:

- Beginnen Sie immer mit dem Experimentieren mit AWS FIS in einer Testumgebung. Beginnen Sie niemals mit einer Produktionsumgebung. Im Laufe Ihrer Fehlersimulationsexperimente können Sie in anderen kontrollierten Umgebungen experimentieren, die über die Testumgebung hinausgehen.
- Bauen Sie das Vertrauen Ihres Teams in die Ausfallsicherheit Ihrer Anwendung auf, indem Sie mit kleinen, einfachen Experimenten beginnen, z. B. der Ausführung der Aktion aws:ec2:stop-instances auf einem Ziel.
- Die Fehlersimulation kann echte Probleme verursachen. Gehen Sie mit Bedacht vor und stellen Sie sicher, dass sich Ihre erste Fehlerunterdrückung auf Test-Instances befindet, damit keine Kunden betroffen sind.
- Testen, testen und testen Sie einige weitere. Die Fehlersimulation 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 Alarmprogramm verfügen, bevor Sie beginnen. Andernfalls können Sie die Auswirkungen Ihrer Experimente nicht verstehen oder messen, was für nachhaltige Fehlersimulationspraktiken von entscheidender Bedeutung ist.

Richtlinien zur Experimentplanung

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

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

- Überprüfen des Ausfallverlaufs – Überprüfen Sie die vorherigen Ausfälle und Ereignisse für Ihr System. Dies kann Ihnen helfen, ein Bild des Gesamtzustands und der Ausfallsicherheit Ihres Systems zu erstellen. Bevor Sie mit der Ausführung von Experimenten auf Ihrem System beginnen, sollten Sie bekannte Probleme und Schwachstellen in Ihrem System beheben.
- Identifizieren von Services mit den größten Auswirkungen – Überprüfen Sie Ihre Services und identifizieren Sie diejenigen, die die größten Auswirkungen auf Ihre Endbenutzer oder Kunden haben, wenn sie ausfallen oder nicht richtig funktionieren.
- Identifizieren des Zielsystems – Das Zielsystem ist das System, auf dem Sie Experimente ausführen werden. Wenn Sie noch nie mit AWS FIS vertraut sind oder noch nie Fehlersimulationsexperimente durchgeführt haben, empfehlen wir Ihnen, zunächst Experimente auf einem Vorproduktions- oder Testsystem durchzuführen.
- Wenden Sie sich an Ihr Team – Fragen Sie, wozu es gefallen ist. Sie können eine Schutzmaßnahme bilden, um ihre Bedenken nachzuweisen oder zu widerlegen. Sie können Ihr Team auch fragen, wozu es nicht geht. Diese Frage kann zwei häufige Fallacien aufdecken: die Fallacy der Sunk-Kosten und die Fallacy der Bestätigungsverzerrung. Wenn Sie auf der Grundlage der Antworten Ihres Teams einen Arzt bilden, können Sie mehr Informationen über die Realität des Zustands Ihres Systems liefern.
- Überprüfen Sie Ihre Anwendungsarchitektur – Überprüfen Sie Ihr System oder Ihre Anwendung 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 dabei helfen, eine sichere, leistungsstarke, belastbare und effiziente Infrastruktur für Ihre Anwendungen und Workloads aufzubauen. Weitere Informationen finden Sie unter AWSWell-Architected.
- Identifizieren der entsprechenden Metriken – Sie können die Auswirkungen eines Experiments auf Ihre AWS Ressourcen mithilfe von Amazon CloudWatch-Metriken überwachen. Sie können diese Metriken verwenden, um die Baseline oder den „steady state“ zu bestimmen, wenn Ihre Anwendung optimal funktioniert. Anschließend können Sie diese Metriken während oder nach dem

Experiment überwachen, um die Auswirkungen zu ermitteln. Weitere Informationen finden Sie unter Überwachen SieAWS die FIS-Nutzungskennzahlen mit Amazon CloudWatch.

- Definieren eines akzeptablen Leistungsschwellenwerts 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.

Anleitungen für den AWS Fault Injection Service

In den folgenden Tutorials erfahren Sie, wie Sie Experimente mit dem AWS AWS Fault Injection Service (FIS) erstellen und ausführen.

Tutorials

- Tutorial: Testen des Anhaltens und Startens von Instances mit AWS FIS
- Tutorial: CPU-Auslastung auf einer Instance mit AWS FIS ausführen
- Tutorial: Testen von Spot-Instance-Unterbrechungen mit AWS FIS
- Tutorial: Simulieren eines Konnektivitätsereignisses
- Tutorial: Planen eines wiederkehrenden Experiments

Tutorial: Testen des Anhaltens und Startens von Instances mit AWS FIS

Sie können AWS Fault Injection Service (AWS FIS) verwenden, um zu testen, wie Ihre Anwendungen mit dem Anhalten und Starten von Instances umgehen. Verwenden Sie dieses Tutorial, um eine Experimentvorlage zu erstellen, die die AWS -FIS-`aws:ec2:stop-instances`Aktion verwendet, um eine Instance und dann eine zweite Instance zu stoppen.

Voraussetzungen

Um dieses Tutorial abzuschließen, stellen Sie sicher, dass Sie Folgendes tun:

- Starten Sie zwei Test-EC2-Instances in Ihrem Konto. Notieren Sie sich nach dem Start Ihrer Instances die IDs beider Instances.
- Erstellen Sie eine IAM-Rolle, die es dem AWS FIS-Service ermöglicht, die `aws:ec2:stop-instances` Aktion in Ihrem Namen auszuführen. Weitere Informationen finden Sie unter IAM-Rollen für AWS FIS-Experimente.
- Stellen Sie sicher, dass Sie Zugriff auf AWS FIS haben. Weitere Informationen finden Sie unter AWS Beispiele für -FIS-Richtlinien.

Schritt 1: Erstellen einer Experimentvorlage

Erstellen Sie die Experimentvorlage mit der AWS FIS-Konsole. In der Vorlage geben Sie zwei Aktionen an, die jeweils drei Minuten lang sequenziell ausgeführt werden. Die erste Aktion stoppt eine der Test-Instances, die AWS FIS nach dem Zufallsprinzip auswählt. Die zweite Aktion stoppt beide Test-Instances.

So erstellen Sie eine Experimentvorlage

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. Geben Sie unter Beschreibung und Name eine Beschreibung und einen Namen für die Vorlage ein.
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 **`stopOneInstance`**.
    c. Wählen Sie für Aktionstyp aws:ec2:stop-instances aus.
    d. Behalten Sie für Ziel das Ziel bei, das AWS FIS für Sie erstellt.
    e. Geben Sie für Aktionsparameter , Instances nach der Dauer starten 3 Minuten (PT3M) an.
    f. Wählen Sie Speichern.
6. Führen Sie für Targets (Ziele) Folgendes aus:
    a. Wählen Sie Bearbeiten für das Ziel aus, das AWS FIS im vorherigen Schritt automatisch für Sie erstellt hat.
    b. Ersetzen Sie den Standardnamen durch einen aussagekräftigeren Namen. Geben Sie z. B. ei **`oneRandomInstance`**.
    c. Stellen Sie sicher, dass der Ressourcentyp aws:ec2:instance ist.
    d. Wählen Sie für Zielmethode die Option Ressourcen-IDs und dann die IDs der beiden Test-Instancesaus.
    e. Wählen Sie für Auswahlmodus die Option Zählen aus. Geben Sie für Anzahl der Ressourcen ein**`1`**.
    f. Wählen Sie Speichern.
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 für Ressourcentyp aws:ec2:instance aus.
c. Wählen Sie für Zielmethode die Option Ressourcen-IDs und dann die IDs der beiden Test-Instancesaus.
d. Wählen Sie für Auswahlmodus die Option Alle aus.
e. Wählen Sie Speichern.

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 für Aktionstyp aws:ec2:stop-instances aus.
c. Wählen Sie unter Nach beginnen 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 , Instances nach der Dauer starten 3 Minuten (PT3M) an.
f. Wählen Sie Speichern.

9. Wählen Sie für Service Access die Option Vorhandene IAM-Rolle verwenden und 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 “Experimentrolle”.
10. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus und geben Sie einen Tag-Schlüssel und einen Tag-Wert an. Die Tags, die Sie hinzufügen, werden auf Ihre Experimentvorlage angewendet, nicht auf die Experimente, die mit der Vorlage ausgeführt werden.
11. Wählen Sie Experimentvorlage erstellen aus. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie ein **`create`** und wählen Sie dann Experimentvorlage erstellen aus.

(Optional) So zeigen Sie das JSON der Experimentvorlage an

Wählen Sie die Registerkarte Export ierenaus. Im Folgenden finden Sie ein Beispiel für das JSON, das durch das vorherige Konsolenverfahren 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 des Experiments

Wenn Sie mit der Erstellung Ihrer Experimentvorlage fertig sind, können Sie damit ein Experiment starten.

So starten Sie ein Experiment

1. Sie sollten sich auf der Detailseite für die Experimentvorlage befinden, die Sie gerade erstellt haben. Wählen Sie andernfalls 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 ein **`start`** und wählen Sie Experiment starten aus.

Schritt 3: Verfolgen des Experimentfortschritts

Sie können den Fortschritt eines laufenden Experiments verfolgen, bis das Experiment abgeschlossen, gestoppt oder fehlgeschlagen ist.

So verfolgen Sie den Fortschritt eines Experiments

1. Sie sollten sich auf der Detailseite für das Experiment befinden, das Sie gerade gestartet haben. Wählen Sie andernfalls Experiments und dann die ID des Experiments aus, um die Detailseite zu öffnen.
2. Um den Status des Experiments anzuzeigen, überprüfen Sie Status im Bereich Details. Weitere Informationen finden Sie unter Experimentstatus .
3. Wenn der Status des Experiments Wird ausgeführt lautet, fahren Sie mit dem nächsten Schritt fort.

Schritt 4: Überprüfen des Experimentergebnisses

Sie können überprüfen, ob die Instances vom Experiment wie erwartet gestoppt und gestartet wurden.

So überprüfen Sie das Ergebnis des Experiments

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/ in einer neuen Browser-Registerkarte oder einem neuen Browserfenster. Auf diese Weise können Sie den Fortschritt des Experiments in der AWS FIS-Konsole weiter verfolgen und gleichzeitig das Ergebnis des Experiments in der Amazon EC2-Konsole anzeigen.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wenn sich der Status der ersten Aktion von Ausstehend in Wird ausgeführt ändert (AWS-FIS-Konsole), ändert sich der Status einer der Ziel-Instances von Wird ausgeführt in Angehalten (Amazon EC2Konsole).
4. Nach drei Minuten ändert sich der Status der ersten Aktion in Abgeschlossen, der Status der zweiten Aktion in Wird ausgeführt und der Status der anderen Ziel-Instance in Angehalten.
5. Nach drei Minuten ändert sich der Status der zweiten Aktion in Abgeschlossen, der Status der Ziel-Instances in Ausführen und der Status des Experiments in Abgeschlossen.

Schritt 5: Bereinigen

Wenn Sie die Test-EC2-Instances, die Sie für dieses Experiment erstellt haben, nicht mehr benötigen, können Sie sie beenden.

So beenden Sie die 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 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 Experimentvorlage nicht mehr benötigen, können Sie sie löschen.

So löschen Sie eine Experimentvorlage mit der AWS FIS-Konsole

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 wählen Sie Aktionen, Experimentvorlage löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie ein **`delete`** und wählen Sie dann Experimentvorlage löschen aus.

Tutorial: CPU-Auslastung auf einer Instance mit AWS FIS ausführen

Sie können den AWS Fault Injection Service (AWS FIS) verwenden, um zu testen, wie Ihre Anwendungen mit CPU-Auslastung umgehen. Verwenden Sie dieses Tutorial, um eine Experimentvorlage zu erstellen, die AWS FIS verwendet, um ein vorkonfiguriertes SSM-Dokument auszuführen, das CPU-Auslastung auf einer Instance ausführt. Das Tutorial verwendet eine Stoppbedingung, um das Experiment anzuhalten, wenn die CPU-Auslastung der Instance einen konfigurierten Schwellenwert überschreitet.

Weitere Informationen finden Sie unter the section called “Vorkonfigurierte AWS FIS SSM-Dokumente”.

Voraussetzungen

Bevor Sie AWS FIS verwenden können, um CPU-Belastung auszuführen, müssen Sie die folgenden Voraussetzungen erfüllen.

Erstellen einer IAM-Rolle

Erstellen Sie eine Rolle und fügen Sie eine Richtlinie an, 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 des Zugriffs auf AWS FIS

Stellen Sie sicher, dass Sie Zugriff auf AWS FIS haben. Weitere Informationen finden Sie unter AWS Beispiele für -FIS-Richtlinien.

Vorbereiten einer Test-EC2-Instance

- Starten Sie eine EC2-Instance mit Amazon Linux 2 oder Ubuntu, wie in den vorkonfigurierten SSM-Dokumenten erforderlich.
- Die Instance muss von SSM verwaltet werden. Um zu überprüfen, ob die Instance von SSM verwaltet wird, öffnen Sie die Fleet Manager-Konsole . Wenn die Instance nicht von SSM verwaltet wird, überprüfen Sie, ob der SSM-Agent installiert ist und ob der Instance eine IAM-Rolle mit der AmazonSSMManagedInstanceCore-Richtlinie angefügt ist. Um den installierten SSM Agent zu überprüfen, stellen Sie eine Verbindung zu Ihrer Instance 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 Instance. Dadurch werden Daten in Abständen von 1 Minute gegen eine zusätzliche Gebühr bereitgestellt. Wählen Sie die Instance aus und wählen Sie Aktionen , Überwachen und Fehlerbehebung, Detaillierte Überwachung verwalten aus.

Schritt 1: Erstellen eines CloudWatch Alarms für eine Stoppbedingung

Konfigurieren Sie einen CloudWatch Alarm, damit Sie das Experiment beenden können, wenn die CPU-Auslastung den von Ihnen angegebenen Schwellenwert überschreitet. Mit dem folgenden Verfahren wird der Schwellenwert auf 50 % CPU-Auslastung für die Ziel-Instance festgelegt. Weitere Informationen finden Sie unter Stoppbedingungen.

So erstellen Sie einen Alarm, der angibt, wann 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 Ziel-Instance aus und wählen Sie Aktionen , Überwachen und Fehler beheben, CloudWatch Alarme verwalten aus.

4. Verwenden Sie für Alarmbenachrichtigung den Schalter, um Amazon SNS-Benachrichtigungen zu deaktivieren.
5. Verwenden Sie für Alarmschwellenwerte die folgenden Einstellungen:
   - Beispiele gruppieren nach : Maximum
   - Art der zu prüfenden Daten: CPU-Auslastung
   - Prozent: **`50`**
   - Zeitraum: **`1 Minute`**
6. Wenn Sie mit der Konfiguration des Alarms fertig sind, wählen Sie Erstellen aus.

Schritt 2: Erstellen einer Experimentvorlage

Erstellen Sie die Experimentvorlage mit der AWS FIS-Konsole. In der Vorlage geben Sie die folgende Aktion an, die ausgeführt werden soll: aws:ssm:send-command/AWSFIS-Run-CPU-Stress .

So erstellen Sie eine Experimentvorlage

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. Geben Sie unter Beschreibung und Name eine Beschreibung und einen Namen für die Vorlage ein.
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 für Aktionstyp aws:ssm:send-command/AWSFIS-Run-CPU-Stress aus. Dadurch wird automatisch der ARN des SSM-Dokuments zum Dokument-ARN hinzugefügt.
   d. Behalten Sie für Ziel das Ziel bei, das AWS FIS für Sie erstellt.
   e. Geben Sie für Aktionsparameter , Dokumentparameter Folgendes ein:

      ```
      {"DurationSeconds":"120"}
      ```

   f. Geben Sie für Aktionsparameter , Dauer 5 Minuten (PT5M) an.
   g. Wählen Sie Speichern.

6. Führen Sie für Targets (Ziele) Folgendes aus:

   a. Wählen Sie Bearbeiten für das Ziel aus, das AWS FIS im vorherigen Schritt automatisch für Sie erstellt hat.

   b. Ersetzen Sie den Standardnamen durch einen aussagekräftigeren Namen. Geben Sie z. B. ei **`testInstance`**.

   c. Stellen Sie sicher, dass der Ressourcentyp aws:ec2:instance ist.

   d. Wählen Sie für Zielmethode die Option Ressourcen-IDs und dann die ID der Test-Instance aus.

   e. Wählen Sie für Auswahlmodus die Option Alle aus.

   f. Wählen Sie Speichern.

7. Wählen Sie für Service Access die Option Vorhandene IAM-Rolle verwenden und 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 “Experimentrolle”.

8. Wählen Sie unter Stoppbedingungen den CloudWatch Alarm aus, den Sie in Schritt 1 erstellt haben.

9. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus und geben Sie einen Tag-Schlüssel und einen Tag-Wert an. Die Tags, die Sie hinzufügen, werden auf Ihre Experimentvorlage angewendet, nicht auf die Experimente, die mit der Vorlage ausgeführt werden.

10. Wählen Sie Experimentvorlage erstellen aus.

(Optional) So zeigen Sie das JSON der Experimentvorlage an

Wählen Sie die Registerkarte Export ierenaus. Im Folgenden finden Sie ein Beispiel für das JSON, das durch das vorherige Konsolenverfahren 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 des Experiments

Wenn Sie mit der Erstellung Ihrer Experimentvorlage fertig sind, können Sie damit ein Experiment starten.

So starten Sie ein Experiment

1. Sie sollten sich auf der Detailseite für die Experimentvorlage befinden, die Sie gerade erstellt haben. Wählen Sie andernfalls 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 des Experimentfortschritts

Sie können den Fortschritt eines laufenden Experiments verfolgen, bis das Experiment abgeschlossen ist, beendet wird oder fehlschlägt.

So verfolgen Sie den Fortschritt eines Experiments

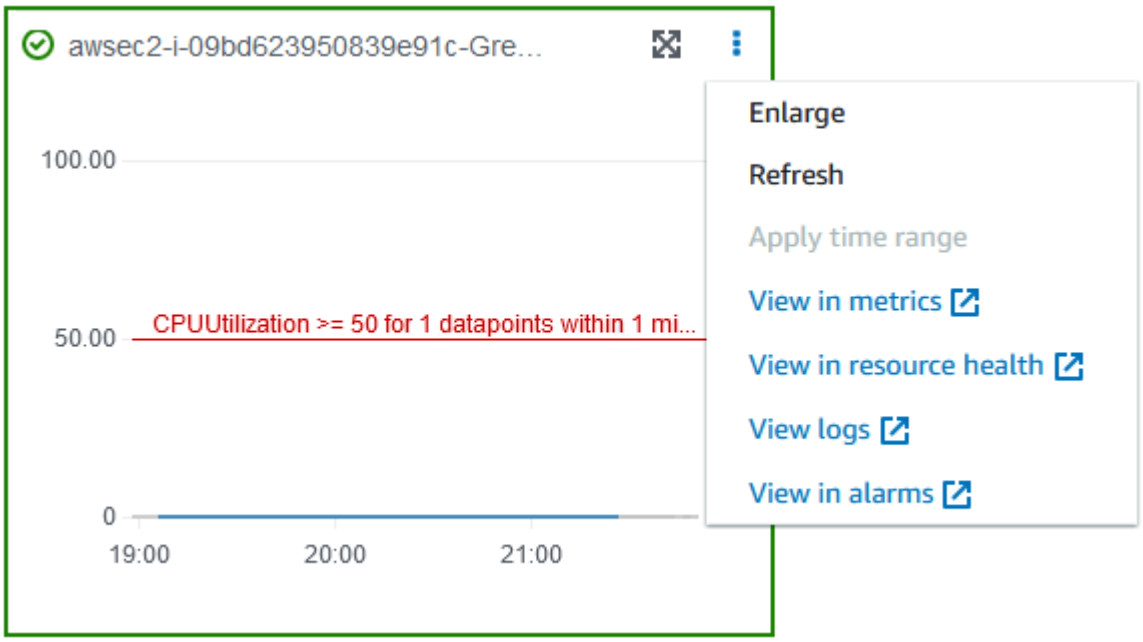
1. Sie sollten sich auf der Detailseite für das Experiment befinden, das Sie gerade gestartet haben. Wählen Sie andernfalls Experimente und dann die ID des Experiments aus, um die Detailseite für das Experiment zu öffnen.
2. Um den Status des Experiments anzuzeigen, überprüfen Sie Status im Bereich Details. Weitere Informationen finden Sie unter Experimentstatus .
3. Wenn der Experimentstatus Wird ausgeführt lautet, fahren Sie mit dem nächsten Schritt fort.

Schritt 5: Überprüfen der Experimentergebnisse

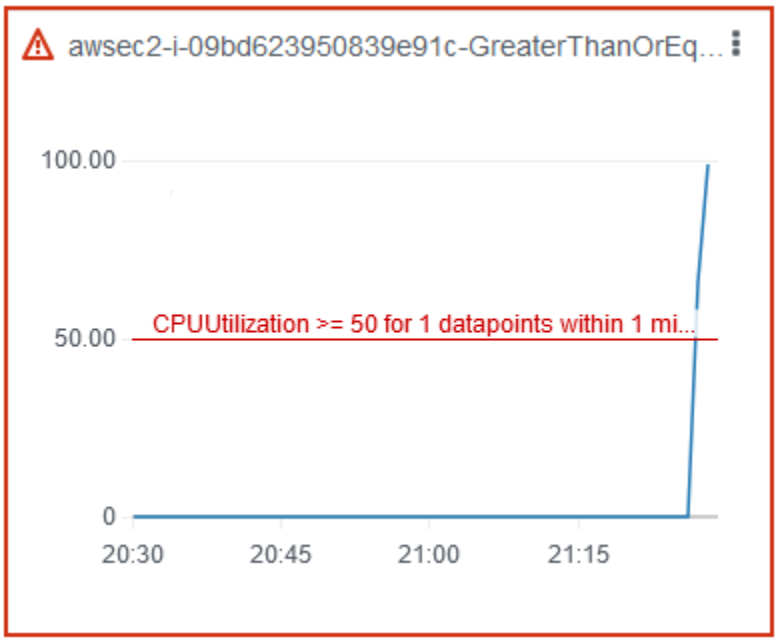
Sie können die CPU-Auslastung Ihrer Instance überwachen, während das Experiment läuft. Wenn die CPU-Auslastung den Schwellenwert erreicht, wird der Alarm ausgelöst und das Experiment wird durch die Stoppbedingung angehalten.

So überprüfen Sie die Ergebnisse des Experiments

1. Wählen Sie die Registerkarte Stoppbedingungen aus. Der grüne Rahmen und das grüne Häkchen zeigen an, dass der Anfangsstatus des Alarms ist0K. Die rote Linie gibt den Alarmschwellenwert an. Wenn Sie ein detaillierteres Diagramm bevorzugen, wählen Sie im Widget-Menü die Option Vergrößern aus.



2. Wenn die CPU-Auslastung den Schwellenwert überschreitet, zeigen das Symbol für den roten Rahmen und den roten Ausrufezeichen auf der Registerkarte Stoppbedingungen an, dass sich der Alarmstatus in geändert hat`ALARM`. Im Bereich Details lautet der Status des Experiments Angehalten . Wenn Sie den Status auswählen, wird die Meldung „Experiment wurde durch Stoppbedingung angehalten“ angezeigt.



3. Wenn die CPU-Auslastung unter den Schwellenwert sinkt, zeigen der grüne Rahmen und das grüne Häkchensymbol an, dass sich der Alarmstatus in geändert hat`OK`.

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 erhalten oder die Alarmeinstellungen bearbeiten können.

Schritt 6: Bereinigen

Wenn Sie die Test-EC2-Instance, 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 Test-Instances und dann Instance-Status, Instance 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.

So löschen Sie eine Experimentvorlage mit der AWS FIS-Konsole

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 und dann Aktionen, Experimentvorlage löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie ein **`delete`** und wählen Sie dann Experimentvorlage löschen aus.

Tutorial: Testen von Spot-Instance-Unterbrechungen mit AWS FIS

Spot-Instances verwenden verfügbare zusätzliche EC2-Kapazität für einen Rabatt 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 Amazon EC2-Benutzerhandbuch.

Sie können AWS Fault Injection Service (AWS FIS) verwenden, um zu testen, wie Ihre Anwendungen mit einer Spot-Instance-Unterbrechung umgehen. Verwenden Sie dieses Tutorial, um eine

Experimentvorlage zu erstellen, die die AWS -FIS-`aws:ec2:send-spot-instance-interruptions`Aktion verwendet, um eine Ihrer Spot-Instances zu unterbrechen.

Um das Experiment mit der Amazon EC2-Konsole zu starten, lesen Sie alternativ Initiieren einer Spot-Instance-Unterbrechung im Amazon EC2-Benutzerhandbuch.

Voraussetzungen

Bevor Sie mit AWS FIS eine Spot-Instance unterbrechen können, müssen Sie die folgenden Voraussetzungen erfüllen.

1. Erstellen einer IAM-Rolle

Erstellen Sie eine Rolle und fügen Sie eine Richtlinie an, die es AWS FIS ermöglicht, die `aws:ec2:send-spot-instance-interruptions` Aktion in Ihrem Namen auszuführen. Weitere Informationen finden Sie unter IAM-Rollen für AWS FIS-Experimente.

2. Überprüfen des Zugriffs auf AWS FIS

Stellen Sie sicher, dass Sie Zugriff auf AWS FIS haben. Weitere Informationen finden Sie unter AWS Beispiele für -FIS-Richtlinien.

3. (Optional) Erstellen einer Spot-Instance-Anforderung

Wenn Sie möchten, dass eine neue Spot-Instance für dieses Experiment verwendet wird, verwenden Sie den Befehl run-instances, um eine Spot-Instance anzufordern. Standardmäßig werden Spot-Instances beendet, die unterbrochen werden. Wenn Sie das Unterbrechungsverhalten auf setzen`stop`, müssen Sie auch den Typ auf setzen`persistent`. Legen Sie für dieses Tutorial das Unterbrechungsverhalten nicht auf fest`hibernate`, 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 bewirkt dies, sodass 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 Name=interruptMe zu Ihrer Ziel-Spot-Instance hinzuzufügen.

```
aws ec2 create-tags \
    --resources i-0abcdef1234567890 \
    --tags Key=Name,Value=interruptMe
```

Schritt 1: Erstellen einer Experimentvorlage

Erstellen Sie die Experimentvorlage mit der AWS FIS-Konsole. In der Vorlage geben Sie die Aktion an, die ausgeführt wird. Die Aktion unterbricht die Spot-Instance mit dem angegebenen Tag. Wenn es mehr als eine Spot-Instance mit dem -Tag gibt, wählt AWS FIS eine davon nach dem Zufallsprinzip aus.

So erstellen Sie eine Experimentvorlage

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. Geben Sie unter Beschreibung und Name eine Beschreibung und einen Namen für die Vorlage ein.

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 **`interruptSpotInstance`**.

   c. Wählen Sie für Aktionstyp aws:ec2:send-spot-instance-interruptions aus.

   d. Behalten Sie für Ziel das Ziel bei, das AWS FIS für Sie erstellt.

   e. Geben Sie für Aktionsparameter , Dauer vor Unterbrechung 2 Minuten (PT2M) an.

   f. Wählen Sie Speichern.

6. Führen Sie für Targets (Ziele) Folgendes aus:

   a. Wählen Sie Bearbeiten für das Ziel aus, das AWS FIS im vorherigen Schritt automatisch für Sie erstellt hat.

   b. Ersetzen Sie den Standardnamen durch einen aussagekräftigeren Namen. Geben Sie z. B. ei **`oneSpotInstance`**.

   c. Stellen Sie sicher, dass der Ressourcentyp aws:ec2:spot-instance ist.

   d. Wählen Sie für Zielmethode 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 aus und geben Sie **`State.Name`** als Pfad und **`running`** als Wert ein.

   g. Wählen Sie für Auswahlmodus die Option Zählen aus. Geben Sie für Anzahl der Ressourcen ein**`1`**.

   h. Wählen Sie Speichern.

7. Wählen Sie für Service Access die Option Vorhandene IAM-Rolle verwenden und 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 “Experimentrolle”.

8. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus und geben Sie einen Tag-Schlüssel und einen Tag-Wert an. Die Tags, die Sie hinzufügen, werden auf Ihre Experimentvorlage angewendet, nicht auf die Experimente, die mit der Vorlage ausgeführt werden.

9. Wählen Sie Experimentvorlage erstellen aus. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie ein **`create`** und wählen Sie dann Experimentvorlage erstellen aus.

(Optional) So zeigen Sie die JSON-Datei für die Experimentvorlage an

Wählen Sie die Registerkarte Export ierenaus. Im Folgenden finden Sie ein Beispiel für das JSON, das durch das vorherige Konsolenverfahren 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 des Experiments

Wenn Sie mit der Erstellung Ihrer Experimentvorlage fertig sind, können Sie damit ein Experiment starten.

So starten Sie ein Experiment

1. Sie sollten sich auf der Detailseite für die Experimentvorlage befinden, die Sie gerade erstellt haben. Wählen Sie andernfalls 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 ein **`start`** und wählen Sie Experiment starten aus.

Schritt 3: Verfolgen des Experimentfortschritts

Sie können den Fortschritt eines laufenden Experiments verfolgen, bis das Experiment abgeschlossen, gestoppt oder fehlgeschlagen ist.

So verfolgen Sie den Fortschritt eines Experiments

1. Sie sollten sich auf der Detailseite für das Experiment befinden, das Sie gerade gestartet haben. Wählen Sie andernfalls Experiments und dann die ID des Experiments aus, um die Detailseite zu öffnen.
2. Um den Status des Experiments anzuzeigen, überprüfen Sie Status im Bereich Details. Weitere Informationen finden Sie unter Experimentstatus .
3. Wenn der Status des Experiments Wird ausgeführt lautet, fahren Sie mit dem nächsten Schritt fort.

Schritt 4: Überprüfen des Experimentergebnisses

Wenn die Aktion für dieses Experiment abgeschlossen ist, geschieht Folgendes:

- Die Ziel-Spot-Instance erhält eine Empfehlung zum Neuausgleich der Instance.
- Zwei Minuten, bevor Amazon EC2 Ihre Instance beendet oder beendet, wird eine Benachrichtigung über die Unterbrechung der Spot-Instance ausgegeben.
- Nach zwei Minuten wird die Spot-Instance beendet oder angehalten.
- Eine Spot-Instance, die von AWS FIS angehalten wurde, bleibt angehalten, bis Sie sie neu starten.

So überprüfen Sie, ob die Instance durch das Experiment unterbrochen wurde

1. Öffnen Sie die Amazon EC2-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 in `instance-terminated-by-experiment`.
   - `stop` – Der Status ändert sich in `marked-for-stop-by-experiment` und dann in `instance-stopped-by-experiment`.
4. Wählen Sie unter Instances die Spot Instance aus. Der ursprüngliche Status ist `Running`. Zwei Minuten nach Erhalt der Spot-Instance-Unterbrechungsbenachrichtigung ändert sich der Status wie folgt:
   - `stop` – Der Status ändert sich in `Stopping` und dann in `Stopped`.
   - `terminate` – Der Status ändert sich in `Shutting-down` und dann in `Terminated`.

Schritt 5: Bereinigen

Wenn Sie die Test-Spot-Instance für dieses Experiment mit einem Unterbrechungsverhalten von erstellt haben `stop` und sie nicht mehr benötigen, können Sie die Spot-Instance-Anforderung abbrechen und die Spot-Instance beenden.

So brechen Sie die Anforderung ab und beenden die Instance mit der AWS CLI

1. Verwenden Sie den cancel-spot-instance-requests Befehl , um die Spot-Instance-Anforderung abzubrechen.

```
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.

So löschen Sie eine Experimentvorlage mit der AWS FIS-Konsole

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 und dann Aktionen, Experimentvorlage löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie ein **`delete`** und wählen Sie dann Experimentvorlage löschen aus.

Tutorial: Simulieren eines Konnektivitätsereignisses

Sie können AWS Fault Injection Service (AWS FIS) verwenden, um eine Vielzahl von Konnektivitätsereignissen zu simulieren. AWS FIS simuliert Konnektivitätsereignisse, indem Netzwerkverbindungen auf eine der folgenden Arten blockiert werden:

- `all` – Verweigert den gesamten Datenverkehr, der in das Subnetz gelangt und dieses verlässt. Beachten Sie, dass diese Option den Datenverkehr innerhalb des Subnetzes zulässt, einschließlich des Datenverkehrs zu und von Netzwerkschnittstellen im Subnetz.
- `availability-zone` – Verweigert den Intra-VPC-Datenverkehr zu und von Subnetzen in anderen Availability Zones.
- `dynamodb` – Verweigert den Datenverkehr zum und vom regionalen Endpunkt für DynamoDB in der aktuellen Region.
- `prefix-list` – Verweigert den Datenverkehr zur und von der angegebenen Präfixliste.

- `s3` – Verweigert den Datenverkehr zum und vom regionalen Endpunkt für Amazon S3 in der aktuellen Region.
- `vpc` – Verweigert das Ein- und Verlassen des Datenverkehrs in der VPC.

Verwenden Sie dieses Tutorial, um eine Experimentvorlage zu erstellen, die die AWS -FIS-`aws:network:disrupt-connectivity`Aktion verwendet, um einen Verbindungsverlust mit Amazon S3 in einem Zielsubnetz einzuführen.

Themen

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, benötigen Sie eine Rolle mit den entsprechenden Berechtigungen in Ihrem und eine Test-AWS-KontoAmazon-EC2-Instance: Amazon EC2

Eine Rolle mit Berechtigungen in Ihrem AWS-Konto

Erstellen Sie eine Rolle und fügen Sie eine Richtlinie an, die es AWS FIS ermöglicht, die `aws:network:disrupt-connectivity` Aktion in Ihrem Namen auszuführen.

Ihre IAM-Rolle erfordert die folgende Richtlinie:

- AWSFaultInjectionSimulatorNetworkAccess – Gewährt AWS FIS-Serviceberechtigungen in Amazon EC2-Netzwerken und anderen erforderlichen Services, um AWS FIS-Aktionen im Zusammenhang mit der Netzwerkinfrastruktur durchzuführen.

> **Note**
>
> Der Einfachheit halber verwendet dieses Tutorial eine von AWS verwaltete Richtlinie. Für den Produktionseinsatz empfehlen wir Ihnen, stattdessen nur die Mindestberechtigungen zu erteilen, die für Ihren Anwendungsfall erforderlich sind.
> 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-Test-Instance

Starten Sie eine Test-Amazon EC2 und stellen Sie eine Verbindung mit ihr her. Sie können das folgende Tutorial verwenden, um eine Amazon EC2-Instance zu starten und eine Verbindung zu ihr herzustellen: Tutorial: Erste Schritte mit Amazon EC2-Linux-Instances im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Schritt 1: Erstellen einer AWS -FIS-Experimentvorlage

Erstellen Sie die Experimentvorlage mithilfe der AWS FIS AWS Management Console. Eine AWS -FIS-Vorlage besteht aus Aktionen, Zielen, Stoppbedingungen und einer Experimentrolle. Weitere Informationen zur Funktionsweise der Vorlagen finden Sie unter Experimentvorlagen für AWS FIS.

Bevor Sie beginnen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Eine IAM-Rolle mit den richtigen Berechtigungen.
- Eine Amazon-EC2-Instance.
- Die Subnetz-ID Ihrer Amazon 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 aus.
4. Geben Sie eine Beschreibung für die Vorlage ein, z. B. `Amazon S3 Network Disrupt Connectivity`.
5. Wählen Sie unter Aktionen die Option Aktion hinzufügen aus.

a. Geben Sie für den Namen ein`disruptConnectivity`.

b. Wählen Sie für Aktionstyp aws:network:disrupt-connectivity aus.

c. Legen Sie unter Aktionsparameter die Dauer auf fest2 `minutes`.

d. Wählen Sie unter Bereich die Option s3 aus.

e. Wählen Sie oben Speichern aus.

6. Unter Ziele sollte das automatisch erstellte Ziel angezeigt werden. Wählen Sie Bearbeiten aus.

   a. Stellen Sie sicher, dass der Ressourcentyp ist`aws:ec2:subnet`.

   b. Wählen Sie unter Zielmethode die Option Ressourcen-IDs und dann das Subnetz aus, das Sie beim Erstellen Ihrer Amazon EC2-Instance in den Schritten Voraussetzungen verwendet haben.

   c. Stellen Sie sicher, dass der Auswahlmodus Alle ist.

   d. Wählen Sie Speichern.

7. Wählen Sie unter Servicezugriff die IAM-Rolle aus, die Sie wie im Tutorial 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 “Experimentrolle”.

8. (Optional) Unter Stoppbedingungen können Sie einen CloudWatch Alarm auswählen, um das Experiment zu beenden, wenn die Bedingung eintritt. Weitere Informationen finden Sie unter Stoppbedingungen für AWS FIS .

9. (Optional) Unter Protokolle können Sie einen Amazon S3-Bucket auswählen oder Protokolle an CloudWatch für Ihr Experiment senden.

10. Wählen Sie Experimentvorlage erstellen und geben Sie ein, wenn Sie zur Bestätigung aufgefordert werden`create`. Wählen Sie dann Experimentvorlage erstellen aus.

Schritt 2: Pingen eines Amazon S3-Endpunkts

Stellen Sie sicher, dass Ihre Amazon EC2-Instance einen Amazon S3-Endpunkt erreichen kann.

1. Stellen Sie eine Verbindung mit der Amazon EC2-Instance her, die Sie in den Schritten Voraussetzungen erstellt haben.

   Informationen zur Fehlerbehebung finden Sie unter Fehlerbehebung bei der Verbindung mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.

2. Überprüfen Sie , um die zu sehenAWS-Region, in der sich Ihre Instance befindet. Sie können dies in der Amazon EC2-Konsole oder durch Ausführen des folgenden Befehls tun.

   ```
   hostname
   ```

   Wenn Sie beispielsweise eine Amazon EC2-Instance in gestartet habenus-west-2, wird die folgende Ausgabe angezeigt.

   ```
   [ec2-user@ip-172.16.0.0 ~]$ hostname
   ip-172.16.0.0.us-west-2.compute.internal
   ```

3. Pingen eines Amazon S3-Endpunkts in Ihrem AWS-Region. Ersetzen Sie *AWS-Region* durch Ihre Region.

   ```
   ping -c 1 s3.AWS-Region.amazonaws.com
   ```

   Für die Ausgabe sollten Sie einen erfolgreichen Ping mit 0 % Paketverlust sehen, 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 Ihres AWS FIS-Experiments

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 Experimentvorlage aus, die Sie erstellt haben, um die 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 Experiment starten aus.

Schritt 4: Verfolgen des Fortschritts 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 Sie dies nicht tun, wählen Sie Experiments und dann die ID des Experiments aus, um die Detailseite zu öffnen.
2. Um den Status des Experiments anzuzeigen, überprüfen Sie den Status im Detailbereich. Weitere Informationen finden Sie unter Experimentstatus.
3. Wenn der Status des Experiments Wird ausgeführt lautet, fahren Sie mit dem nächsten Schritt fort.

Schritt 5: Überprüfen der Amazon S3-Netzwerkunterbrechung

Sie können den Fortschritt des Experiments überprüfen, indem Sie einen Ping an den Amazon S3Endpunkt senden.

- Pingen Sie von Ihrer Amazon EC2-Instance aus an den Amazon S3-Endpunkt in Ihrem AWS-Region. Ersetzen Sie *AWS-Region* durch Ihre Region.

  ```
  ping -c 1 s3.AWS-Region.amazonaws.com
  ```

  Für die Ausgabe sollten Sie einen erfolglosen Ping mit 100 % Paketverlust sehen, 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-Instance, die Sie für dieses Experiment erstellt haben, oder die AWS FIS-Vorlage nicht mehr benötigen, können Sie sie entfernen.

So entfernen Sie die Amazon EC2-Instance

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 aus, wählen Sie Instance-Status und dann Instance beenden aus.
4. Wählen Sie Terminate (Kündigen) aus, wenn Sie zur Bestätigung aufgefordert werden.

So löschen Sie die Experimentvorlage mithilfe der AWS FIS-Konsole

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 und dann Aktionen, Experimentvorlage löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie ein `delete`und wählen Sie dann Experimentvorlage löschen aus.

Tutorial: Planen eines wiederkehrenden Experiments

Mit AWS Fault Injection Service (AWS FIS) können Sie Fehlersimulationsexperimente für Ihre AWSWorkloads durchführen. Diese Experimente werden auf Vorlagen ausgeführt, die eine oder mehrere Aktionen enthalten, die auf bestimmten Zielen ausgeführt werden sollen. Wenn Sie auch verwendenAmazon EventBridge, können Sie Ihre Experimente als einmalige oder wiederkehrende Aufgaben planen.

Verwenden Sie dieses Tutorial, um einen EventBridge Zeitplan zu erstellen, der alle 5 Minuten eine AWS -FIS-Experimentvorlage ausführt.

Aufgaben

- Voraussetzungen
- Schritt 1: Erstellen einer IAM-Rolle und -Richtlinie
- Schritt 2: Erstellen eines Amazon EventBridge Schedulers
- Schritt 3: Überprüfen Ihres Experiments
- Schritt 4: Bereinigen

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, muss über eine AWS -FIS-Experimentvorlage verfügen, die Sie nach einem Zeitplan ausführen möchten. Wenn Sie bereits über eine funktionierende Experimentvorlage verfügen, notieren Sie sich die Vorlagen-ID und AWS-Region. Andernfalls können Sie eine Vorlage erstellen, indem Sie den Anweisungen unter folgen the section called “Testen von Instance-Stopp und -Start”und dann zu diesem Tutorial zurückkehren.

Schritt 1: Erstellen einer IAM-Rolle und -Richtlinie

So erstellen Sie eine IAM-Rolle und -Richtlinie

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Rollen und dann 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.

   ```
   {
       "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.

   ```
   {
       "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-Experimente ausführt, die einen bestimmten Tag-Wert haben. Die folgende Richtlinie gewährt beispielsweise die `StartExperiment` Berechtigung für alle AWS FIS-Experimentvorlagen, schränkt den Scheduler jedoch so ein, dass er nur Experimente ausführt, die mit gekennzeichnet sind`Purpose=Schedule`.

```
{
    "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. Benennen Sie unter Richtlinie überprüfen Ihre Richtlinie `FIS_RecurringExperiment`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. Unter Name, überprüfen und erstellen Sie , benennen Sie die Rolle `FIS_RecurringExperiment_role`und wählen Sie dann Rolle erstellen aus.

Schritt 2: Erstellen eines Amazon EventBridge Schedulers

So erstellen Sie einen -Amazon EventBridgeScheduler

1. Öffnen Sie die Amazon- EventBridge Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im linken Navigationsbereich Zeitpläne aus.
3. Stellen Sie sicher, dass Sie sich in derselben AWS-Region wie Ihre AWS FIS-Experimentvorlage befinden.
4. Wählen Sie Zeitplan erstellen und füllen Sie Folgendes aus:
   - Fügen Sie unter Zeitplanname 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 Rate-Ausdruck 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 APIs aus und suchen Sie dann nach AWS FIS .
6. Wählen Sie AWS FIS und dann ausStartExperiment.
7. Fügen Sie unter Eingabe die folgende JSON-Nutzlast ein. Ersetzen Sie den *`your-experiment-template-id`* Wert durch die Vorlagen-ID Ihres Experiments. ist `ClientToken` eine eindeutige Kennung für den Scheduler. In diesem Tutorial verwenden wir ein vom Amazon EventBridge Scheduler erlaubtes Kontextschlüsselwort. Weitere Informationen finden Sie unter Hinzufügen von Kontextattributen im Amazon- EventBridge Benutzerhandbuch.

```
{
    "ClientToken": "<aws.scheduler.execution-id>",
```

```
    "ExperimentTemplateId": "your-experiment-template-id"
}
```

Wählen Sie Weiter aus.

8. (Optional) Unter Einstellungen können Sie die Wiederholungsrichtlinie , die Warteschlange für unzustellbare Nachrichten (DLQ) und die Verschlüsselungseinstellungen festlegen. Alternativ können Sie die Standardwerte beibehalten.
9. Wählen Sie unter Berechtigungen die Option Vorhandene Rolle verwenden aus und suchen Sie dann nach `FIS_RecurringExperiment_role`.
10. Wählen Sie Weiter aus.
11. Überprüfen Sie unter Zeitplan überprüfen und erstellen Ihre Scheduler-Details und wählen Sie dann Zeitplan erstellen aus.

Schritt 3: Überprüfen Ihres Experiments

So überprüfen Sie, ob Ihr AWS FIS-Experiment planmäßig ausgeführt wurde

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 nach der Erstellung Ihres Zeitplans sollte Ihr Experiment ausgeführt werden.

Schritt 4: Bereinigen

So deaktivieren Sie Ihren Amazon EventBridge Scheduler

1. Öffnen Sie die Amazon- EventBridge Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im linken Navigationsbereich Zeitpläne aus.
3. Wählen Sie Ihren neu erstellten Scheduler und dann Deaktivieren aus.

Aktionen für AWS FIS

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.

Inhalt

Aktions-Identifikatoren

Jede AWS FIS Aktion hat einen Bezeichner 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 unterAWS FIS Referenz für Aktionen. Informationen zum Abrufen der Liste mithilfe von finden Sie unterListet die Aktionen auf. AWS CLI

Aktionsparameter

Einige AWS FIS Aktionen haben zusätzliche Parameter, die für die Aktion spezifisch sind. Diese Parameter werden verwendet, um Informationen an den AWS FIS Zeitpunkt weiterzuleiten, an dem 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 unterVerwenden 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 versetzt das Ziel in den Zustand zurück, in dem es sich vor der Ausführung der Aktion befand. 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": {
    "resource_type": "resource_name"
}
```

AWS FIS Aktionen unterstützen die folgenden Ressourcentypen für Aktionsziele:

- Auto Scaling Scaling-Gruppen — Amazon EC2 Auto Scaling Scaling-Gruppen
- Eimer — Amazon S3 S3-Eimer
- Cluster — Amazon EKS-Cluster
- Cluster — Amazon ECS-Cluster oder Amazon Aurora Aurora-DB-Cluster
- DBInstances — Amazon RDS-DB-Instances
- Verschlüsselte globale Tabellen — Amazon DynamoDB; globale Tabellen, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt sind
- Instanzen — Amazon EC2 EC2-Instances
- Knotengruppen — Amazon EKS-Knotengruppen

- Pods — Kubernetes-Pods auf Amazon EKS
- ReplicationGroups— ElastiCache Redis-Replikationsgruppen
- Rollen — IAM-Rollen
- SpotInstances— Amazon EC2-Spot-Instances
- Subnetze — VPC-Subnetze
- Aufgaben — Amazon ECS-Aufgaben
- TransitGateways— Transit-Gateways
- Volumen — Amazon EBS-Volumen

Beispiele finden Sie unter the section called “Beispielaktionen”.

AWS FIS Referenz für Aktionen

In dieser Referenz werden die häufigsten Aktionen unter 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.AWS CLI

Weitere Informationen finden Sie unter Aktionen für AWS FIS und So funktioniert der AWS Fault Injection Service mit IAM.

Aktionen

- Aktionen zur Fehlerinjektion
- Warten Sie auf die Aktion
- CloudWatch Amazon-Aktionen
- Amazon DynamoDB DynamoDB-Aktionen
- Amazon EBS-Aktionen
- Amazon EC2 EC2-Aktionen
- Amazon ECS-Aktionen
- Amazon EKS-Aktionen
- ElastiCache Amazon-Aktionen
- Netzwerkaktionen

- Amazon RDS-Aktionen
- Amazon-S3-Aktionen
- Systems Manager Manager-Aktionen

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.

Ressourcentyp

- aws:iam:role

Parameter

- 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`.
- percentage— Der Prozentsatz (1—100) der Aufrufe, in die der Fehler eingefügt 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 Aktionen in der Amazon EC2 API-Referenz.

Berechtigungen

- `fis:InjectApiInternalError`

aws:fis:inject-api-throttle-error

Fügt Drosselungsfehler in Anfragen ein, die von der IAM-Zielrolle gestellt wurden.

Ressourcentyp

- aws:iam:role

Parameter

- 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`.
- percentage— Der Prozentsatz (1—100) der Aufrufe, in die der Fehler eingefügt 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 Aktionen in der Amazon EC2 API-Referenz.

Berechtigungen

- `fis:InjectApiThrottleError`

aws:fis:inject-api-unavailable-error

Fügt Unverfügbare-Fehler in Anfragen ein, die von der IAM-Zielrolle gestellt wurden.

Ressourcentyp

- aws:iam:role

Parameter

- 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`.

- percentage— Der Prozentsatz (1—100) der Aufrufe, in die der Fehler eingefügt 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 Aktionen in der Amazon EC2 API-Referenz.

Berechtigungen

- `fis:InjectApiUnavailableError`

Warten Sie auf die Aktion

AWS FIS unterstützt die folgende Warteaktion.

aws:fis:wait

Führt die AWS FIS Warteaktion aus.

Parameter

- 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

- None

CloudWatch Amazon-Aktionen

AWS FIS unterstützt die folgende CloudWatch Amazon-Aktion.

aws:cloudwatch:assert-alarm-state

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

Ressourcentyp

- None

Parameter

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

Berechtigungen

- `cloudwatch:DescribeAlarms`

Amazon DynamoDB DynamoDB-Aktionen

AWS FIS unterstützt die folgende Amazon DynamoDB DynamoDB-Aktion.

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

Unterbricht die globale Tabellenreplikation von Amazon DynamoDB für Tabellen, die mit vom AWS Key Management Service Kunden verwalteten Schlüsseln (CMK) verschlüsselt wurden. Durch diese Aktion werden der dienstverknüpften Rolle für die DynamoDB-Replikation die Berechtigungen für den Zugriff auf den AWS KMS Schlüssel entzogen, der zum Schutz von Daten in der globalen DynamoDB-Zieltabelle verwendet wird. Tabellen können bis zu 5 Minuten nach Beginn der Aktion weiter repliziert werden.

Die folgende Anweisung wird dynamisch an die Richtlinie für den AWS KMS Schlüssel angehängt, der zum Schutz von Daten in den globalen DynamoDB-Zieltabellen verwendet wird:

```
{
    "Sid": "DO_NOT_MODIFY_FIS_DDB_PAUSE_REPLICATION-EXP123456789012345",
    "Effect": "Deny",
    "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
        },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
```

```
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:aws:dynamodb:tableName": [
                "transactions-global-table",
                "inventory-global-table"
            ]
        }
    }
}
```

Mit der obigen Richtlinienanweisung werden der dienstverknüpften DynamoDB-Rolle die Berechtigungen zum Replizieren von Daten in und aus den im Kontextschlüssel aufgeführten Tabellen entzogen. `kms:EncryptionContext:aws:dynamodb:tableName` Im obigen Beispiel würde die Replikation für globale DynamoDB-Tabellen mit den Namen:`transaction-global-table`, beendet. `inventory-global-table`

Ressourcentyp

- aws:dynamodb:encrypted-global-table

Parameter

- 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

- `kms:DescribeKey`
- `kms:GetKeyPolicy`
- `kms:PutKeyPolicy`
- `dynamodb:DescribeTable`
- `dynamodb:DescribeGlobalTable`
- `tag:GetResources`

Eine Beispielrichtlinie finden Sie unter Beispiel: Experimentieren Sie mit der Rolle mit Ausführungsberechtigungen `aws:dynamodb:encrypted-global-table-pause-replication`.

> **Note**
>
> AWS FIS verwendet`kms:PutKeyPolicy`, um den Zugriff auf DynamoDB zu verweigern; auf den vom Kunden verwalteten AWS KMS Schlüssel, wodurch die Replikation gestoppt wird. Wir empfehlen, die Rolle nur zu verwenden, wenn Sie aktiv ein Experiment mit dieser Aktion ausführen. Andernfalls empfehlen wir, sie zu löschen. Durch das Löschen der Rolle werden die FIS-Berechtigungen für entfernt. `kms:PutKeyPolicy` Suchen Sie nach Abschluss des Experiments die Rolle in den Details der Experimentvorlage. Wählen Sie in der IAM-Konsole den Link zur IAM-Rolle aus und klicken Sie auf Löschen. Navigieren Sie nach dem Löschen der Rolle zur AWS KMS Konsole und suchen Sie den AWS KMS Schlüssel, der zum Schutz von Daten verwendet wird, in der DynamoDB-Zieltabelle. Stellen Sie sicher, dass die AWS KMS Schlüsselrichtlinie Ihren Erwartungen entspricht. Sie sollten keine AWS FIS Aussage mehr sehen (z. B.`FIS_DDB_PAUSE_REPLICATION-EXP123456789012345_DO_NOT_MODIFY`).

Amazon EBS-Aktionen

AWS FIS unterstützt die folgende Amazon EBS-Aktion.

aws:ebs:pause-volume-io

Unterbricht I/O-Operationen 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

Parameter

- 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, PT5S für fünf Sekunden und PT6H für sechs Stunden. In der AWS FIS Konsole geben Sie die Anzahl der Sekunden, Minuten oder Stunden ein. Wenn die Dauer kurz ist, wie z. B. PT5S, wird der I/O 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`

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 “Verwenden Sie 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.

Ressourcentyp

- aws:iam:role

Parameter

- 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.
- availabilityzoneIdentifiers— 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

Parameter

- 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.
- availabilityzoneIdentifiers— 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 RebootInstancesauf den Ziel-EC2-Instances aus.

Ressourcentyp

- aws:ec2:instance

Parameter

- None

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 von 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

Parameter

- 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 StopInstancesauf den Ziel-EC2-Instances aus.

Ressourcentyp

- aws:ec2:instance

Parameter

- 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 TerminateInstancesauf den Ziel-EC2-Instances aus.

Ressourcentyp

- aws:ec2:instance

Parameter

- None

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 leeren.

Ressourcentyp

- aws:ecs:cluster

Parameter

- 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

Parameter

- None

Berechtigungen

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

AWS verwaltete Richtlinie

- AWSFaultInjectionSimulatorECSAccess

aws:ecs:task-cpu-stress

Führt die CPU-Belastung der Zielaufgaben aus. Verwendet das AWSFISSSM-Dokument -Run-CPU-Stress. Die Aufgaben müssen von verwaltet werden. AWS Systems Manager Weitere Informationen finden Sie unter Verwenden Sie die ECS-Aufgabenaktionen.

Ressourcentyp

- aws:ecs:task

Parameter

- 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

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

aws:ecs:task-io-stress

Führt I/O-Stress für die Zielaufgaben aus. Verwendet das AWSFISSSM-Dokument -Run-IO-Stress. Die Aufgaben müssen von verwaltet werden. AWS Systems Manager Weitere Informationen finden Sie unter Verwenden Sie die ECS-Aufgabenaktionen.

Ressourcentyp

- aws:ecs:task

Parameter

- 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 Lese-/Schreibvorgängen, erzwungener Synchronisation und Löschen des Caches durch. Mehrere untergeordnete Prozesse führen unterschiedliche I/O-Operationen an derselben Datei durch. 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

- `ssm:SendCommand`

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

aws:ecs:task-kill-process

Stoppt den angegebenen Prozess in den Aufgaben mithilfe des killall Befehls. Verwendet das AWSFISSSM-Dokument -Run-Kill-Process. Die Aufgabendefinition muss auf eingestellt sein. `pidMode` `task` Die Aufgaben müssen von verwaltet werden AWS Systems Manager. Weitere Informationen finden Sie unter Verwenden Sie die ECS-Aufgabenaktionen.

Ressourcentyp

- aws:ecs:task

Parameter

- 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

- `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. Verwendet das SSM-Dokument AWSFIS-Run-Network-Blackhole-Port. Die Aufgabendefinition muss auf 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 Verwenden Sie die ECS-Aufgabenaktionen.

Ressourcentyp

- aws:ecs:task

Parameter

- duration— Die Dauer des Tests im ISO 8601-Format.
- 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 sindatd, unddig. iptables

Berechtigungen

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

aws:ecs:task-network-latency

Fügt der Netzwerkschnittstelle Latenz und Jitter hinzu, indem das tc Tool für den Datenverkehr zu oder von bestimmten Quellen verwendet wird. Verwendet das AWSFISSSM-Dokument -Run-Network-Latency-Sources. Die Aufgabendefinition muss auf eingestellt seinpidMode. `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 Verwenden Sie die ECS-Aufgabenaktionen.

Ressourcentyp

- aws:ecs:task

Parameter

- duration— Die Dauer des Tests im ISO 8601-Format.
- interface Optional. Die Netzwerkschnittstelle. Der Standardwert ist `eth0`.
- delayMilliseconds – Optional. Die Verzögerung in Millisekunden. Die Standardeinstellung ist 200.
- jitterMilliseconds Optional. Der Jitter in Millisekunden. Der Standardwert ist 10.
- sources Optional. Die Quellen, durch Kommas getrennt. Die möglichen Werte sind: eine IPv4-Adresse, ein IPv4-CIDR-Block, ein Domainname und. `DYNAMODB` `S3` Wenn Sie `DYNAMODB` oder angeben`S3`, gilt dies nur für den regionalen Endpunkt in der aktuellen Region. Die Standardeinstellung ist 0.0.0.0/0, 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 sindatd,dig, jq und. tc

Berechtigungen

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

aws:ecs:task-network-packet-loss

Fügt der Netzwerkschnittstelle mithilfe des tc Tools Paketverlust hinzu. Verwendet das AWSFISSSM-Dokument -Run-Network-Packet-Loss-Sources. Die Aufgabendefinition muss auf 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 Verwenden Sie die ECS-Aufgabenaktionen.

Ressourcentyp

- aws:ecs:task

Parameter

- duration— Die Dauer des Tests im ISO 8601-Format.
- interface Optional. Die Netzwerkschnittstelle. Der Standardwert ist `eth0`.

- lossPercent – Optional. Der Prozentsatz des Paketverlusts. Die Standardeinstellung ist 7%.
- sources Optional. Die Quellen, durch Kommas getrennt. Die möglichen Werte sind: eine IPv4-Adresse, ein IPv4-CIDR-Block, ein Domainname und. `DYNAMODB` `S3` Wenn Sie `DYNAMODB` oder angeben`S3`, gilt dies nur für den regionalen Endpunkt in der aktuellen Region. Die Standardeinstellung ist 0.0.0.0/0, 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 sindatd,dig, jq und. tc

Berechtigungen

- `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

Parameter

- 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 Verwenden Sie die EKS-Pod-Aktionen.

Ressourcentyp

- aws:eks:pod

Parameter

- 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.

Berechtigungen

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

AWS verwaltete Richtlinie

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-delete

Löscht die Ziel-Pods. Weitere Informationen finden Sie unter Verwenden Sie die EKS-Pod-Aktionen.

Ressourcentyp

- aws:eks:pod

Parameter

- 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.

Berechtigungen

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

AWS verwaltete Richtlinie

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-io-stress

Führt I/O-Stress auf den Ziel-Pods aus. Weitere Informationen finden Sie unter Verwenden Sie die EKS-Pod-Aktionen.

Ressourcentyp

- aws:eks:pod

Parameter

- duration— Die Dauer des Stresstests im Format ISO 8601.
- workers Optional. Die Anzahl der Worker. Worker führen eine Mischung aus sequentiellen, zufälligen und speicherbezogenen Lese-/Schreibvorgängen, erzwungener Synchronisation und Löschen des Caches durch. Mehrere untergeordnete Prozesse führen unterschiedliche I/O-Operationen an derselben Datei durch. 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.

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 Verwenden Sie die EKS-Pod-Aktionen.

Ressourcentyp

- aws:eks:pod

Parameter

- 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.

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. Weitere Informationen finden Sie unter Verwenden Sie die EKS-Pod-Aktionen.

Ressourcentyp

- aws:eks:pod

Parameter

- duration— Die Dauer des Tests im ISO 8601-Format.
- protocol Optional. Das Protokoll. Die möglichen Werte sind `tcp` und `udp`. Der Standardwert ist `tcp`.
- 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.

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. Weitere Informationen finden Sie unter Verwenden Sie die EKS-Pod-Aktionen.

Ressourcentyp

- aws:eks:pod

Parameter

- duration— Die Dauer des Tests im ISO 8601-Format.
- interface Optional. Die Netzwerkschnittstelle. Der Standardwert ist `eth0`.
- delayMilliseconds – Optional. Die Verzögerung in Millisekunden. Die Standardeinstellung ist 200.
- jitterMilliseconds Optional. Der Jitter in Millisekunden. Der Standardwert ist 10.
- sources Optional. Die Quellen, durch Kommas getrennt. Die möglichen Werte sind: eine IPv4-Adresse, ein IPv4-CIDR-Block, ein Domainname und. `DYNAMODB` `S3` Wenn Sie `DYNAMODB` oder angeben`S3`, gilt dies nur für den regionalen Endpunkt in der aktuellen Region. Die Standardeinstellung ist 0.0.0.0/0, was dem gesamten IPv4-Verkehr 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.

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. Weitere Informationen finden Sie unter Verwenden Sie die EKS-Pod-Aktionen.

Ressourcentyp

- aws:eks:pod

Parameter

- duration— Die Dauer des Tests im ISO 8601-Format.
- interface Optional. Die Netzwerkschnittstelle. Der Standardwert ist `eth0`.
- lossPercent – Optional. Der Prozentsatz des Paketverlusts. Die Standardeinstellung ist 7%.
- sources Optional. Die Quellen, durch Kommas getrennt. Die möglichen Werte sind: eine IPv4-Adresse, ein IPv4-CIDR-Block, ein Domainname und. `DYNAMODB` `S3` Wenn Sie `DYNAMODB` oder angeben`S3`, gilt dies nur für den regionalen Endpunkt in der aktuellen Region. Die Standardeinstellung ist 0.0.0.0/0, was dem gesamten IPv4-Verkehr 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.

Berechtigungen

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

AWS verwaltete Richtlinie

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:terminate-nodegroup-instances

Führt die Amazon EC2 EC2-API-Aktion für TerminateInstancesdie Zielknotengruppe aus.

Ressourcentyp

- aws:eks:nodegroup

Parameter

- 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:interrupt-cluster-az-power

Unterbricht die Stromversorgung der Knoten in der angegebenen Availability Zone für Redis-Zielreplikationsgruppen. 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.

Ressourcentyp

- aws:elasticache:redis-replicationgroup

Parameter

- 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`

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-connectivity

Verweigert den angegebenen Datenverkehr zu den Zielsubnetzen.

Ressourcentyp

- aws:ec2:subnet

Parameter

- scope— Die Art des Datenverkehrs, der verweigert werden soll. 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.
  - `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.
  - `vpc`— Verweigert den ein- und ausgehenden Datenverkehr in die VPC.
- 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.

Ressourcentyp

- aws:ec2:subnet

Parameter

- `region`— Der Code der Region, die isoliert werden soll (z. B. 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

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

Parameter

- `region`— Der Code der Region, die isoliert werden soll (z. B. 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

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 FailoverDBCluster auf dem Aurora-DB-Zielcluster aus.

Ressourcentyp

- aws:rds:cluster

Parameter

- None

Berechtigungen

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

AWS verwaltete Richtlinie

- AWSFaultInjectionSimulatorRDSAccess

aws:rds:reboot-db-instances

Führt die Amazon RDS-API-Aktion RebootDBInstance auf der Ziel-DB-Instance aus.

Ressourcentyp

- aws:rds:db

Parameter

- 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

Parameter

- 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 SendCommandauf 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

Parameter

- 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— Befriedigend. 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 API-Aktion aus StartAutomationExecution.

Ressourcentyp

- None

Parameter

- documentArn— Der Amazon-Ressourcenname (ARN) des Automatisierungsdokuments.
- documentVersion Optional. Die Version des Dokuments. Wenn leer, wird die Standardversion ausgeführt.
- documentParameters— Befriedigend. 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

Verwenden Sie Systems Manager SSM-Dokumente mit AWS FIS

AWS FIS unterstützt benutzerdefinierte Fehlertypen über den AWS Systems Manager SSM-Agenten und die FIS-Aktion. AWS 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 (VMs) installiert und konfiguriert werden kann. 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: https://console.aws.amazon.com/systems-manager/documents. AWS 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 Documents (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.

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, Amazon Linux
  - Ubuntu
  - REGEL 7, 8, 9
  - CentOS 7, 8, 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.

-Documents

- 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

AWSFIS-Run-CPU-Stress

Führt mithilfe des stress-ng Tools die CPU-Belastung auf einer Instanz aus. Verwendet das AWSFISSSM-Dokument -Run-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 iststress-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 AWSFISSSM-Dokument -Run-Disk-Fill.

Wenn das Experiment, bei dem dieser Fehler ausgelöst wurde, entweder manuell oder durch eine Stopp-Bedingung gestoppt 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 atd undfallocate.

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 AWSFISSSM-Dokument -Run-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 Lese-/Schreibvorgängen, erzwungener Synchronisation und Cache-Löschen ausführen. Mehrere untergeordnete Prozesse führen unterschiedliche I/O-Operationen an derselben Datei durch. 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 iststress-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 AWSFISSSM-Dokument -Run-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 istkillall.

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 AWSFISSSM-Dokument -Run-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 iststress-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 SSM-Dokument AWSFIS-Run-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, undiptables.

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 AWSFISSSM-Dokument -Run-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, undtc.

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 AWSFISSSM-Dokument -Run-Network-Latency-Sources.

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 Netzwerkschnittstelle. Der Standardwert ist `eth0`.
- DelayMilliseconds – Optional. Die Verzögerung in Millisekunden. Die Standardeinstellung ist 200.
- JitterMilliseconds Optional. Der Jitter in Millisekunden. Der Standardwert ist 10.
- Sources – Erforderlich. Die Quellen, durch Kommas getrennt. Die möglichen Werte sind: eine IPv4-Adresse, ein IPv4-CIDR-Block, ein Domainname 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 sind atddig,jq, undtc.

Im Folgenden finden Sie 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 AWSFISSSM-Dokument -Run-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 atddig, 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 AWSFISSSM-Dokument -Run-Network-Packet-Loss-Sources.

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 Netzwerkschnittstelle. Der Standardwert ist `eth0`.

- LossPercent – Optional. Der Prozentsatz des Paketverlusts. Die Standardeinstellung ist 7%.
- Sources – Erforderlich. Die Quellen, durch Kommas getrennt. Die möglichen Werte sind: eine IPv4-Adresse, ein IPv4-CIDR-Block, ein Domainname 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, undtc.

Im Folgenden finden Sie 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 unterthe section called “Ausführen eines vorkonfigurierten AWS FIS-SSM-Dokuments”.

Ein Tutorial finden Sie unter CPU-Auslastung auf einer Instance ausführen.

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.

Diese Aktionen verwenden einen SSM-Agenten als Sidecar-Container, um SSM-Dokumente auszuführen, die die Fehlerinjektion durchführen, und registrieren Amazon ECS-Aufgaben über den Sidecar-Container als SSM-verwaltete Instances. Um diese Aktionen verwenden zu können, müssen Sie Ihre Amazon ECS-Aufgabendefinitionen aktualisieren, um den SSM-Agenten als Sidecar-Container hinzuzufügen, sodass er die Aufgabe, bei der er ausgeführt wird, als SSM-verwaltete Instance registriert. 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, das der verwalteten Instance hinzugefügt wird`ECS_TASK_ARN`, einer Gruppe von SSM-verwalteten Instances zu. Der Tag-Wert ist der ARN der zugehörigen Amazon ECS-Aufgabe, in der die SSM-Dokumente ausgeführt werden sollen, und sollte daher bei der Ausführung des Experiments nicht entfernt werden.

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 funktionieren nicht mit AWS Fargate:
  - aws:ecs:task-kill-process
  - aws:ecs:task-network-blackhole-port
  - aws:ecs:task-network-latency
  - aws:ecs:task-network-packet-loss
- Wenn Sie ECS Exec aktiviert haben, müssen Sie es deaktivieren, bevor Sie diese Aktionen verwenden können.

Voraussetzungen

- Fügen Sie der AWS FIS-Experimentrolle die folgenden Berechtigungen hinzu:
  - `ssm:SendCommand`
  - `ssm:ListCommands`
  - `ssm:CancelCommand`
- Fügen Sie der Amazon ECS-Task-IAM-Rolle die folgenden Berechtigungen hinzu:
  - `ssm:CreateActivation`
  - `ssm:AddTagsToResource`
  - `iam:PassRole`

  Beachten Sie, dass Sie den ARN der verwalteten Instanzrolle als Ressource für angeben können`iam:PassRole`.
- Erstellen Sie eine IAM-Rolle für die Ausführung von Amazon ECS-Aufgaben und fügen Sie die TaskExecutionRolePolicy verwaltete AmazonECS-Richtlinie hinzu.
- Fügen Sie der verwalteten Instance-Rolle, die Aufgaben zugewiesen ist, die als verwaltete Instances registriert sind, die folgenden Berechtigungen hinzu:
  - `ssm:DeleteActivation`
  - `ssm:DeregisterManagedInstance`
- Fügen Sie die verwaltete AmazonSSM-Richtlinie der Rolle für ManagedInstanceCore verwaltete Instanzen hinzu, die Aufgaben zugeordnet sind, die als verwaltete Instances registriert sind.
- Setzen Sie die Umgebungsvariable `MANAGED_INSTANCE_ROLE_NAME` auf den Namen der verwalteten Instance-Rolle.
- Fügen Sie der ECS-Aufgabendefinition einen SSM-Agent-Container hinzu. Das Befehlsskript registriert ECS-Aufgaben als verwaltete Instanzen.

```
{
    "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; yum upgrade -y; yum 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_SIDECAR,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 unterthe section called “Referenzversion des Skripts”.

- Wenn Sie die `aws:ecs:task-network-packet-loss` Aktionen `aws:ecs:task-network-blackhole-portaws:ecs:task-network-latency`, und verwenden, müssen Sie den SSM-Agent-Container in der ECS-Aufgabendefinition mit einer der folgenden Optionen aktualisieren.
  - Option 1 — Fügen Sie die spezifische Linux-Funktion hinzu.

    ```
    "linuxParameters": {
        "capabilities": {
            "add": [
                "NET_ADMIN"
            ]
        }
    },
    ```

  - Option 2 — Fügen Sie alle Linux-Funktionen hinzu.

    ```
    "privileged": true,
    ```

- Wenn Sie die `aws:ecs:task-network-packet-loss` Aktionen `aws:ecs:task-kill-processaws:ecs:task-network-blackhole-port`,`aws:ecs:task-network-latency`, und verwenden, muss die ECS-Aufgabendefinition auf `pidMode` eingestellt sein`task`.

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 form 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_SIDECAR: 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

yum upgrade -y
yum 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_SIDECAR,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.

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
- Sie können in Ihrer Experimentvorlage keine Ziele vom Typ aws:eks:pod mithilfe von Ressourcen-ARNs oder Ressourcen-Tags identifizieren. 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 erreicht`maxErrors`, 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 mit unterschiedlichen Verkehrsarten werden unterstützt.
- FIS kann den Status der Fehlerinjektion nur überwachen oder überwachen, wenn `securityContext` der Ziel-Pods auf eingestellt ist. `readOnlyRootFilesystem: false` 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 Servicerolle für das Kubernetes-Dienstkonto

Erstellen Sie eine IAM-Rolle, die als Servicerolle verwendet werden soll. Weitere Informationen finden Sie unter the section called “Experimentrolle”.

Das Kubernetes-Servicekonto konfigurieren

Konfigurieren Sie ein Kubernetes-Dienstkonto, um Experimente mit Zielen im angegebenen Kubernetes-Namespace durchzuführen. *Im folgenden Beispiel ist das Dienstkonto myserviceaccount und der Namespace ist Standard.* 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
```

Ordnen Sie Ihre Experimentrolle dem Kubernetes-Benutzer zu

Verwenden Sie den folgenden Befehl, 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
```

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.

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

AWS-Region	Image-URI
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
Europa (Stockholm)	263175118295.dkr.ecr.eu-north-1.amazonaws.com/aws-fis-pod:0.1
Naher Osten (Bahrain)	065825543785.dkr.ecr.me-south-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"
    }
}
```

Listen Sie die AWS FIS Aktionen mit dem auf AWS CLI

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um Informationen zu den Aktionen anzuzeigen, die AWS FIS unterstützt werden.

Voraussetzung

Installieren Sie das AWS CLI auf Ihrem Computer. Informationen zu den ersten Schritten finden Sie im AWS Command Line Interface -Benutzerhandbuch. Weitere Informationen zu den Befehlen für AWS FIS finden Sie unter fis in der AWS CLI Befehlsreferenz.

Beispiel: Listet die Namen aller Aktionen auf

Sie können die Namen aller Aktionen mit dem Befehl list-actions wie folgt auflisten.

```
aws fis list-actions --query "actions[*].[id]" --output text | sort
```

Es folgt eine Beispielausgabe.

```
aws:cloudwatch:assert-alarm-state
aws:dynamodb:encrypted-global-table-pause-replication
aws:ebs:pause-volume-io
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:ecs:drain-container-instances
aws:ecs:stop-task
aws:eks:inject-kubernetes-custom-resource
aws:eks:terminate-nodegroup-instances
aws:elasticache:interrupt-cluster-az-power
aws:fis:inject-api-internal-error
aws:fis:inject-api-throttle-error
aws:fis:inject-api-unavailable-error
aws:fis:wait
```

```
aws:network:disrupt-connectivity
aws:network:route-table-disrupt-cross-region-connectivity
aws:network:transit-gateway-disrupt-cross-region-connectivity
aws:rds:failover-db-cluster
aws:rds:reboot-db-instances
aws:s3:bucket-pause-replication
aws:ssm:send-command
aws:ssm:start-automation-execution
```

Beispiel: Informationen zu einer Aktion anzeigen

Nachdem Sie den Namen einer Aktion gefunden haben, können Sie detaillierte Informationen zu der Aktion mit dem Befehl get-action wie folgt anzeigen.

```
aws fis get-action --id aws:ec2:reboot-instances
```

Es folgt eine Beispielausgabe.

```
{
    "action": {
        "id": "aws:ec2:reboot-instances",
        "description": "Reboot the specified EC2 instances.",
        "targets": {
            "Instances": {
                "resourceType": "aws:ec2:instance"
            }
        },
        "tags": {}
    }
}
```

Experimentvorlagen für AWS FIS

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. Nachdem Sie eine Versuchsvorlage erstellt haben, können Sie sie verwenden, um ein Experiment durchzuführen.

Komponenten der Vorlage

Sie verwenden die folgenden Komponenten, um Versuchsvorlagen zu erstellen:

Aktionssatz

Die AWS FIS-Aktionen, die Sie ausführen möchten. Aktionen können in einer von Ihnen festgelegten Reihenfolge oder gleichzeitig ausgeführt werden. Weitere Informationen finden Sie unter Aktionssatz.

Targets (Ziele)

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

Bedingungen beenden

Die CloudWatch Alarme, 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 Stoppbedingungen.

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 Experimentrolle.

Optionen für Experimente

Optionen für die Experimentvorlage. Weitere Informationen finden Sie unter Experimentieroptionen.

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",
            "experimentOptions":{},
            "tags": {}
        }
```

Beispiele finden Sie unter Beispielvorlagen.

Erste Schritte

Informationen zum Erstellen einer Versuchsvorlage mit dem AWS Management Console finden Sie unterErstellen Sie eine Versuchsvorlage.

Informationen zum Erstellen einer Experimentvorlage mit dem AWS CLI finden Sie unterBeispielvorlagen für AWS FIS-Experimente.

Aktionssatz für AWS FIS

Um eine Versuchsvorlage zu erstellen, müssen Sie eine oder mehrere Aktionen definieren, aus denen das Aktionssatz besteht. Eine Liste der von AWS FIS bereitgestellten vordefinierten Aktionen finden Sie unterAktionen.

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

- Beispielaktionen

Syntax der Aktion

Im Folgenden finden Sie die Syntax für einen Aktionssatz.

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

Wenn Sie eine Aktion definieren, geben Sie Folgendes an:

Aktionsname

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”.

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 haben`skip`, 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 fast sofort auftreten, wie z. B. das 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

Beispiel: Stoppen Sie EC2-Instances

Die folgende Aktion stoppt die EC2-Instances, die mithilfe des Ziels TargetInstances identifiziert wurden. 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: EKS-Mitarbeiter kündigen

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"
        }
    }
}
```

Ziele für AWS FIS

Ein Ziel ist eine oder mehrere AWS Ressourcen, für die eine Aktion von AWS Fault Injection Service (AWS FIS) während eines Experiments ausgeführt wird. Ziele können sich im selben AWS-Konto wie das Experiment oder in einem anderen Konto befinden, indem ein Experiment mit mehreren Konten verwendet wird. Weitere Informationen zum Targeting von Ressourcen in einem anderen Konto finden Sie unter Experimente mit mehreren Konten.

Sie definieren Ziele, wenn Sie eine Experimentvorlage 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

Im Folgenden finden Sie 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 werden.

parameters

Die Parameter, die Ziele mithilfe bestimmter Attribute identifizieren.

filters

Die Ressourcenfilter schränken die identifizierten Zielressourcen mithilfe bestimmter Attribute ein.

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 das Ziel der von der Aktion unterstützte Ressourcentyp sein.

Die folgenden Ressourcentypen werden von AWS FIS unterstützt:

- aws:dynamodb:encrypted-global-table – Eine globale Tabelle, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist
- aws:ec2:autoscaling-group – Eine Amazon EC2 Auto Scaling-Gruppe
- aws:ec2:ebs-volume – Ein Amazon-EBS-Volume
- aws:ec2:instance – Eine Amazon EC2Instance
- aws:ec2:spot-instance – Eine Amazon EC2-Spot-Instance
- aws:ec2:subnet – Ein Amazon-VPC-Subnetz
- aws:ec2:transit-gateway – Ein Transit-Gateway
- 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:redis-replicationgroup – Eine ElastiCache Redis-Replikationsgruppe
- aws:iam:role – Eine IAM-Rolle
- aws:rds:cluster – Ein Amazon Aurora-DB-Cluster
- aws:rds:db – Eine Amazon RDS-DB-Instance
- aws:s3:bucket – Ein Amazon S3-Bucket

Identifizieren von Zielressourcen

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

Um Ihre Zielressourcen zu identifizieren, können Sie Folgendes angeben:

- Ressourcen-IDs – Die Ressourcen-IDs bestimmter AWS Ressourcen. Alle Ressourcen-IDs müssen denselben Ressourcentyp darstellen.
- Ressourcen-Tags – Die Tags, die auf bestimmte AWS Ressourcen angewendet werden.
- Ressourcenfilter – Der Pfad und die Werte, die Ressourcen mit bestimmten Attributen darstellen. Weitere Informationen finden Sie unter Ressourcenfilter.

- Ressourcenparameter – Die Parameter, die Ressourcen darstellen, die bestimmte Kriterien erfüllen. Weitere Informationen finden Sie unter Ressourcenparameter.

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 es keinem Platzhalter. Sie gleicht Ressourcen ab, die ein Tag mit dem angegebenen Tag-Schlüssel und einem leeren Tag-Wert haben.

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 enthält, entsprechend dem von Ihnen angegebenen Ressourcentyp. Ressourcen mit Attributen, die mit der Abfrage übereinstimmen, sind in der Zieldefinition enthalten.

Jeder Filter wird als Attributpfad und mögliche Werte ausgedrückt. Ein Pfad ist eine Folge von Elementen, die durch Punkte getrennt sind und den Pfad beschreiben, um ein Attribut in der Ausgabe der Aktion Beschreiben für eine Ressource zu erreichen. Jedes Element muss in Pascal-Fall ausgedrückt werden, auch wenn die Ausgabe der Describe-Aktion für eine Ressource in Camel-Fall vorliegt. Sie sollten beispielsweise verwenden`AvailabilityZone`, nicht `availablityZone` als Attributelement.

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

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 kann die Dokumentation für DescribeInstances Status, die kürzlich beendete Instances haben, in den Ergebnissen erscheinen.

Ressourcentyp	API-Aktion	AWS CLI -Befehl
aws:ec2:autoscaling-group	DescribeAutoScalingGroups	describe-auto-scaling-groups
aws:ec2:ebs-volume	DescribeVolumes	describe-volumes
aws:ec2:instance	DescribeInstances	describe-instances
aws:ec2:subnet	DescribeSubnets	describe-subnets
aws:ec2:transit-gateway	DescribeTransitGateways	describe-transit-gateways
aws:ecs:cluster	DescribeClusters	describe-clusters
aws:ecs:task	DescribeTasks	describe-tasks
aws:eks:cluster	DescribeClusters	describe-clusters
aws:eks:nodegroup	DescribeNodegroup	describe-nodegroup
aws:elasticache:redis-replicationgroup	DescribeReplicationGroups	describe-replication-groups
aws:iam:role	ListRoles	list-roles
aws:rds:cluster	DescribeDBClusters	describe-db-clusters
aws:rds:db	DescribeDBInstances	describe-db-instances
aws:s3:bucket	ListBuckets	list-buckets

Die folgende Logik gilt für alle Ressourcenfilter:

- Werte innerhalb eines Filters – `OR`
- Werte über Filter hinweg – `AND`

Beispiele finden Sie unter the section called “Beispielfilter”.

Ressourcenparameter

Ressourcenparameter identifizieren Zielressourcen gemäß bestimmten Kriterien.

Der folgende Ressourcentyp unterstützt Parameter.

aws:ec2:ebs-volume

- `availabilityZoneIdentifier` – Der Code (z. B. us-east-1a) der Availability Zone, die die Ziel-Volumes enthält.

aws:ec2:subnet

- `availabilityZoneIdentifier` – 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 Service, der die Zielaufgaben enthält.

aws:eks:pod

- `availabilityZoneIdentifier` Optional. Die Availability Zone, die die Ziel-Pods enthält. Beispiel: `us-east-1d` Wir bestimmen die Availability Zone eines Pods, indem wir seine hostIP und das CIDR des Cluster-Subnetzes vergleichen.
- `clusterIdentifier` – 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 `abselectorType`.
- `targetContainerName` Optional. Der Name des Zielcontainers, wie in der Pod-Spezifikation definiert. Der Standardwert ist der erste Container, der in jeder Ziel-Pod-Spezifikation definiert ist.

aws:rds:cluster

- `writerAvailabilityZoneIdentifiers` Optional. Die Availability Zones des Writers des DB-Clusters. Mögliche Werte sind: eine kommagetrennte Liste von Availability Zone-Kennungen, `all`.

aws:rds:db

- `availabilityZoneIdentifiers` Optional. Die Availability Zones der DB-Instance, die betroffen sein sollen. Mögliche Werte sind: eine kommagetrennte Liste von Availability Zone-Kennungen, `all`.

aws:elasticache:redis-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` – Führen Sie die Aktion für alle Ziele aus.
- `COUNT(n)` – Führen Sie die Aktion für die angegebene Anzahl von Zielen aus, die zufällig aus den identifizierten Zielen ausgewählt wurden. Beispielsweise wählt COUNT(1) eines der identifizierten Ziele aus.
- `PERCENT(n)` – Führen Sie die Aktion für den angegebenen Prozentsatz von Zielen aus, die zufällig aus den identifizierten Zielen ausgewählt wurden. PERCENT(25) wählt beispielsweise 25 % der identifizierten Ziele aus.

Wenn Sie eine ungerade Anzahl von Ressourcen haben und 50 % angeben, rundet AWS FIS ab. Wenn Sie beispielsweise fünf Amazon EC2-Instances als Ziele und Bereich zu 50 % hinzufügen, rundet AWS FIS auf zwei Instances ab. Sie können keinen Prozentsatz angeben, der kleiner als eine Ressource ist. Wenn Sie beispielsweise vier Amazon EC2-Instances und einen Bereich zu 5 % hinzufügen, 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

- Instances 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 Amazon EC2-Instances in der angegebenen VPC mit dem Tag env=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-Instances

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

Der describe-instances Befehl gibt die JSON-Ausgabe zurück, wobei jede Instance eine Struktur unter ist`Instances`. Im Folgenden finden Sie eine Teilausgabe, die Felder enthält, die mit *kursiv* markiert sind. Wir stellen Beispiele bereit, die diese Felder verwenden, um einen Attributpfad aus der Struktur der JSON-Ausgabe anzugeben.

```
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "ImageId": "ami-00111111111111111",
                    "InstanceId": "i-00aaaaaaaaaaaaaa",
```

```
                    "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-00bbbbbbbbbbbbbbb",
                    ...
                },
                ...
                {
                    ...
                }
            ],
            "OwnerId": "123456789012",
            "ReservationId": "r-aaaaaabbbbb111111"
        },
        ...
    ]
}
```

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

```
"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. Beispielsweise:

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

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

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

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-RDS-describe-db-clustersBefehl aus und wendet den Filter an, um die Ziele zu identifizieren.

Der describe-db-clusters Befehl gibt für jeden DB-Cluster eine JSON-Ausgabe ähnlich der folgenden zurück. Im Folgenden finden Sie eine Teilausgabe, die Felder enthält, die mit *`kursiv`* markiert sind. Wir stellen Beispiele bereit, 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 `aurora-postgresql` wie im folgenden Beispiel gezeigt an.

```
"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 Fault Injection Service (AWS FIS) bietet Kontrollen und Integritätsschutz, mit denen Sie Experimente sicher auf AWSWorkloads ausführen können. Eine Stoppbedingung ist ein Mechanismus zum Stoppen eines Experiments, wenn sie einen Schwellenwert erreicht, den Sie als Amazon- CloudWatch Alarm definieren. Wenn während eines Experiments eine Stoppbedingung ausgelöst wird, stoppt AWS FIS das Experiment. Sie können ein gestopptes Experiment nicht fortsetzen.

Um eine Stopp-Bedingung zu erstellen, definieren Sie zunächst den stabilen Zustand für Ihre Anwendung oder Ihren Service. Der Steady State ist , wenn Ihre Anwendung optimal funktioniert, definiert in Bezug auf geschäftliche oder technische Metriken. Zum Beispiel Latenz, CPU-Last oder Anzahl der Wiederholungen. Sie können den stabilen Zustand verwenden, um einen CloudWatch Alarm zu erstellen, mit dem Sie ein Experiment stoppen können, wenn Ihre Anwendung oder Ihr Service einen Zustand erreicht, in dem ihre Leistung nicht akzeptabel ist. Weitere Informationen finden Sie unter Verwenden von Amazon- CloudWatch Alarmen im Amazon- CloudWatch Benutzerhandbuch.

Ihr Konto verfügt über ein Kontingent für die Anzahl der Stoppbedingungen, die Sie in einer Experimentvorlage angeben können. Weitere Informationen finden Sie unter Kontingente und Einschränkungen für AWS Fault Injection Service.

Syntax der Stoppbedingung

Wenn Sie eine Experimentvorlage erstellen, geben Sie eine oder mehrere Stoppbedingungen an, indem Sie die von Ihnen erstellten CloudWatch Alarme angeben.

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

Das folgende Beispiel zeigt, dass die Experimentvorlage keine Stoppbedingung angibt.

```
{
```

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

Weitere Informationen

Ein Tutorial, das zeigt, wie Sie einen CloudWatch Alarm erstellen und einer Experimentvorlage eine Stoppbedingung hinzufügen, finden Sie unter CPU-Auslastung auf einer Instance ausführen.

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

- Überwachen Ihrer Instances mit CloudWatch
- Amazon-ECS- CloudWatch Metriken
- Überwachen von Amazon-RDS-Metriken mit CloudWatch
- Überwachen von Run Command-Metriken mit CloudWatch

IAM-Rollen für AWS FIS-Experimente

AWS Identity and Access Management (IAM) ist ein AWS-Service, mit dem ein Administrator den Zugriff auf AWS-Ressourcen sicher steuern kann. Um AWS FIS verwenden zu können, müssen Sie eine IAM-Rolle erstellen, die AWS FIS die erforderlichen Berechtigungen gewährt, damit AWS FIS Experimente in Ihrem Namen ausführen 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 Berechtigung erteilen, die Ressourcen zu ändern, die Sie als Ziele in Ihrer Experimentvorlage angeben. Für ein Experiment mit mehreren Konten muss die Experimentrolle der Orchestratorrolle die Berechtigung erteilen, die IAM-Rolle für jedes Zielkonto zu übernehmen. Weitere Informationen finden Sie unter Berechtigungen für Experimente mit mehreren Konten.

Wir empfehlen Ihnen, die Standardsicherheitsmethode der Erteilung der geringsten Rechte zu befolgen. Sie können dies tun, indem Sie bestimmte Ressourcen-ARNs oder Tags in Ihren Richtlinien angeben.

Um Ihnen den schnellen Einstieg in AWS FIS zu erleichtern, stellen wir AWS verwaltete Richtlinien bereit, die Sie beim Erstellen einer Experimentrolle angeben können. Alternativ können Sie diese

Richtlinien auch als Modell verwenden, wenn Sie Ihre eigenen eingebundenen Richtliniendokumente erstellen.

Inhalt

Voraussetzungen

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

AWS CLI installieren

Bevor Sie beginnen, installieren und konfigurieren Sie die AWS CLI. Wenn Sie den AWS CLI konfigurieren, 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 einer Vertrauensbeziehungsrichtlinie

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

```
{
    "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/*"
    }
}
```

Hinzufügen von Berechtigungen zur Übernahme von Zielkontorollen (nur Experimente mit mehreren Konten)

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

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

Option 1: Erstellen einer Experimentrolle und Anfügen einer von AWS verwalteten Richtlinie

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

So erstellen Sie eine Experimentrolle und fügen eine von AWS verwaltete Richtlinie an

1. Stellen Sie sicher, dass es eine verwaltete Richtlinie für die AWS FIS-Aktionen in Ihrem Experiment 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 create-role-Befehl, 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 anzufügen.

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

   Dabei *fis-policy-arn* ist einer der folgenden Werte:

   - 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 einer Experimentrolle und Hinzufügen eines eingebundenen Richtliniendokuments

Verwenden Sie diese Option für Aktionen, die keine verwaltete Richtlinie haben, oder um nur die Berechtigungen aufzunehmen, die für Ihr spezifisches Experiment erforderlich sind.

So erstellen Sie ein Experiment und fügen ein Inline-Richtliniendokument hinzu

1. Verwenden Sie den folgenden create-role-Befehl, 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 Fehlersimulation – Beginnen Sie mit der folgenden Richtlinie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFISExperimentRoleFaultInjectionActions",
            "Effect": "Allow",
            "Action": [
                "fis:InjectApiInternalError",
                "fis:InjectApiThrottleError",
                "fis:InjectApiUnavailableError"
            ],
            "Resource": "arn:*:fis:*:*:experiment/*"
        }
    ]
}
```

   - Amazon-EBS-Aktionen – Beginnen Sie mit der folgenden Richtlinie.

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

```
            "Resource": "arn:aws:ec2:*:*:volume/*"
        }
    ]
}
```

- Amazon EC2-Aktionen – Beginnen Sie mit der AWSFaultInjectionSimulatorEC2Access Richtlinie.
- 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-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
```

Experimentieroptionen

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 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

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 besitzt die AWS FIS Versuchsvorlage und das Experiment. Ein Zielkonto ist ein einzelnes AWS-Konto mit Ressourcen, die durch ein AWS FIS Experiment beeinträchtigt werden können. Weitere Informationen finden Sie unter Experimente mit mehreren Konten für AWS FIS.

Sie verwenden das Konto-Targeting, um den Standort Ihrer Zielressourcen anzugeben. Sie können zwei Werte für das Konten-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. Die Konto-IDs der Zielkontokonfigurationen für eine Versuchsvorlage müssen eindeutig sein.

Wenn Sie eine Versuchsvorlage mit mehreren Konten erstellen, gibt die Experimentvorlage ein schreibgeschütztes Feld zurück`targetAccountConfigurationsCount`, 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 unterthe section called “Arbeiten mit Experimenten mit mehreren Konten”.

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 mithilfe eindeutiger Kennungen wie ARNs definiert wurden, können nicht übersprungen werden. Wenn ein Ziel, das mit einer eindeutigen Kennung definiert wurde, nicht gefunden wird, wird das Experiment sofort mit dem Status von beendet `failed`

Aktionsmodus

Der Aktionsmodus ist ein optionaler Parameter, den Sie angeben können, wenn Sie ein Experiment starten. Sie können den Aktionsmodus auf einstellen, `skip-all` 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 unterGenerieren Sie eine Zielvorschau aus einer Experimentvorlage.

Arbeiten Sie mit AWS FIS-Versuchsvorlagen

Sie können Versuchsvorlagen mit der AWS FIS-Konsole oder der Befehlszeile erstellen und verwalten. Nachdem Sie eine Experimentvorlage erstellt haben, können Sie sie verwenden, um ein Experiment auszuführen.

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

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. (Optional) Wählen Sie für die Ausrichtung auf Konten die Option Mehrere Konten aus, um eine Versuchsvorlage für mehrere Konten zu konfigurieren.
5. Wählen Sie für die Kontoausrichtung die Option Bestätigen aus.

6. Geben Sie unter Beschreibung und Name eine Beschreibung und einen Namen für die Vorlage ein.
7. 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.
8. 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 Ressourcen-IDs und wählen Sie dann die Ressourcen-IDs aus oder fügen Sie sie hinzu.

- 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 von 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.

9. Um eine Aktion mit dem von Ihnen erstellten Ziel zu aktualisieren, suchen Sie die Aktion unter Aktionen, wählen Sie Bearbeiten aus, und aktualisieren Sie dann Ziel. Sie können dasselbe Ziel für mehrere Aktionen verwenden.
10. (Nur Experimente mit mehreren Konten) Fügen Sie für Target-Kontokonfigurationen einen Rollen-ARN und eine optionale Beschreibung für jedes Zielkonto hinzu. Um die ARNs der Zielkontorolle mit einer CSV-Datei hochzuladen, wählen Sie Rollen-ARNs für alle Zielkonten hochladen und wählen Sie dann CSV-Datei auswählen
11. 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 “Experimentrolle”.
12. (Optional) Wählen Sie unter Stoppbedingungen die CloudWatch Amazon-Alarme für die Stoppbedingungen aus. Weitere Informationen finden Sie unter Stoppbedingungen für AWS FIS.
13. (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.
14. (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.
15. Wählen Sie Experimentvorlage erstellen. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den **`create`** Text ein und wählen Sie Experimentvorlage erstellen.

So erstellen Sie eine Experimentvorlage mit der CLI

Verwenden Sie den create-experiment-template-Befehl.

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 unterBeispielvorlagen für AWS FIS-Experimente.

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.
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-templatesBefehl, um eine Liste von Experimentvorlagen abzurufen, und verwenden Sie den get-experiment-templateBefehl, 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 dienstbezogene Rollen für den Fault Injection AWS 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. (Optional) Um eine Vorschau zur Überprüfung Ihrer Ziele zu generieren:
   - Wählen Sie Ziele aus.
   - Wählen Sie „Vorschau generieren“.
4. Wählen Sie die Experimentvorlage aus und klicken Sie auf Experiment starten.
5. (Optional) Um Ihrem Experiment ein Tag hinzuzufügen, wählen Sie Neues Tag hinzufügen und geben Sie einen Tag-Schlüssel und einen Tag-Wert ein.
6. 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 update-experiment-template-Befehl.

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 delete-experiment-template-Befehl.

Beispielvorlagen für AWS FIS-Experimente

Wenn Sie die AWS FIS-API oder ein Befehlszeilen-Tool 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 Komponenten der Vorlage.

Um ein Experiment mit einer der Beispielvorlagen zu erstellen, speichern Sie es in einer JSON-Datei (z. B. `my-template.json`), ersetzen Sie die *kursiv gedruckten* Platzhalterwerte 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
```

Beispielvorlagen

- Anhalten von EC2-Instances basierend auf Filtern
- Anhalten einer bestimmten Anzahl von EC2-Instances
- Ausführen eines vorkonfigurierten AWS FIS-SSM-Dokuments
- Ausführen eines vordefinierten Automation-Runbooks
- Drosselungs-API-Aktionen auf EC2-Instances mit der Ziel-IAM-Rolle
- CPU-Auslastungstest von Pods in einem Kubernetes-Cluster

Anhalten von EC2-Instances basierend auf Filtern

Im folgenden Beispiel werden alle ausgeführten Amazon EC2-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"
}
```

Anhalten einer bestimmten Anzahl von EC2-Instances

Im folgenden Beispiel werden drei Instances mit dem angegebenen Tag gestoppt. AWS FIS wählt die spezifischen Instances aus, die nach dem Zufallsprinzip angehalten werden sollen. Diese Instances 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"
}
```

Ausführen eines vorkonfigurierten AWS FIS-SSM-Dokuments

Im folgenden Beispiel wird 60 Sekunden lang eine CPU-Fehlersimulation auf der angegebenen EC2-Instance unter Verwendung eines vorkonfigurierten AWS FIS-SSM-Dokuments, AWSFIS-Run-CPU-Stress, ausgeführt. 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"
}
```

Ausführen eines vordefinierten Automation-Runbooks

Im folgenden Beispiel wird eine Benachrichtigung an Amazon SNS unter Verwendung eines Runbooks veröffentlicht, das von Systems Manager, AWS-PublishSNSNotification, bereitgestellt wird. Die Rolle muss über Berechtigungen zum Veröffentlichen von Benachrichtigungen für das angegebene 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"
}
```

Drosselungs-API-Aktionen auf EC2-Instances mit der Ziel-IAM-Rolle

Im folgenden Beispiel werden 100 % der Aufrufe an die angegebenen API-Aktionen auf EC2-Instances mit der angegebenen IAM-Rolle gedrosselt.

```
{
    "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"
}
```

CPU-Auslastungstest von Pods in einem Kubernetes-Cluster

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

```
{
```

```
    "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\"],
\"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 zu testen.

```
{
    "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": {}
}
```

Experimente mit mehreren Konten für AWS FIS

Mit einem Experiment mit mehreren Konten können Sie reale Fehlerszenarien für eine Anwendung einrichten und ausführen, die sich über mehrere AWS Konten innerhalb einer Region erstreckt. Sie führen Experimente mit mehreren Konten von einem Orchestratorkonto aus, das 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 Benutzer in den Zielkonten darauf hingewiesen werden. Mit Experimenten mit mehreren Konten können Sie:

- Führen Sie reale Fehlerszenarien für Anwendungen aus, die sich über mehrere Konten erstrecken, mit den zentralen Kontrollen und Integritätsschutz, die AWS FIS bietet.
- Kontrollieren Sie die Auswirkungen eines Experiments mit mehreren Konten mithilfe von IAM-Rollen mit detaillierten Berechtigungen und Tags, um den Umfang jedes Ziels zu definieren.
- Zeigen Sie die Aktionen AWS FIS in jedem Konto zentral über die AWS Management Console und über AWS FIS Protokolle an.
- Überwachen und prüfen AWS FIS Sie API-Aufrufe in jedem Konto mit AWS CloudTrail.

Dieser Abschnitt hilft Ihnen bei den ersten Schritten mit Experimenten mit mehreren Konten.

Themen

Konzepte für Experimente mit mehreren Konten

Im Folgenden sind die wichtigsten Konzepte für Experimente mit mehreren Konten aufgeführt:

Orchestrator-Konto

Das Orchestratorkonto fungiert als zentrales Konto, um das Experiment in der - AWS FIS Konsole zu konfigurieren und zu verwalten sowie die Protokollierung zu zentralisieren. Das Orchestratorkonto besitzt die AWS FIS Experimentvorlage und das Experiment.

Zielkonten

Ein Zielkonto ist ein einzelnes AWS-Konto mit Ressourcen, die von einem Experiment mit AWS FIS mehreren Konten betroffen sein können.

Konfigurationen des Zielkontos

Sie definieren die Zielkonten, die Teil eines Experiments sind, indem Sie der Experimentvorlage Zielkontokonfigurationen hinzufügen. Eine Zielkontokonfiguration ist ein Element der Experimentvorlage, die 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.

Voraussetzungen für Experimente mit mehreren Konten

Um Stoppbedingungen für ein Experiment mit mehreren Konten zu verwenden, müssen Sie zunächst kontoübergreifende Alarme konfigurieren. IAM-Rollen werden definiert, wenn Sie eine Experimentvorlage mit mehreren Konten erstellen. Sie können die erforderlichen IAM-Rollen erstellen, bevor Sie die Vorlage erstellen.

Inhalt

Berechtigungen für Experimente mit mehreren Konten

Experimente mit mehreren Konten verwenden die IAM-Rollenverkettung, um Berechtigungen zu erteilen AWS FIS , Aktionen für Ressourcen in Zielkonten durchzuführen. Für Experimente mit mehreren Konten richten Sie IAM-Rollen in jedem Zielkonto und dem Orchestratorkonto ein. Diese IAM-Rollen erfordern eine Vertrauensstellung zwischen den Zielkonten und dem Orchestratorkonto sowie zwischen dem Orchestratorkonto und AWS FIS.

Die IAM-Rollen für die Zielkonten enthalten die Berechtigungen, die erforderlich sind, um Aktionen für -Ressourcen durchzuführen, und werden für eine Experimentvorlage erstellt, indem Zielkontokonfigurationen hinzugefügt werden. Sie erstellen eine IAM-Rolle für das Orchestrator-Konto mit der Berechtigung, die Rollen von Zielkonten zu übernehmen und eine Vertrauensstellung mit herzustellen AWS FIS. Diese IAM-Rolle wird als `roleArn` für die Experimentvorlage verwendet.

Weitere Informationen zur Rollenverkettung finden Sie unter Rollenbegriffe und -konzepte im IAM-Benutzerhandbuch.

Im folgenden Beispiel richten Sie Berechtigungen für ein Orchestratorkonto A ein, um ein Experiment mit `aws:ebs:pause-volume-io` im Zielkonto B auszuführen.

1. Erstellen Sie in Konto B eine IAM-Rolle mit den Berechtigungen, die zum Ausführen der Aktion erforderlich sind. Die für jede Aktion erforderlichen Berechtigungen finden Sie unter the section called “Referenz zu Aktionen”. Das folgende Beispiel zeigt die Berechtigungen, die ein Zielkonto zum Ausführen der EBS-Pause-Volume-IO-Aktion erteiltthe section called “aws:ebs:pause-volume-io”.

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

2. Fügen Sie als Nächstes eine Vertrauensrichtlinie in Konto B hinzu, die eine Vertrauensstellung mit Konto A erstellt. Wählen Sie einen Namen für die IAM-Rolle für Konto A aus, die Sie in Schritt 3 erstellen werden.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
           "Effect": "Allow",
            "Principal": {
                "AWS": "AccountIdA"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringLike":{
                    "sts:ExternalId": "arn:aws:fis:region:accountIdA:experiment/*"
                },
                "ArnEquals": {
                    "aws:PrincipalArn": "arn:aws:iam::accountIdA:role/role_name"
                }
            }
        }
    ]
}
```

3. Erstellen Sie in Konto A eine IAM-Rolle. Dieser Rollenname muss mit der Rolle übereinstimmen, die Sie in Schritt 2 in der Vertrauensrichtlinie angegeben haben. Um mehrere Konten anzuvisieren, erteilen Sie dem Orchestrator Berechtigungen, jede Rolle zu übernehmen. Das folgende Beispiel zeigt die Berechtigungen für Konto A zur Übernahme von Konto B. Wenn Sie über zusätzliche Zielkonten verfügen, fügen Sie dieser Richtlinie zusätzliche Rollen-ARNs hinzu. Sie können nur einen Rollen-ARN pro Zielkonto haben.

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

4. Diese IAM-Rolle für Konto A wird als `roleArn` für die Experimentvorlage verwendet. Das folgende Beispiel zeigt die Vertrauensrichtlinie, die in der IAM-Rolle erforderlich ist, die AWS FIS Berechtigungen zur Übernahme von Konto A, dem Orchestratorkonto, erteilt.

```
{
    "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. Um zu verwenden CloudFormation StackSets, müssen Sie die erforderlichen StackSet Berechtigungen in Ihren AWS Konten einrichten. Weitere Informationen finden Sie unter Arbeiten mit AWS. CloudFormation StackSets

Stoppbedingungen für Experimente mit mehreren Konten (optional)

Eine Stoppbedingung ist ein Mechanismus zum Stoppen eines Experiments, wenn sie einen Schwellenwert erreicht, den Sie als Alarm definieren. Um eine Stoppbedingung für Ihr Experiment mit mehreren Konten einzurichten, können Sie kontoübergreifende Alarme verwenden. Sie müssen die Freigabe in jedem Zielkonto aktivieren, um den Alarm mithilfe von schreibgeschützten Berechtigungen für das Orchestratorkonto verfügbar zu machen. Nach der Freigabe können Sie Metriken aus verschiedenen Zielkonten mithilfe von Metrikberechnungen kombinieren. Anschließend können Sie diesen Alarm als Stoppbedingung für das Experiment hinzufügen.

Weitere Informationen zu kontoübergreifenden Dashboards finden Sie unter Aktivieren der kontoübergreifenden Funktionalität in CloudWatch.

Arbeiten mit Experimenten mit mehreren Konten

Sie können Experimentvorlagen mit mehreren Konten mithilfe der AWS FIS Konsole oder der Befehlszeile erstellen und verwalten. Sie erstellen ein Experiment mit mehreren Konten`"multi-account"`, indem Sie die Option für das Experiment mit Kontoausrichtung als angeben und Zielkontokonfigurationen hinzufügen. Nachdem Sie eine Experimentvorlage mit mehreren Konten erstellt haben, können Sie damit ein Experiment ausführen.

Inhalt

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, mit konsistenten Ressourcen-Tags über alle Zielkonten hinweg vorzugehen. Ein AWS FIS Experiment löst Ressourcen mit konsistenten Tags in jedem Zielkonto auf. Eine Aktion muss mindestens eine Zielressource in einem Zielkonto auflösen oder schlägt fehl, mit Ausnahme von Experimenten, bei denen auf `emptyTargetResolutionMode` festgelegt ist`skip`. Pro Konto gelten Aktionskontingente. Wenn Sie Ressourcen nach Ressourcen-ARNs als Ziel auswählen möchten, gilt dasselbe Einzelkontolimit pro Aktion.
- Wenn Sie Ressourcen in einer oder mehreren Availability Zones mithilfe von Parametern oder Filtern anvisieren, sollten Sie eine AZ-ID angeben, keinen AZ-Namen. Die AZ-ID ist eine eindeutige und konsistente Kennung für eine Availability Zone über -Konten hinweg. 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.

Erstellen einer Experimentvorlage mit mehreren Konten

So erfahren Sie, wie Sie eine Experimentvorlage über die erstellen AWS Management Console

Siehe Erstellen Sie eine Versuchsvorlage.

So erstellen Sie eine Experimentvorlage mit der CLI

1. Öffnen der AWS Command Line Interface
2. Um ein Experiment aus einer gespeicherten JSON-Datei zu erstellen, bei der die Option Experiment für das Konto auf `"multi-account"` (z. B. `my-template.json`) festgelegt ist, ersetzen Sie die Platzhalterwerte in *`Kursivschrift`* 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 Experimentvorlage in der Antwort zurückgegeben. Kopieren Sie die `id` aus der Antwort, bei der es sich um die ID der Experimentvorlage handelt.
3. Führen Sie den create-target-account-configuration Befehl aus, um der Experimentvorlage eine Zielkontokonfiguration hinzuzufügen. Ersetzen Sie die Platzhalterwerte in *`Kursivschrift`* durch Ihre eigenen Werte, wobei Sie die `id` aus Schritt 2 als Wert für den `--experiment-template-id` Parameter verwenden, und führen Sie dann Folgendes 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 Befehl ausführen, um zu sehen, dass 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 eine Zählung aller Zielkontokonfigurationen in der Experimentvorlage ist.

6. Wenn Sie bereit sind, können Sie die Experimentvorlage mit dem Befehl start-experiment ausführen.

```
aws fis start-experiment --experiment-template-id EXTxxxxxxxxx
```

Aktualisieren einer Zielkontokonfiguration

Sie können eine vorhandene Zielkontokonfiguration aktualisieren, wenn Sie den Rollen-ARN oder die Beschreibung für das Konto ändern möchten. Wenn Sie eine Zielkontokonfiguration aktualisieren, wirken sich die Änderungen nicht auf laufende Experimente aus, die die Vorlage verwenden.

So aktualisieren Sie eine Zielkontokonfiguration mithilfe der AWS Management Console

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 und Aktionen, Experimentvorlage aktualisieren aus.
4. Ändern Sie die Konfigurationen des Zielkontos und wählen Sie Experimentvorlage aktualisieren aus.

So aktualisieren Sie eine Zielkontokonfiguration mithilfe der CLI

Führen Sie den update-target-account-configuration Befehl aus, um den Befehl auszuführen, und ersetzen Sie die *kursiv gedruckten* Platzhalterwerte 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 einer 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 weiter ausgeführt, bis es abgeschlossen oder gestoppt ist.

So löschen Sie eine Zielkontokonfiguration mithilfe der AWS Management Console

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 und Aktionen , Aktualisieren aus.
4. Wählen Sie unter Zielkontokonfigurationen die Option Entfernen für den Zielkonto-Rollen-ARN aus, den Sie löschen möchten.

So löschen Sie eine Zielkontokonfiguration mithilfe der CLI

Führen Sie den delete-target-account-configuration Befehl aus und ersetzen Sie die *kursiv gedruckten* Platzhalterwerte durch Ihre eigenen Werte.

```
aws fis update-target-account-configuration --experiment-template-id EXTxxxxxxxxx --
account-id 111122223333
```

AWS FIS Szenariobibliothek

Szenarien definieren Ereignisse oder Bedingungen, die Kunden anwenden können, um die Ausfallsicherheit ihrer Anwendungen zu testen, z. B. die Unterbrechung der Rechenressourcen, auf denen die Anwendung ausgeführt wird. Szenarien werden von AWS erstellt und gehören AWS und minimieren undifferenzierte Aufgaben, indem Sie Ihnen eine Gruppe vordefinierter Ziele und Fehleraktionen (z. B. das Stoppen von 30 % der Instances in einer Auto Scaling-Gruppe) für häufige Anwendungseinschränkungen zur Verfügung stellen.

Themen

Arbeiten mit AWS FIS Szenarien

Szenarien werden über eine reine Konsolenszenariobibliothek bereitgestellt und mithilfe einer AWS FIS Experimentvorlage ausgeführt. Um ein Experiment mit einem Szenario auszuführen, wählen Sie das Szenario aus der Bibliothek aus, geben Parameter an, die Ihren Workload-Details entsprechen, und speichern es als Experimentvorlage in Ihrem Konto.

Themen

Anzeigen eines Szenarios

So zeigen Sie ein Szenario mit der Konsole an:

1. Öffnen Sie die -AWS FISKonsole unter https://console.aws.amazon.com/fis/.
2. Wählen Sie im Navigationsbereich Szenariobibliothek aus.

3. Um Informationen zu einem bestimmten Szenario anzuzeigen, wählen Sie die Szenariokarte aus, um einen geteilten Bereich aufzurufen.
   - Auf der Registerkarte Beschreibung im geteilten Bereich unten auf der Seite können Sie eine kurze Beschreibung des Szenarios anzeigen. 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 können Sie auch zusätzliche Informationen zu den Zielen und Aktionen im Szenario sowie die erwartete Dauer sehen, in der das Experiment erfolgreich mit Standardeinstellungen 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 aus dem Szenario 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 annähert werden. Gegebenenfalls können Sie auch darüber lesen, welche Metriken als Stoppbedingungen verwendet werden sollen, und um Beobachtbarkeit zu bieten, um aus dem Experiment zu lernen. Schließlich finden Sie Empfehlungen zur Erweiterung der resultierenden Experimentvorlage.

Verwenden eines Szenarios

So verwenden Sie ein Szenario mit der Konsole:

1. Öffnen Sie die -AWS FISKonsole unter https://console.aws.amazon.com/fis/.
2. Wählen Sie im Navigationsbereich Szenariobibliothek aus.
3. Um Informationen zu einem bestimmten Szenario anzuzeigen, wählen Sie die Szenariokarte aus, um einen geteilten Bereich 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, die für mehrere Aktionen oder Ziele freigegeben sind, massenweise bearbeiten. Diese Funktionalität wird deaktiviert, sobald Sie Änderungen am Szenario vornehmen, einschließlich Änderungen durch die Massen-Parameterbearbeitung. Um diese Funktion zu verwenden, wählen Sie die Schaltfläche Massenparameter bearbeiten aus. Bearbeiten Sie Parameter im Modal und wählen Sie die Schaltfläche Speichern.

b. Einige Experimentvorlagen haben möglicherweise fehlende Aktions- oder Zielparameter, die auf jeder Aktion und Zielkarte hervorgehoben sind. Wählen Sie für jede Karte die Schaltfläche Bearbeiten, fügen Sie die fehlenden Informationen hinzu und wählen Sie auf der Karte die Schaltfläche Speichern.

c. Alle Vorlagen erfordern eine Ausführungsrolle für den Servicezugriff. Sie können eine vorhandene Rolle auswählen oder eine neue Rolle für diese Experimentvorlage erstellen.

d. Wir empfehlen, eine oder mehrere optionale Stopp-Bedingungen zu definieren, indem Sie einen vorhandenen AWS- CloudWatch Alarm auswählen. Weitere Informationen zu Stoppbedingungen für AWS FIS. Wenn Sie noch keinen Alarm konfiguriert haben, können Sie den Anweisungen unter Verwenden von Amazon CloudWatch Alarmen folgen und die Experimentvorlage später aktualisieren.

e. Wir empfehlen, optionale Experimentprotokolle in Amazon-Protokollen oder in einem Amazon S3-Bucket zu aktivieren. CloudWatch Weitere Informationen zu Experimentprotokollierung für AWS FIS. Wenn Sie noch keine geeigneten Ressourcen konfiguriert haben, können Sie die Experimentvorlage später aktualisieren.

6. Wählen Sie in der Vorlage Experiment erstellen die Option Experimentvorlage erstellen aus.
7. Wählen Sie in der Ansicht Experimentvorlagen der AWS FISKonsole Experiment starten aus. Weitere Informationen zu Experimente für AWS FIS.

Exportieren eines Szenarios

Szenarien sind eine reine Konsolenerfahrung. Obwohl es sich bei Szenarien um keine vollständigen Experimentvorlagen handelt, können sie nicht direkt in importiert werdenAWS FIS. Wenn Sie Szenarien als Teil Ihrer eigenen Automatisierung verwenden möchten, können Sie einen von zwei Pfaden verwenden:

1. Führen Sie die Schritte unter ausVerwenden eines Szenarios, um eine gültige AWS FIS Experimentvorlage zu erstellen und diese Vorlage zu exportieren.
2. Folgen Sie den Schritten in Anzeigen eines Szenarios und in Schritt 3 auf der Registerkarte Inhalt, kopieren und speichern Sie den Szenarioinhalt und fügen Sie dann fehlende Parameter manuell hinzu, um eine gültige Experimentvorlage zu erstellen.

Szenarien in der AWS FIS Szenarienbibliothek

Die in der Szenariobibliothek enthaltenen Szenarien sind so konzipiert, dass sie nach Möglichkeit Tags verwenden, und jedes Szenario beschreibt die erforderlichen Tags in den Abschnitten Voraussetzungen und Funktionsweise der Szenariobeschreibung. Sie können Ihre Ressourcen mit diesen vordefinierten Tags markieren oder Ihre eigenen Tags mithilfe der Massenbearbeitung von Parametern festlegen (siehe Verwenden eines Szenarios).

Diese Referenz beschreibt die gängigen Szenarien in der AWS FIS-Szenariobibliothek. Sie können die unterstützten Szenarien auch über die AWS FIS-Konsole auflisten.

Weitere Informationen finden Sie unter Arbeiten mit Szenarien.

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. Einige dieser Szenarien verwenden SSM-Dokumente .

- EC2-Ausfall: Instance-Ausfall – Erkunden Sie die Auswirkungen eines Instance-Ausfalls, indem Sie eine oder mehrere EC2-Instances anhalten.

  Ziel-Instances in der aktuellen Region, denen ein bestimmtes Tag angefügt ist. In diesem Szenario stoppen wir diese Instances und starten sie am Ende der Aktionsdauer neu, standardmäßig 5 Minuten.
- EC2-Auslastung: Festplatte – 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 angefügt ist. In diesem Szenario können Sie eine zunehmende Festplattenauslastung anpassen, die für die Aktionsdauer auf Ziel-EC2-Instances injiziert wird, standardmäßig 5 Minuten für jede Festplattenbelastungsaktion.
- EC2-Auslastung: CPU – Erkunden Sie die Auswirkungen einer erhöhten CPU auf Ihre EC2-basierte Anwendung.

  In diesem Szenario zielen wir auf EC2-Instances in der aktuellen Region ab, denen ein bestimmtes Tag angefügt ist. In diesem Szenario können Sie für die Aktionsdauer eine zunehmende CPU-Auslastung anpassen, die auf Ziel-EC2-Instances injiziert wird, standardmäßig 5 Minuten für jede CPU-Auslastungsaktion.
- EC2-Auslastung: Speicher – 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 angefügt ist. In diesem Szenario können Sie eine zunehmende Speicherauslastung, die auf Ziel-EC2-Instances injiziert wird, für die Aktionsdauer anpassen, standardmäßig 5 Minuten für jede Speicherauslastungsaktion.

- EC2-Auslastung: 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 angefügt ist. In diesem Szenario können Sie eine zunehmende Netzwerklatenz anpassen, die auf Ziel-EC2-Instances für die Aktionsdauer injiziert wird, standardmäßig 5 Minuten für jede Latenzaktion.

AWS FIS unterstützt die folgenden Amazon EKS-Szenarien. Diese Szenarien zielen auf EKS-Pods mithilfe von Kubernetes-Anwendungsbezeichnungen ab. Sie können Ihre eigenen Labels oder die im Szenario enthaltenen Standardlabels verwenden. Weitere Informationen zu EKS mit FIS finden Sie unter Verwenden Sie die EKS-Pod-Aktionen.

- EKS-Konstrukte: Pod-Löschung – 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 einer Anwendungsbezeichnung zugeordnet sind. In diesem Szenario werden wir alle übereinstimmenden Pods beenden. Die Neuerstellung von Pods wird durch die Kubernetes-Konfiguration gesteuert.
- EKS-Konstrukte: CPU – Erkunden Sie die Auswirkungen einer erhöhten CPU auf Ihre EKS-basierte Anwendung.

  In diesem Szenario zielen wir auf Pods in der aktuellen Region ab, die einer Anwendungsbezeichnung zugeordnet sind. In diesem Szenario können Sie für die Aktionsdauer eine zunehmende CPU-Auslastung anpassen, die auf Ziel-EKS-Pods injiziert wird, standardmäßig 5 Minuten für jede CPU-Auslastungsaktion.
- EKS-Effort: Festplatte – 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 einer Anwendungsbezeichnung zugeordnet sind. In diesem Szenario können Sie für die Aktionsdauer eine zunehmende Festplattenbelastung anpassen, die auf Ziel-EKS-Pods injiziert wird, standardmäßig 5 Minuten für jede CPU-Streitaktion.

- EKS-Konstrukt: Speicher – 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 einer Anwendungsbezeichnung zugeordnet sind. In diesem Szenario können Sie für die Aktionsdauer eine zunehmende Speicherauslastung anpassen, die auf Ziel-EKS-Pods injiziert wird, standardmäßig 5 Minuten für jede Speicherauslastungsaktion.
- EKS-Konflikt: 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 einer Anwendungsbezeichnung zugeordnet sind. In diesem Szenario können Sie eine zunehmende Netzwerklatenz anpassen, die für die Aktionsdauer auf Ziel-EKS-Pods injiziert wird, standardmäßig 5 Minuten für jede Latenzaktion.

AWS FIS unterstützt die folgenden Szenarien für Multi-AZ- und Multi-Regions-Anwendungen. Diese Szenarien zielen auf mehrere Ressourcentypen ab.

- AZ Availability: Power Interruption – Injizieren Sie die erwarteten Reaktionen einer vollständigen Unterbrechung der Stromversorgung in einer Availability Zone (AZ). Weitere Informationen zu AZ Availability: Power Interruption.
- Cross-Region: Connectivity – Blockieren Sie den Netzwerkverkehr der Anwendung von der Experimentregion zur Zielregion und unterbrechen Sie die regionsübergreifende Datenreplikation. Weitere Informationen zur Verwendung von Cross-Region: Connectivity.

AZ Availability: Power Interruption

Sie können das -AZ Availability: Power InterruptionSzenario verwenden, um die erwarteten Wahrscheinlichkeiten einer vollständigen Unterbrechung der Stromversorgung in einer Availability Zone (AZ) auszulösen.

Dieses Szenario kann verwendet werden, um zu zeigen, dass Multi-AZ-Anwendungen während einer einzigen, vollständigen AZ-Leistungsunterbrechung wie erwartet funktionieren. Sie umfasst den Verlust von zonaler Datenverarbeitung (Amazon EC2, EKS und ECS), keine Neuskalierung der Datenverarbeitung in der AZ, Verlust der Subnetzkonnektivität, RDS-Failover, ElastiCache Failover und nicht reagierende EBS-Volumes. Standardmäßig werden Aktionen, für die keine Ziele gefunden werden, übersprungen.

Aktionen

Zusammen erzeugen die folgenden Aktionen viele der erwarteten Faktoren einer vollständigen Stromunterbrechung in einer einzigen AZ. AZ-Verfügbarkeit: Stromunterbrechung wirkt sich nur auf Services aus, bei denen zu erwarten ist, dass sie während einer einzelnen AZ-Leistungsunterbrechung Auswirkungen haben. Standardmäßig injiziert das Szenario 30 Minuten lang Stromunterbrechungsfaktoren und dann weitere 30 Minuten lang die während der Wiederherstellung auftretenden Reaktionen.

Stopp-Instances

Während einer AZ-Leistungsunterbrechung werden EC2-Instances in der betroffenen AZ heruntergefahren. Nachdem die Stromversorgung wiederhergestellt wurde, werden Instances neu gestartet. AZ Availability: Power Interruption enthält aws:ec2:stop-instances, um alle Instances in der betroffenen AZ für die Unterbrechungsdauer anzuhalten. Nach Ablauf der Dauer werden die Instances 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 mit einem Tag namens `AzImpairmentPower` mit dem Wert ab`StopInstances`. Sie können dieses Tag zu Ihren Instances hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Instances gefunden werden, wird diese Aktion standardmäßig übersprungen.

Stop-ASG-Instances

Während einer AZ-Leistungsunterbrechung werden EC2-Instances, die von einer Auto Scaling-Gruppe in der betroffenen AZ verwaltet werden, heruntergefahren. Nachdem die Stromversorgung wiederhergestellt wurde, werden Instances neu gestartet. AZ Availability: Power Interruption enthält aws:ec2:stop-instances, um alle Instances, einschließlich der von Auto Scaling verwalteten Instances, in der betroffenen AZ für die Unterbrechungsdauer 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 es auf Instances mit einem Tag namens `AzImpairmentPower` mit dem Wert ab`IceAsg`. Sie können dieses Tag zu Ihren Instances hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Instances gefunden werden, wird diese Aktion standardmäßig übersprungen.

Instance-Starts anhalten

Während einer AZ-Leistungsunterbrechung schlagen EC2-API-Aufrufe zur Bereitstellung von Kapazität in der AZ fehl. Insbesondere sind die folgenden APIs betroffen: `ec2:StartInstances`, und `ec2:RunInstances`. AZ Availability: Power Interruption includes enthält aws:ec2:api-insufficient-instance-capacity-error, um zu verhindern`ec2:CreateFleet`, dass neue Instances in der betroffenen AZ bereitgestellt werden.

Diese Aktion zielt auf IAM-Rollen ab, die zur Bereitstellung von Instances verwendet werden. Diese müssen mit einem ARN als Ziel ausgewählt werden. Wenn keine gültigen IAM-Rollen gefunden werden, wird diese Aktion standardmäßig übersprungen.

ASG-Skalierung anhalten

Während einer AZ-Leistungsunterbrechung schlagen EC2-API-Aufrufe fehl, die von der Auto Scaling-Steuerebene ausgeführt werden, um verlorene Kapazität in der AZ wiederherzustellen. Insbesondere sind die folgenden APIs betroffen: `ec2:StartInstances`, und `ec2:RunInstances`. AZ Availability: Power Interruption enthält aws:ec2:asg-insufficient-instance-capacity-error, um zu verhindern`ec2:CreateFleet`, dass neue Instances in der betroffenen AZ bereitgestellt werden. Dies verhindert auch, dass Amazon EKS und Amazon ECS in der betroffenen AZ skalieren.

Diese Aktion zielt auf Auto Scaling-Gruppen ab. Standardmäßig zielt es auf Auto Scaling-Gruppen mit einem Tag namens `AzImpairmentPower` mit einem Wert von ab`IceAsg`. Sie können dieses Tag zu Ihren Auto Scaling-Gruppen hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Auto Scaling-Gruppen gefunden werden, wird diese Aktion standardmäßig übersprungen.

Anhalten der Netzwerkverbindung

Während einer AZ-Leistungsunterbrechung ist das Netzwerk in der AZ nicht verfügbar. In diesem Fall kann es einige AWS-Services bis zu einigen Minuten dauern, bis DNS aktualisiert wird, um zu berücksichtigen, dass private Endpunkte in der betroffenen AZ nicht verfügbar sind. Während dieser Zeit können DNS-Lookups unzugängliche IP-Adressen zurückgeben. AZ Availability: Power Interruption enthält aws:network:disrupt-connectivity, um die gesamte Netzwerkkonnektivität für alle Subnetze in der betroffenen AZ für 2 Minuten zu blockieren. Dies erzwingt für die meisten Anwendungen Timeouts und DNS-Aktualisierungen. Das Beenden der Aktion nach 2 Minuten ermöglicht eine nachfolgende Wiederherstellung des regionalen Service-DNS, während die AZ weiterhin nicht verfügbar ist.

Diese Aktion zielt auf Subnetze ab. Standardmäßig zielt sie auf Cluster mit einem Tag namens `AzImpairmentPower` mit dem Wert ab`DisruptSubnet`. Sie können dieses Tag zu Ihren Subnetzen hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Subnetze gefunden werden, wird diese Aktion standardmäßig übersprungen.

Failover-RDS

Während einer AZ-Leistungsunterbrechung werden 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 die Reader-Knoten in der betroffenen AZ sind nicht verfügbar. Bei Multi-AZ-Clustern AZ Availability: Power Interruption umfasst aws:rds:failover-db-cluster zum Failover, wenn sich der Writer in der betroffenen AZ befindet.

Diese Aktion zielt auf RDS-Cluster ab. Standardmäßig zielt es auf Cluster mit einem Tag namens `AzImpairmentPower` mit dem Wert ab`DisruptRds`. Sie können dieses Tag zu Ihren Clustern hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Cluster gefunden werden, wird diese Aktion standardmäßig übersprungen.

ElastiCache Redis anhalten

Während einer AZ-Leistungsunterbrechung sind ElastiCache Knoten in der AZ nicht verfügbar. AZ Availability: Power Interruption enthält aws:elasticache:interrupt-cluster-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 der Cluster weiterhin eine geringere Kapazität hat.

Diese Aktion zielt auf ElastiCache Cluster ab. Standardmäßig zielt es auf Cluster mit einem Tag namens `AzImpairmentPower` mit einem Wert von ab`ElasticacheImpact`. Sie können dieses Tag zu Ihren Clustern hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Cluster gefunden werden, wird diese Aktion standardmäßig übersprungen. Beachten Sie, dass nur Cluster mit Writer-Knoten in der betroffenen AZ als gültige Ziele betrachtet werden.

EBS-E/A anhalten

Nach einer AZ-Leistungsunterbrechung kann es bei einem sehr kleinen Prozentsatz der Instances zu nicht reagierenden EBS-Volumes kommen. AZ Availability: Power Interruption enthält aws:ebs:pause-io, um ein EBS-Volume in einem nicht reagierenden Zustand zu belassen.

Standardmäßig werden nur Volumes anvisiert, die nach dem Beenden der Instance bestehen bleiben sollen. Diese Aktion zielt auf Volumes mit einem Tag namens `AzImpairmentPower` mit dem Wert ab`APIPauseVolume`. Sie können dieses Tag zu Ihren Volumes hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Volumes gefunden werden, wird diese Aktion standardmäßig übersprungen.

Einschränkungen

- Dieses Szenario enthält keine Stoppbedingungen. Die richtigen Stoppbedingungen für Ihre Anwendung sollten der Experimentvorlage hinzugefügt werden.
- Amazon EKS Pods, die auf AWS Fargate ausgeführt werden, werden nicht unterstützt.
- Amazon ECS-Aufgaben, die auf AWS Fargate ausgeführt werden, werden nicht unterstützt.
- Amazon RDS Multi-AZ mit zwei lesbaren Standby-DB-Instances wird nicht unterstützt. In diesem Fall werden die Instances beendet, RDS wird ein Failover durchführen und die Kapazität wird sofort wieder in der betroffenen AZ bereitgestellt. Die lesbare Standby-Instance 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

Die folgende Richtlinie gewährt AWS FIS die erforderlichen Berechtigungen, um ein Experiment mit dem AZ Availability: Power Interruption Szenario durchzuführen. Diese Richtlinie muss an die Experimentrolle angehängt werden.

```
{
    "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": [
                "*"
            ]
```

```
        }
    ]
}
```

Szenarioinhalt

Der folgende Inhalt definiert das Szenario. Dieses JSON kann gespeichert und verwendet werden, um eine Experimentvorlage mit dem create-experiment-template Befehl aus der AWS Command Line Interface (AWS CLI) 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:redis-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:interrupt-cluster-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."
}
```

Cross-Region: Connectivity

Sie können das Cross-Region: Connectivity Szenario verwenden, um den Datenverkehr des Anwendungsnetzwerks von der Experimentregion zur Zielregion zu blockieren und die regionsübergreifende Replikation für Amazon S3 und Amazon DynamoDB anzuhalten. Regionsübergreifend: Die Konnektivität wirkt sich auf den ausgehenden Anwendungsverkehr aus der Region aus, in der Sie das Experiment ausführen (Experimentregion). Zustandslosen eingehenden Datenverkehr aus der Region, die Sie von der Experimentregion (Zielregion) isolieren möchten, wird möglicherweise nicht blockiert. Der Datenverkehr von AWS-verwalteten Services wird möglicherweise nicht blockiert.

Dieses Szenario kann verwendet werden, um zu zeigen, dass Anwendungen mit mehreren Regionen wie erwartet funktionieren, wenn Ressourcen in der Zielregion von der Experimentregion aus nicht zugänglich sind. Dazu gehört das Blockieren des Netzwerkverkehrs von der Experimentregion zur Zielregion durch Ausrichtung auf Transit-Gateways und Routing-Tabellen. Außerdem wird die regionsübergreifende Replikation für S3 und DynamoDB angehalten. Standardmäßig werden Aktionen, für die keine Ziele gefunden werden, 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 Datenverkehr für 3 Stunden, was Sie auf eine maximale Dauer von 12 Stunden erhöhen können.

Unterbrechen der Transit-Gateway-Konnektivität

Cross Region: Connectivity umfasst aws:network:transit-gateway-disrupt-cross-region-connectivity, um regionsübergreifenden Netzwerkverkehr von VPCs in der Experimentregion zu VPCs in der

Zielregion zu blockieren, die über ein Transit Gateway verbunden sind. Dies wirkt sich nicht auf den Zugriff auf VPC-Endpunkte innerhalb der Experimentregion aus, blockiert aber den Datenverkehr aus der Experimentregion, die für einen VPC-Endpunkt in der Zielregion bestimmt ist.

Diese Aktion zielt auf Transit-Gateways ab, die die Experimentregion und die Zielregion verbinden. Standardmäßig zielt sie auf Transit-Gateways mit einem Tag namens `DisruptTransitGateway` mit einem Wert von ab`Allowed`. Sie können dieses Tag zu Ihren Transit-Gateways hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Transit-Gateways gefunden werden, wird diese Aktion standardmäßig übersprungen.

Unterbrechen der Subnetzkonnektivität

Cross Region: Connectivity umfasst aws:network:route-table-disrupt-cross-region-connectivity, um regionsübergreifenden Netzwerkverkehr von VPCs in der Experimentregion zu öffentlichen AWS-IP-Blöcken in der Zielregion zu blockieren. Zu diesen öffentlichen IP-Blöcken gehören AWS-Service-Endpunkte 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 Experimentregion zur Zielregion . Sie wirkt sich nicht auf den Zugriff auf VPC-Endpunkte in der Experimentregion aus, blockiert aber den Datenverkehr aus der Experimentregion, die für einen VPC-Endpunkt in der Zielregion bestimmt ist.

Diese Aktion zielt auf Subnetze in der Experimentregion ab. Standardmäßig zielt es auf Subnetze mit einem Tag namens `DisruptSubnet` mit dem Wert ab`Allowed`. Sie können dieses Tag zu Ihren Subnetzen hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Subnetze gefunden werden, wird diese Aktion standardmäßig übersprungen.

S3-Replikation anhalten

Cross Region: Connectivity enthält aws:s3:bucket-pause-replication um die S3-Replikation von der Experimentregion zur Zielregion für die Ziel-Buckets anzuhalten. Die Replikation von der Zielregion zur Experimentregion ist davon nicht betroffen. Nach dem Ende des Szenarios wird die Bucket-Replikation ab dem Zeitpunkt 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 des Objekt-Uploads in den Bucket variiert.

Diese Aktion zielt auf S3-Buckets in der Experimentregion mit regionsübergreifender Replikation (CRR) ab, die für einen S3-Bucket in der Zielregion aktiviert sind. Standardmäßig zielt es auf Buckets

mit einem Tag namens `DisruptS3` mit einem Wert von ab`Allowed`. Sie können dieses Tag zu Ihren Buckets hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen Buckets gefunden werden, wird diese Aktion standardmäßig übersprungen.

Anhalten der DynamoDB-Replikation

Cross-Region: Connectivity enthält aws:dynamodb:encrypted-global-table-pause-replication, um die Replikation zwischen der Experimentregion und allen anderen Regionen, einschließlich der Zielregion, anzuhalten. Dies verhindert die Replikation in und aus der Experimentregion, hat jedoch keinen Einfluss auf die Replikation zwischen anderen Regionen. Nach dem Ende des Szenarios wird die Tabellenreplikation ab dem Zeitpunkt fortgesetzt, an dem sie angehalten wurde. Beachten Sie, dass die Zeit, die für die Replikation benötigt wird, um alle Daten synchron zu halten, je nach Dauer des Experiments und der Änderungsrate der Tabelle variiert.

Diese Aktion zielt auf verschlüsselte globale DynamoDB-Tabellen in der Experimentregion ab, die mit vom Kunden verwalteten Schlüsseln verschlüsselt sind. Standardmäßig zielt sie auf Tabellen mit einem Tag namens `DisruptDynamoDb` mit einem Wert von ab`Allowed`. Sie können dieses Tag zu Ihren Tabellen hinzufügen oder das Standard-Tag durch Ihr eigenes Tag in der Experimentvorlage ersetzen. Wenn keine gültigen globalen Tabellen gefunden werden, wird diese Aktion standardmäßig übersprungen.

Einschränkungen

- Dieses Szenario enthält keine Stoppbedingungen. Die richtigen Stoppbedingungen für Ihre Anwendung sollten der Experimentvorlage 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 zum Ausführen eines Experiments mit dem Cross-Region: Connectivity Szenario. Diese Richtlinie muss an die Experimentrolle angehängt 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",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity12",
            "Effect": "Allow",
            "Action": "ec2:ModifyManagedPrefixList",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity13",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeManagedPrefixLists",
                "ec2:DescribeSubnets",
                "ec2:DescribeRouteTables",
                "ec2:DescribeVpcEndpoints"
            ],
            "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": "RouteTableDisruptConnectivity20",
            "Effect": "Allow",
            "Action": "ec2:ModifyVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:*:*:vpc-endpoint/*"
            ]
        },
        {
            "Sid": "TransitGatewayDisruptConnectivity1",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateTransitGatewayRouteTable",
                "ec2:AssociateTransitGatewayRouteTable"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:transit-gateway-route-table/*",
                "arn:aws:ec2:*:*:transit-gateway-attachment/*"
            ]
        },
        {
            "Sid": "TransitGatewayDisruptConnectivity2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeTransitGatewayPeeringAttachments",
                "ec2:DescribeTransitGatewayAttachments",
                "ec2:DescribeTransitGateways"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion1",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion2",
```

```
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "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": "DdbCrossRegion1",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DdbCrossRegion2",
            "Effect": "Allow",
```

```
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:DescribeGlobalTable"
            ],
            "Resource": [
                "arn:aws:dynamodb:*:*:table/*",
                "arn:aws:dynamodb:*:*:global-table/*"
            ]
        },
        {
            "Sid": "DdbCrossRegion3",
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey",
                "kms:GetKeyPolicy",
                "kms:PutKeyPolicy"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        }
    ]
}
```

Szenarioinhalt

Der folgende Inhalt definiert das Szenario. Dieses JSON kann gespeichert und verwendet werden, um eine Experimentvorlage mit dem create-experiment-template Befehl aus der AWS Command Line Interface (AWS CLI) 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": {}
                },
                "S3-Bucket": {
                        "resourceType": "aws:s3:bucket",
                        "resourceTags": {
                                "S3Impact": "Allowed"
                        },
                        "selectionMode": "ALL"
                },
                "DynamoDB-Global-Table": {
                        "resourceType": "aws:dynamodb:encrypted-global-table",
                        "resourceTags": {
                                "DisruptDynamoDb": "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"
                        }
                },
                "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:encrypted-global-table-pause-
replication",
                        "parameters": {
                                "duration": "PT3H"
                        },
                        "targets": {
                                "Tables": "DynamoDB-Global-Table"
                        }
                }
        },
        "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"
}
```

Experimente für AWS FIS

AWS Mit FIS können Sie Fault-Injection-Experimente 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: Planen eines wiederkehrenden Experiments.

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 AWS sich FIS-Nutzungsmetriken an. Weitere Informationen finden Sie unter Überwachen 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.

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 vor Beginn der Aktion beendet.
- 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 list-experiment-resolved-targets-Befehl.

Experiment-Scheduler

Mit AWS Fault Injection Service (FIS) können Sie Fehlersimulationsexperimente für Ihre AWS-Workloads durchführen. Diese Experimente werden auf Vorlagen ausgeführt, die eine oder mehrere Aktionen enthalten, die auf bestimmten Zielen ausgeführt werden sollen. Sie können Ihre Experimente jetzt als einmalige oder wiederkehrende Aufgaben nativ über die FIS-Konsole 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 Serverless-Scheduler, mit dem Sie Aufgaben von einem zentralen, verwalteten Service aus erstellen, ausführen und verwalten können.

> **Important**
>
> Experiment Scheduler mit AWS Fault Injection Service ist in AWS GovCloud (USA-Ost) und AWS GovCloud (USA-West) nicht verfügbar.

Themen

Erste Schritte

Eine Ausführungsrolle ist eine IAM-Rolle, die AWS Fault Injection Service annimmt, um mit dem EventBridge Scheduler zu interagieren und damit der Event Bridge-Scheduler das FIS-Experiment starten kann. Sie fügen dieser Rolle Berechtigungsrichtlinien an, um Scheduler Zugriff zum Aufrufen von FIS Experiment zu gewähren EventBridge. In den folgenden Schritten wird beschrieben, wie Sie eine neue Ausführungsrolle und eine Richtlinie erstellen, damit ein Experiment EventBridge starten kann.

Erstellen einer Scheduler-Rolle mit der AWS CLI

Dies ist eine IAM-Rolle, die benötigt wird, damit Event Bridge Experimente im Namen des Kunden planen kann.

1. Kopieren Sie die folgende JSON-Richtlinie für die Übernahmerolle und speichern Sie sie lokal als `fis-execution-role.json`. Diese Vertrauensrichtlinie ermöglicht es EventBridge Scheduler, die Rolle in Ihrem Namen zu übernehmen.

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

2. Geben Sie in der AWS Command Line Interface (AWS CLI) den folgenden Befehl ein, um eine neue Rolle zu erstellen. Ersetzen Sie durch `FisSchedulerExecutionRole` den Namen, den Sie dieser Rolle zuweisen 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": "AROAZL22PDN5A6WKRBQNU",
        "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"
                }
            ]
        }
    }
}
```

3. 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 als `fis-start-experiment-permissions.json`. Mit der folgenden Richtlinie kann EventBridge Scheduler die `fis:StartExperiment` Aktion für alle Experimentvorlagen in Ihrem Konto aufrufen. Ersetzen Sie `*` am Ende von durch `"arn:aws:fis:*:*:experiment-template/*"` die ID Ihrer Experimentvorlage, wenn Sie die Rolle auf eine einzelne Experimentvorlage beschränken möchten.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": [
                "arn:aws:fis:*:*:experiment-template/*",
                "arn:aws:fis:*:*:experiment/*"
            ]
        }
    ]
}
```

4. Führen Sie den folgenden Befehl aus, um die neue Berechtigungsrichtlinie zu erstellen. Ersetzen Sie durch `FisSchedulerPolicy` 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. Notieren Sie sich 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. Ersetzen Sie durch `your-policy-arn` den ARN der Richtlinie, die Sie im vorherigen Schritt erstellt haben. Ersetzen Sie durch `FisSchedulerExecutionRole` 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 auf der Befehlszeile zurück.

6. Sie können den Scheduler so einschränken, dass er nur AWS-FIS-Experimente ausführt, die einen bestimmten Tag-Wert haben. Die folgende Richtlinie gewährt beispielsweise die `fis:StartExperiment` Berechtigung für alle AWS-FIS-Experimentvorlagen, schränkt den Scheduler jedoch so ein, dass nur Experimente ausgeführt werden, die mit gekennzeichnet sind `Purpose=Schedule`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

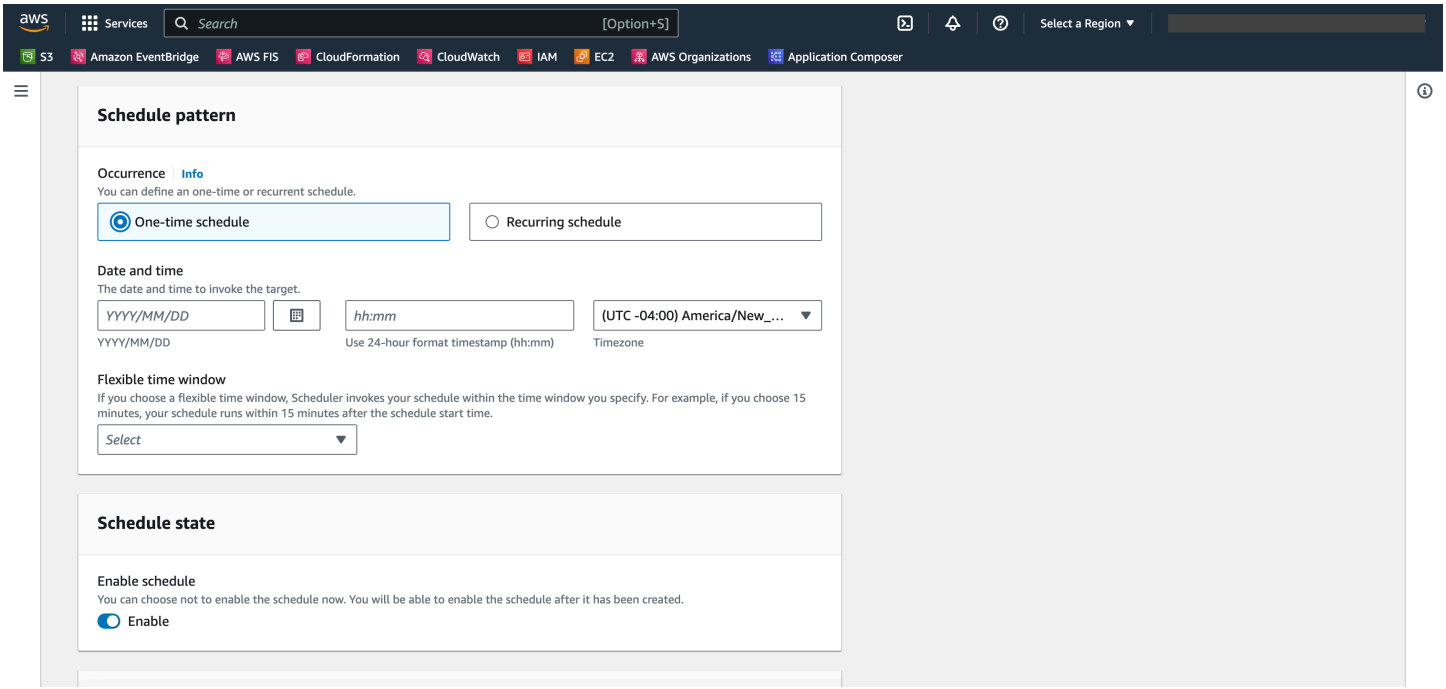
```
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": "arn:aws:fis:*:*:experiment/*"
        },
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": "arn:aws:fis:*:*:experiment-template/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Schedule"
                }
            }
        }
    ]
}
```

Planen eines -FIS-Experiments

Bevor Sie ein Experiment planen, benötigen Sie einen oder mehrere Experimentvorlagen für den Aufruf Ihres Zeitplans. Sie können eine vorhandene AWS-Ressource verwenden oder eine neue erstellen.

Sobald die Experimentvorlage 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 Zeitplanmuster und wählen Sie entweder Einmaliger Zeitplan oder Wiederkehrend aus. Füllen Sie die erforderlichen Eingabefelder aus und navigieren Sie zu Berechtigungen.



Der Zeitplanstatus ist standardmäßig aktiviert. Hinweis: Wenn Sie den Zeitplanstatus deaktivieren, wird das Experiment nicht geplant, selbst wenn Sie einen Zeitplan erstellen.

AWS FIS Der Experiment Scheduler baut auf dem EventBridge Scheduler auf. Sie können die Dokumentation für die verschiedenen unterstützten Zeitplantypen lesen.

So aktualisieren Sie den Zeitplan mithilfe der Konsole

1. Öffnen Sie die AWS FIS-Konsole.
2. Wählen Sie im linken Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Experimentvorlage aus, für die Sie den Zeitplan erstellen möchten.
4. Klicken Sie auf Aktionen und wählen Sie in der Dropdownliste die Option Experiment planen aus.
   a. Unter Zeitplanname wird der Name automatisch ausgefüllt.
   b. Wählen Sie unter Zeitplanmuster die Option Wiederkehrender Zeitplan aus.
   c. Unter Zeitplantyp können Sie einen ratenbasierten 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. Aktivieren Sie unter Zeitplanstatus die Option Zeitplan aktivieren .

i. Wählen Sie unter Berechtigungen die Option Vorhandene Rolle verwenden aus und suchen Sie dann nach `FisSchedulerExecutionRole`.

j. Wählen Sie Weiter aus.

5. Wählen Sie Zeitplan überprüfen und erstellen, überprüfen Sie Ihre Scheduler-Details und wählen Sie dann Zeitplan erstellen aus.

Aktualisieren des Experimentplans

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 eine Experimentausführung mithilfe der Konsole

1. Öffnen Sie die Amazon-FIS-Konsole .
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Ressourcentyp: Experimentvorlage aus, für die bereits ein Zeitplan erstellt wurde.
4. Klicken Sie auf die Experiment-ID für die Vorlage. Navigieren Sie dann zur Registerkarte Zeitpläne.
5. Überprüfen Sie, ob dem Experiment ein Zeitplan zugeordnet ist. Wählen Sie den zugehörigen Zeitplan aus und klicken Sie auf die Schaltfläche Zeitplan aktualisieren .

Deaktivieren oder Löschen einer Experimentausführung mithilfe der Konsole

Um zu verhindern, dass ein Experiment nach einem Zeitplan ausgeführt oder ausgeführt wird, können Sie die Regel löschen oder deaktivieren. Die folgenden Schritte führen Sie durch das Löschen oder Deaktivieren einer Experimentausführung.

So löschen oder deaktivieren Sie eine Regel

1. Öffnen Sie die Amazon-FIS-Konsole .
2. Wählen Sie im Navigationsbereich Experimentvorlagen aus.
3. Wählen Sie Ressourcentyp: Experimentvorlage aus, für die bereits ein Zeitplan erstellt wurde.
4. Klicken Sie auf die Experiment-ID für die Vorlage. Navigieren Sie dann zur Registerkarte Zeitpläne.

5. Überprüfen Sie, ob dem Experiment ein 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, wählen Sie die Schaltfläche neben der Regel Zeitplan löschen aus. Geben Sie ein `delete` und klicken Sie auf die Schaltfläche Zeitplan löschen.
   b. Um den Zeitplan zu deaktivieren, wählen Sie die Schaltfläche neben der Regel Zeitplan deaktivieren aus. Geben Sie ein `disable` und klicken Sie auf die Schaltfläche Zeitplan deaktivieren.

Überwachung von AWS FIS

Sie können die folgenden Tools verwenden, um den Fortschritt und die Auswirkungen Ihrer AWS Fault Injection Service (AWS FIS)-Experimente 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 anzeigen. Weitere Informationen finden Sie unter the section called “Sehen Sie sich Ihre Experimente an”.

CloudWatch -Nutzungsmetriken und Alarme

Verwenden Sie CloudWatch Nutzungsmetriken, um einen Einblick in die Ressourcennutzung Ihres Kontos zu erhalten. AWS FIS-Nutzungsmetriken entsprechen AWS Service Quotas. Sie können Alarme konfigurieren, mit denen Sie benachrichtigt werden, wenn sich Ihre Nutzung einem Servicekontingent nähert. Weitere Informationen finden Sie unter Überwachen mit CloudWatch.

Sie können auch Stoppbedingungen für Ihre AWS FIS-Experimente erstellen, indem Sie CloudWatch Alarme erstellen, die definieren, wann ein Experiment außerhalb der Grenzen liegt. Wenn der Alarm ausgelöst wird, stoppt das Experiment. Weitere Informationen finden Sie unter Stoppbedingungen. Weitere Informationen zum Erstellen von CloudWatch Alarmen finden Sie unter Erstellen eines CloudWatch Alarms basierend auf einem statischen Schwellenwert und Erstellen eines CloudWatch Alarms basierend auf Anomalieerkennung im Amazon- CloudWatch Benutzerhandbuch.

AWS Protokollierung von FIS-Experimenten

Aktivieren Sie die Experimentprotokollierung, um während der Ausführung detaillierte Informationen zu Ihrem Experiment zu erfassen. Weitere Informationen finden Sie unter Protokollierung von Experimenten.

Ereignisse zur Änderung des Experimentstatus

Mit Amazon EventBridge können Sie automatisch auf Systemereignisse oder Ressourcenänderungen reagieren. AWS FIS gibt eine Benachrichtigung aus, wenn sich der Status eines Experiments ändert. Sie können Regeln für die Ereignisse erstellen, an denen Sie interessiert sind und die die automatisierte Aktion angeben, die ausgeführt werden soll, wenn ein Ereignis mit einer Regel übereinstimmt. Zum Beispiel das Senden einer Benachrichtigung an ein

Amazon SNS-Thema oder das Aufrufen einer Lambda-Funktion. Weitere Informationen finden Sie unter Überwachen mit EventBridge.

CloudTrail -Protokolle

Verwenden Sie AWS CloudTrail, um detaillierte Informationen zu den Aufrufen der AWS FIS-API zu erfassen und sie als Protokolldateien in Amazon S3 zu speichern. protokolliert CloudTrail auch Aufrufe an Service-APIs für die Ressourcen, auf denen Sie Experimente ausführen. Sie können diese CloudTrail Protokolle verwenden, um zu bestimmen, welche Aufrufe getätigt wurden, von welcher Quell-IP-Adresse der Aufruf stammt, wer den Aufruf getätigt hat, wann der Aufruf getätigt wurde usw.

AWS Benachrichtigungen zum Zustands-Dashboard

AWS Health bietet einen fortlaufenden Einblick in Ihre Ressourcenleistung 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 in einem Experiment abzielt, einschließlich Experimenten mit mehreren Konten. Experimente mit mehreren Konten und nur Aktionen, die keine Ziele enthalten, wie und `aws:fis:wait`, geben keine `aws:ssm:start-automation-execution` Benachrichtigung aus. Informationen über die Rolle, die zum Zulassen des Experiments verwendet wird, werden unter Betroffene Ressourcen aufgeführt. Weitere Informationen zum AWS Health Dashboard finden Sie unter AWS Health Dashboard im AWS Health-Benutzerhandbuch.

> **Note**
>
> AWS Health liefert Ereignisse nach bestem Bemühen.

Überwachen SieAWS die FIS-Nutzungskennzahlen mit Amazon CloudWatch

Sie können Amazon verwenden CloudWatch , um die Auswirkungen vonAWS FIS-Experimenten auf Ziele zu überwachen. Sie können auch IhreAWS FIS-Nutzung überwachen.

Weitere Informationen über das Anzeigen des Zustands eines Experiments finden Sie unterSehen Sie sich Ihre Experimente an.

AWSFIS-Experimente überwachen

Identifizieren Sie bei der Planung IhrerAWS FIS-Experimente die CloudWatch Metriken, anhand derer Sie den Ausgangswert oder den „Steady-State“ für die Zielressourcentypen für das Experiment identifizieren 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 vonAWS FIS unterstützten Zielressourcentyp finden Sie im Folgenden:

- Überwachen Sie Ihre Instanzen mit CloudWatch
- Amazon CloudWatch ECS-Metriken
- Überwachung von Amazon RDS-Metriken mit CloudWatch
- Überwachung der Run Command-Metriken mithilfe von CloudWatch

AWSFIS-Nutzungsmetriken

Sie können CloudWatch -Nutzungsmetriken verwenden, um einen Einblick in die Ressourcennutzung Ihres Kontos zu gewähren. Verwenden Sie diese Metriken, um Ihre aktuelle Servicenutzung für CloudWatch -Diagramme und -Dashboards zu visualisieren.

AWSDie FIS-Nutzungsmetriken entsprechenAWS -Servicekontingenten. Sie können Alarme konfigurieren, mit denen Sie benachrichtigt werden, wenn sich Ihre Nutzung einem Servicekontingent nähert. Weitere Informationen zu CloudWatch Alarmen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

AWSFIS veröffentlicht die folgende Metrik im AWS/Usage-Namespace.

Metrik	Beschreibung
`ResourceCount`	Die Gesamtanzahl der angegebenen Ressourcen, die in Ihrem Konto ausgeführ t werden. Die Ressource wird durch die Dimensionen definiert, die der Metrik zugeordnet sind.

Die folgenden Dimensionen werden verwendet, um die Nutzungsmetriken zu verfeinern, die vonAWS FIS veröffentlicht werden.

Dimension	Beschreibung
`Service`	Der Name des AWS-Service, der die Ressource enthält. FürAWS FIS-Nutzungsmetriken lautet der Wert für diese Dimension`FIS`.
`Type`	Der Typ von Entität, die gemeldet wird. Derzeit ist der einzige gültige Wert fürAWS FIS-Nutzungsmetriken`Resource`.
`Resource`	Der Typ der Ressource, die ausgeführt wird. Die möglichen Werte gelten`ExperimentTemplates` für Versuchsvorlagen und`ActiveExperiments` für aktive Experimente.
`Class`	Diese Dimension ist für future reserviert.

Überwachen von AWS FIS-Experimenten mit Amazon EventBridge

Wenn sich der Status eines Experiments ändert, gibt AWS FIS eine Benachrichtigung aus. Diese Benachrichtigungen werden als Ereignisse über Amazon EventBridge (früher als CloudWatch Ereignisse bezeichnet) zur Verfügung gestellt. AWS FIS gibt diese Ereignisse nach bestem Bemühen aus. Ereignisse werden nahezu EventBridge in Echtzeit an übermittelt.

Mit können Sie Regeln erstellen EventBridge, 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 Maßnahmen zu ergreifen.

Weitere Informationen zu EventBridgefinden Sie unter Erste Schritte mit Amazon EventBridge im Amazon- EventBridge Benutzerhandbuch.

Im Folgenden finden Sie die Syntax eines Ereignisses zur Änderung des Experimentstatus:

```
{
    "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 Status des Experiments. Die möglichen Werte sind:

- `initiating`
- `pending`
- `running`
- `stopping`

Experimentprotokollierung für AWS FIS

Sie können die Experimentprotokollierung verwenden, um während der Ausführung detaillierte Informationen zu Ihrem Experiment zu erfassen.

Die Protokollierung von Experimenten erfolgt auf der Grundlage der Kosten, die mit den einzelnen Protokollzieltypen verbunden sind. Weitere Informationen finden Sie unter Amazon- CloudWatch Preise (unter Zahlungspflichtiges Kontingent, Protokolle, Vended Logs) und Amazon S3-Preise.

Berechtigungen

Sie müssen AWS FIS-Berechtigungen erteilen, um Protokolle an jedes von Ihnen konfigurierte Protokollziel zu senden. Weitere Informationen finden Sie im Amazon- CloudWatch Logs-Benutzerhandbuch:

- An CloudWatch Protokolle gesendete Protokolle
- An Amazon S3 gesendete Protokolle

Protokollschema

Im Folgenden finden Sie das Schema, das bei der Experimentprotokollierung verwendet wird. Die aktuelle Schemaversion ist 2. Die Felder für `details`hängen vom Wert von a`log_type`. Die Felder für `resolved_targets`hängen vom Wert von a`target_type`. Weitere Informationen finden Sie unter the section called “Beispielprotokolldatensätze”.

```
{
    "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 Folgendes ein:
  - Das `target_type` Feld und ändert das `resolved_targets` Feld von einer Liste von ARNs in eine Liste von Objekten. Die gültigen Felder für das `resolved_targets` Objekt hängen vom Wert von ab`target_type`, der der Ressourcentyp der Ziele ist.
  - Die `target-resolution-detail` Ereignistypen `action-error` und , die das `account_id` Feld hinzufügen.
- Version 1 ist die erste Version.

Protokollziele

AWS FIS unterstützt die Protokollzustellung an die folgenden Ziele:

- Ein Amazon-S3-Bucket
- Eine Amazon- CloudWatch Logs-Protokollgruppe

S3-Protokollbereitstellung

Die Protokolle werden an den folgenden Speicherort geliefert.

```
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 einige Minuten dauern, bis die Protokolle an den Bucket übermittelt werden.

CloudWatch Protokollzustellung

Die Protokolle werden an einen Protokollstream mit dem Namen /aws/fis/*experiment-id* übermittelt.

Protokolle werden in weniger als einer Minute an die Protokollgruppe übermittelt.

Beispielprotokolldatensätze

Im Folgenden finden Sie Beispielprotokolldatensätze für ein Experiment, das die aws:ec2:reboot-instances Aktion auf einer zufällig ausgewählten EC2-Instance ausführt.

Datensätze

- Experimentstart
- target-resolution-start
- target-resolution-detail
- target-resolution-end
- action-start
- action-end
- action-error
- Experimentende

Experimentstart

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 größer als 1 `total_pages` 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 für eine 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-0f7ee2abffc330de5"
            }
        ],
        "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
    }
}
```

action-start

Im Folgenden finden Sie einen Beispieldatensatz für das `action-start` Ereignis. Wenn die Experimentvorlage Parameter für die Aktion angibt, 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"}
    }
}
```

action-error

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, in 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]:"
        }
    }
}
```

Aktionsendpunkt

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."
        }
    }
}
```

Experimentende

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 der Experimentprotokollierung

Die Experimentprotokollierung ist standardmäßig deaktiviert. Um Experimentprotokolle für ein Experiment zu erhalten, müssen Sie das Experiment aus einer Experimentvorlage mit aktivierter Protokollierung erstellen. Wenn Sie zum ersten Mal ein Experiment ausführen, das für die

Verwendung eines Ziels konfiguriert ist, 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.

So aktivieren Sie die Experimentprotokollierung mithilfe der Konsole

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, Experimentvorlage aktualisieren.
4. Konfigurieren Sie für Protokolle die Zieloptionen. Um Protokolle an einen S3-Bucket zu senden, wählen Sie An einen Amazon S3-Bucket senden und geben Sie den Bucket-Namen und das Präfix ein. Um Protokolle an CloudWatch Protokolle zu senden, wählen Sie An CloudWatch Protokolle senden und geben Sie die Protokollgruppe ein.
5. Wählen Sie Experimentvorlage aktualisieren aus.

So aktivieren Sie die Experimentprotokollierung mit der AWS CLI

Verwenden Sie den update-experiment-template Befehl und geben Sie eine Protokollkonfiguration an.

Deaktivieren der Experimentprotokollierung

Wenn Sie keine Protokolle mehr für Ihre Experimente erhalten möchten, können Sie die Experimentprotokollierung deaktivieren.

So deaktivieren Sie die Experimentprotokollierung mithilfe der Konsole

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, Experimentvorlage aktualisieren.
4. Deaktivieren Sie für Protokolle die Optionen An einen Amazon S3-Bucket senden und An CloudWatch Protokolle senden.
5. Wählen Sie Experimentvorlage aktualisieren aus.

So deaktivieren Sie die Experimentprotokollierung mithilfe der AWS CLI

Verwenden Sie den -update-experiment-templateBefehl und geben Sie eine leere Protokollkonfiguration an.

API-Aufrufe mit AWS CloudTrail protokollieren

AWS Fault Injection Service (AWS FIS) ist in integriert, einem ServiceAWS CloudTrail, der die Aktionen eines Benutzers, einer Rolle oder eines -AWSServices in AWS FIS. CloudTrail captures aller API-Aufrufe für AWS FIS als Ereignisse aufzeichnet. Zu den erfassten Aufrufen gehören Aufrufe von der AWS FIS-Konsole und Codeaufrufe der AWS FIS-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3-Bucket aktivieren, einschließlich Ereignissen für AWS FIS. Wenn Sie keinen Trail konfigurieren, können Sie trotzdem die neuesten Ereignisse in der CloudTrail Konsole unter Ereignisverlauf anzeigen. Anhand der von CloudTrailgesammelten Informationen können Sie die an AWS FIS gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Weitere Informationen zu CloudTrailfinden Sie im AWS CloudTrail -Benutzerhandbuch.

Verwenden von CloudTrail

CloudTrail wird beim Erstellen des Kontos AWS-Konto auf Ihrem aktiviert. Wenn eine Aktivität in AWS FIS auftritt, wird diese Aktivität in einem - CloudTrail Ereignis zusammen mit anderen -AWSServiceereignissen im Ereignisverlauf aufgezeichnet. Sie können in Ihrem AWS-Konto die neusten Ereignisse anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Anzeigen von Ereignissen mit dem CloudTrail Ereignisverlauf.

Erstellen Sie für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich Ereignissen für AWS FIS, einen Trail. Ein Trail ermöglicht CloudTrail die Bereitstellung von Protokolldateien an einen Amazon S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie andere -AWSServices konfigurieren, um die in den CloudTrail Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie hier:

- Erstellen eines Trails für Ihr AWS Konto
- CloudTrail Unterstützte Services und Integrationen
- Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail

- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien aus mehreren Konten

Alle AWS FIS-Aktionen werden von protokolliert CloudTrail und sind in der APIAWS-Referenz zum Fault Injection Service dokumentiert. Informationen zu den Experimentaktionen, die für eine Zielressource ausgeführt werden, finden Sie in der API-Referenzdokumentation für den Service, dem die Ressource gehört. Informationen zu Aktionen, die auf einer Amazon EC2-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 Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter CloudTrail userIdentity-Element.

FISAWS-Protokolldateieinträge verstehen

Ein Trail ist eine Konfiguration, die die Bereitstellung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon 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, das Datum und die Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Im Folgenden finden Sie ein Beispiel für einen CloudTrail Protokolleintrag für einen Aufruf der AWS - FIS-`StopExperiment`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 CloudTrail Beispielprotokolleintrag für eine API-Aktion, die AWS FIS als Teil eines Experiments aufgerufen hat, das die `aws:ssm:send-command` AWS FIS-Aktion enthält. Das -`userIdentity`Element spiegelt eine Anforderung mit temporären Anmeldeinformationen wider, die durch Übernahme einer Rolle abgerufen wurden. Der Name der übernommenen Rolle wird in angezeigt`userName`. Die ID des Experiments, EXP21nT17WMzA6dnUgz, erscheint in `principalId` und als Teil des ARN der angenommenen Rolle.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROATZZZ4JPIXUEXAMPLE:EXP21nT17WMzA6dnUgz",
        "arn": "arn:aws:sts::111122223333:assumed-role/AllowActions/
EXP21nT17WMzA6dnUgz",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROATZZZ4JPIXUEXAMPLE",
                "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"
}
```

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 selbst 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

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).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- 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 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies AWS gilt auch, wenn Sie mit FIS oder anderen Geräten arbeiten und die Konsole, die AWS-Services API oder SDKs verwenden. AWS CLI AWS 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 unterUnterstützte 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

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 Fuktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern 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. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer 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 als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center - Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff: Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe

Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.

- Temporäre IAM-Benutzerberechtigungen: Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle: Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, 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.
  - Dienstbezogene Rolle — 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. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

- Auf Amazon EC2 ausgeführte Anwendungen — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) 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 ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

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 Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

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

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen zu ACLs finden Sie unter Zugriffssteuerungsliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen:Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service Control Policies (SCPs) — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations -Benutzerhandbuch.
- Sitzungsrichtlinien:Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und

der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Bewertungslogik für Richtlinien.

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 (services pezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Ja
Servicerollen	Ja

IAM-Feature	AWS FIS-Unterstützung
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 Erstellen von IAM-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. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. 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. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden im IAM-Benutzerhandbuch.

Politische Maßnahmen für AWS FIS

Unterstützt Richtlinienaktionen	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation 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 bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, 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 die ARNs durch Kommata voneinander.

```
"Resource": [
    "resource1",
    "resource2"
]
```

Eine Liste der AWS FIS-Ressourcentypen und ihrer ARNs finden Sie unter Vom AWS Fault Injection Service definierte Ressourcentypen 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 kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. 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.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. 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 in FIS AWS

Unterstützt ACLs	Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte 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 definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement 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 Was ist ABAC? 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

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln zu einer Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipalberechtigungen für AWS FIS

Unterstützt Forward Access Sessions (FAS)	Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen 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 dienstbezogene Rollen für den Fault Injection AWS 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. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die

sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Einzelheiten zu den von AWS FIS definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für jeden Ressourcentyp, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für den AWS Fault Injection Service in der Service Authorization Reference.

Inhalt

Bewährte Methoden 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. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst

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 AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen 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. AWS 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 Richtlinienvalidierung zum 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 Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden 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.

```
{
    "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.

```
{
    "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`.

```
{
  "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::account-id: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 dienstbezogene Rollen für den Fault Injection AWS Service.

```
{
  "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.

```
{
    "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`.

```
{
    "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`**

Die folgende Beispielrichtlinie verwendet den `ec2:FisTargetArns` Bedingungsschlüssel, 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`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:InjectApiError",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:api-insufficient-instance-capacity-error",
                    ],
                    "ec2:FisTargetArns": [
                        "arn:aws:iam:*:*:role:role-name"
```

```
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:InjectApiError",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:asg-insufficient-instance-capacity-error"
                    ],
                    "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**

In der folgenden Beispielrichtlinie wird der `S3:IsReplicationPauseRequest` Bedingungsschlüssel verwendet, um die FIS im Rahmen der AWS FIS-Aktion zuzulassen `PutReplicationConfiguration` und `GetReplicationConfiguration` nur dann, wenn sie von der AWS FIS ausgeführt wird. `aws:s3:bucket-pause-replication`

```
{
    "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": "*"
        }
    ]
    }
```

Beispiel: Experimentieren Sie mit der Rolle mit Ausführungsberechtigungen **aws:dynamodb:encrypted-global-table-pause-replication**

Die folgende Beispielrichtlinie gewährt AWS FIS die erforderlichen Berechtigungen, um ein Experiment mit einer einzigen Aktion `aws:dynamodb:encrypted-global-table-pause-replication` auszuführen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DynamoDB",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:DescribeGlobalTable"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-2:123456789012:table/MyEncryptedGlobalTable"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/fis-enabled": "true"
                }
            }
        },
        {
            "Sid": "Tagging",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "KMS",
            "Effect": "Allow",
            "Action": [
                "kms:PutKeyPolicy",
                "kms:DescribeKey",
                "kms:GetKeyPolicy"
```

```
            ],
            "Resource": "arn:aws:kms:us-east-2:123456789012:key/
MyGlobalTableEncryptionKey"
        },
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/fis-enabled": "true"
            }
        }
    ]
}
```

> **Note**
>
> AWS FIS wird verwendet`kms:PutKeyPolicy`, um den Zugriff auf DynamoDB, auf den vom Kunden verwalteten AWS KMS Schlüssel, zu verweigern, wodurch die Replikation gestoppt wird. Wir empfehlen, die Rolle nur zu verwenden, wenn Sie aktiv ein Experiment mit dieser Aktion ausführen. Andernfalls empfehlen wir, sie zu löschen. Durch das Löschen der Rolle werden die FIS-Berechtigungen für entfernt. `kms:PutKeyPolicy` Suchen Sie nach Abschluss des Experiments die Rolle in den Details der Experimentvorlage. Wählen Sie in der IAM-Konsole den Link zur IAM-Rolle aus und klicken Sie auf Löschen. Navigieren Sie nach dem Löschen der Rolle zur AWS KMS Konsole und suchen Sie den AWS KMS Schlüssel, der zum Schutz von Daten verwendet wird, in der DynamoDB-Zieltabelle. Stellen Sie sicher, dass die AWS KMS Schlüsselrichtlinie Ihren Erwartungen entspricht. Sie sollten keine AWS FIS-Erklärung mehr sehen (z. B.`FIS_DDB_PAUSE_REPLICATION-EXP123456789012345_DO_NOT_MODIFY`).

`aws:dynamodb:encrypted-global-table-pause-replication`FIS-Aktionen fügen der Richtlinie für den KMS-Schlüssel, der zum Schutz von Daten in den globalen DynamoDB-Zieltabellen verwendet wird, dynamisch die folgenden Berechtigungen hinzu:

```
{
    "Sid": "DO_NOT_MODIFY_FIS_DDB_PAUSE_REPLICATION-EXP123456789012345",
    "Effect": "Deny",
    "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
        },
    "Action": [
```

```
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "kms:EncryptionContext:aws:dynamodb:tableName": [
                    "transactions-global-table",
                    "inventory-global-table"
                ]
            }
        }
    }
```

Diese Berechtigungen werden am Ende des vorhandenen AWS KMS wichtigen Richtliniendokuments angehängt. Mit der obigen Richtlinienanweisung werden der dienstverknüpften DynamoDB-Rolle die Berechtigungen zum Replizieren von Daten in und aus den im Kontextschlüssel aufgeführten Tabellen entzogen. `kms:EncryptionContext:aws:dynamodb:tableName` Im obigen Beispiel würde die Replikation für globale DynamoDB-Tabellen mit den Namen:`transaction-global-table`, beendet. `inventory-global-table`

Verwenden Sie dienstbezogene Rollen für den Fault Injection AWS 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 zur dienstbezogenen Rolle müssen Sie auch eine IAM-Rolle angeben, die die Berechtigung zum Ändern der Ressourcen erteilt, 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 angegebene dienstbezogene Rolle AWSServiceRoleForFIS, um die Überwachung und die Ressourcenauswahl für Experimente zu verwalten.

Die AWSServiceRoleForFISdienstbezogene Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `fis.amazonaws.com`

Die AWSServiceRoleForFISserviceverknüpfte Rolle verwendet die verwaltete Richtlinie AmazonFIS. ServiceRolePolicy Diese Richtlinie ermöglicht es AWS FIS, die Überwachung und die Auswahl der Ressourcen für Experimente zu verwalten. Weitere Informationen finden Sie unter AmazonFIS ServiceRolePolicy in der Referenz zu AWS verwalteten Richtlinien.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Damit die AWSServiceRoleForFISserviceverknüpfte 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.

```
{
    "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 Management Console, 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 serviceverknüpfte Rolle zu bearbeiten. AWSServiceRoleForFIS Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen Sie eine dienstverknüpfte 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 AWS FIS-Ressourcen zu bereinigen, die von AWSServiceRoleForFIS

Vergewissern Sie sich, dass derzeit keines Ihrer Experimente läuft. Falls nötig, beenden Sie Ihre Experimente. Weitere Informationen finden Sie unter Stoppen eines Experiments.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die AWSServiceRoleForFISserviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für dienstverknüpfte AWS FIS-Rollen

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 kundenverwaltete 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: AmazonFIS ServiceRolePolicy

Diese Richtlinie ist der dienstbezogenen Rolle zugeordnet, die benannt wurde AWSServiceRoleForFIS, damit AWS FIS die Überwachung und die Auswahl der Ressourcen für Experimente verwalten kann. Weitere Informationen finden Sie unter Verwenden Sie dienstbezogene Rollen für den Fault Injection AWS Service.

AWS verwaltete Richtlinie: AWSFaultInjectionSimulatorEC2Access

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 “Experimentrolle”.

Die Berechtigungen für diese Richtlinie finden Sie AWSFaultInjectionSimulatorEC2Accessin der Referenz zu AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSFaultInjectionSimulatorECSAccess

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 “Experimentrolle”.

Die Berechtigungen für diese Richtlinie finden Sie AWSFaultInjectionSimulatorECSAccessin der Referenz zu AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSFaultInjectionSimulatorEKSAccess

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 “Experimentrolle”.

Die Berechtigungen für diese Richtlinie finden Sie AWSFaultInjectionSimulatorEKSAccessin der Referenz zu AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSFaultInjectionSimulatorNetworkAccess

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 “Experimentrolle”.

Die Berechtigungen für diese Richtlinie finden Sie AWSFaultInjectionSimulatorNetworkAccessin der Referenz zu AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSFaultInjectionSimulatorRDSAccess

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 “Experimentrolle”.

Die Berechtigungen für diese Richtlinie finden Sie AWSFaultInjectionSimulatorRDSAccessin der Referenz zu AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSFaultInjectionSimulatorSSMAccess

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 “Experimentrolle”.

Die Berechtigungen für diese Richtlinie finden Sie AWSFaultInjectionSimulatorSSMAccessin der Referenz zu AWS verwalteten Richtlinien.

AWS FIS aktualisiert AWS verwaltete Richtlinien

Hier finden Sie Informationen zu Aktualisierungen der AWS verwalteten Richtlinien für AWS FIS, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen.

Änderung	Beschreibung	Datum
AWSFaultInjectionSimulatorECSAccess – 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 AWS es FIS ermöglichen, Experimente mit den aws:network:transit-gateway-disrupt-cross-region-connectivity Aktionen aws:network:route-table-disrupt-cross-region-connectivity und durchzuführen.	25. Januar 2024
AWSFaultInjectionSimulatorEC2Access – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, EC2-Instances aufzulösen.	13. November 2023
AWSFaultInjectionSimulatorEKSAccess – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, damit AWS FIS EKS-Ziele auflösen kann.	13. November 2023

Änderung	Beschreibung	Datum
AWSFaultInjectionSimulatorRDSAccess – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, RDS-Ziele aufzulösen.	13. November 2023
AWSFaultInjectionSimulatorEC2Access – Aktualisierung auf eine bestehende 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. aws:ecs:task	01. Juni 2023
AWSFaultInjectionSimulatorEKSAccess – Aktualisierung auf eine bestehende Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, Experimente mit den neuen aws:eks:pod Aktionen durchzuführen.	01. Juni 2023
AWSFaultInjectionSimulatorEC2Access – 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
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

Änderung	Beschreibung	Datum
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
AWSFaultInjectionSimulatorSSMAccess – 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
AmazonFIS ServiceRolePolicy — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, Subnetze zu beschreiben.	26. Oktober 2022
AmazonFIS ServiceRolePolicy — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, EKS-Cluster zu beschreiben.	7. Juli 2022
AmazonFIS ServiceRolePolicy — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, damit AWS FIS die Aufgaben in Ihren Clustern auflisten und beschreiben kann.	7. Februar 2022

Änderung	Beschreibung	Datum
AmazonFIS ServiceRolePolicy — Aktualisierung einer bestehenden Richtlinie	Die `events:ManagedBy` Bedingung für die Aktion wurde entfernt. `events:DescribeRule`	6. Januar 2022
AmazonFIS ServiceRolePolicy — Aktualisierung einer bestehenden Richtlinie	Es wurden Berechtigungen hinzugefügt, die es AWS FIS ermöglichen, den Verlauf der CloudWatch Alarme 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.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

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-Endpunkte basieren auf einer Technologie AWS PrivateLink, mit der Sie privat auf AWS FIS-APIs zugreifen können, 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 Dienstnamen:. `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-Endpunktrichtlinie für AWS FIS

Sie können Ihrem VPC-Endpunkt eine Endpunktrichtlinie 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 im Handbuch unter Steuern des Zugriffs auf VPC-Endpunkte mithilfe von Endpunktrichtlinien.AWS PrivateLink

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" ]
         }
      }
   ]
}
```

Markieren Ihrer AWS FIS-Ressourcen

Ein Tag ist ein Metadaten-Label, das Sie oder AWS einer AWS-Ressource 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:

- Identify and organize your AWS resources. Viele AWS-Services unterstützen das Markieren mit Tags (kurz: Tagging). So können Ressourcen aus verschiedenen Services dasselbe Tag zuweisen, um anzugeben, dass die Ressourcen verbunden 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 nicht `aws:` als Präfix für Schlüssel verwenden, da es für reserviert istAWS.

Arbeiten mit Tags

Die folgenden AWS FIS-Ressourcen (AWSFault Injection Service) unterstützen das Markieren:

- Aktionen
- Experimente
- Experimentvorlagen

Sie können die Konsole verwenden, um mit Tags für Experimente und Experimentvorlagen 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 – Fügen Sie einer Ressource Tags hinzu.
- untag-resource – Entfernen Sie Tags aus einer Ressource.
- list-tags-for-resource – Listet die Tags für eine bestimmte Ressource auf.

Kontingente und Einschränkungen für AWS Fault Injection Service

Ihr AWS-Konto verfügt über Standardkontingente, die früher als Limits bezeichnet wurden, für jeden AWS Service. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region. Sie können Erhöhungen für einige Kontingente beantragen aber nicht für alle Kontingente.

Um die Kontingente für AWS FIS anzuzeigen, öffnen Sie die Service Quotas-Konsole. Wählen Sie im Navigationsbereich -AWS Services und dann AWS Fault Injection Service aus.

Informationen zur Erhöhung eines Kontingents finden Sie unter Anfordern einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Ihr AWS-Konto verfügt über die folgenden Kontingente im Zusammenhang mit AWS FIS.

Name	Standard	Anpas	Beschreibung
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.
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.

Name	Standard	Anpas	Beschreibung
Datenaufbewahrung abgeschlossener Experimente in Tagen	Jede unterstützte Region: 120	Nein	Die maximale Anzahl von Tagen, die AWS FIS berechtigt ist, Daten über abgeschlossene Experimente in diesem Konto in der aktuellen Region beizubehalten.
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.
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:route-table-disrupt-cross-region-connectivity	Jede unterstützte Region: 15	Nein	Die maximale Anzahl von verwalteten Präfixlisten, die aws:network:route-table-disrupt-cross-region-connectivity pro Aktion zulässt.
Maximale Anzahl von Routing-Tabellen in aws:network:route-table-disrupt-cross-region-connectivity	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von Routing-Tabellen, die aws:network:route-table-disrupt-cross-region-connectivity pro Aktion zulässt.

Name	Standard	Anpas	Beschreibung
Maximale Anzahl von Routen in aws:network:route-table-disrupt-cross-region-connectivity	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.
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-Auto Scaling-Gruppen für aws:ec2:asg-insufficient-instance-capacity-error	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Auto Scaling-Gruppen, auf die aws:ec2:asg-insufficient-instance-capacity-error abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.

Name	Standard	Anpas	Beschreibung
Ziel-Buckets für aws:s3:bucket-pause-replication	Jede unterstützte Region: 20	Ja	Die maximale Anzahl von S3-Buckets, auf die aws:s3:can abzielen bucket-pause-replication kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-Cluster für aws:ecs:drain-container-instances	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Clustern, auf die aws:ecs:drain-container-instances can abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-Cluster für aws:rds:failover-db-cluster	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Clustern, auf die aws:rds:failover-db-cluster can abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-DBInstances für aws:rds:reboot-db-instances	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von DBInstances, auf die aws:rds:reboot-db-instances can abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.

Name	Standard	Anpas	Beschreibung
Ziel-Instances für aws:ec2:reboot-instances	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Instances, auf die aws:ec2:reboot-instances abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-Instances für aws:ec2:stop-instances	Jede unterstützte Region: 5	Yes (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: 5	Yes (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: 5	Yes (Ja)	Die maximale Anzahl von Instances, auf die aws:ssm:send-command abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.

Name	Standard	Anpas	Beschreibung
Zielknotengruppen für aws:eks:terminate-nodegroup-instances	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Knotengruppen, auf die aws:eks:terminate-nodegroup-instances can abzielt, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Ziel-Pods für aws:eks:pod-cpu-stress	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Pods, auf die aws:eks:pod-cpu-stress can abzielen kann, wenn Sie Ziele mithilfe von Parametern identifizieren, pro Experiment.
Ziel-Pods für aws:eks:pod-delete	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Pods, auf die aws:eks:pod-delete abzielen kann, wenn Sie Ziele mithilfe von Parametern identifizieren, pro Experiment.
Ziel-Pods für aws:eks:pod-io-stress	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Pods, auf die aws:eks:pod-io-stress can abzielt, wenn Sie Ziele mithilfe von Parametern identifizieren, pro Experiment.

Name	Standard	Anpas	Beschreibung
Ziel-Pods für aws:eks:pod-memory-stress	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Pods, auf die aws:eks:pod-memory-stress can abzielen kann, wenn Sie Ziele mithilfe von Parametern identifizieren, pro Experiment.
Ziel-Pods für aws:eks:pod-network-blackhole-port	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Pods, auf die aws:eks:pod-network-blackhole-port can abzielt, wenn Sie Ziele mithilfe von Parametern identifizieren, pro Experiment.
Ziel-Pods für aws:eks:pod-network-latency	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Pods, auf die aws:eks:pod-network-latency can abzielt, wenn Sie Ziele mithilfe von Parameter n identifizieren, pro Experiment.
Ziel-Pods für aws:eks:pod-network-packet-loss	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Pods, auf die aws:eks:pod-network-packet-loss can abzielt, wenn Sie Ziele mithilfe von Parametern identifizieren, pro Experiment.

Name	Standard	Anpas	Beschreibung
Ziel ReplicationGroups für aws:elasticache:interrupt-cluster-az-power	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von aws:elasticache:interrupt-cluster-az-power can target ReplicationGroups , wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Ziel SpotInstances für aws:ec2:send-spot-instance-interruptions	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von SpotInstances aws:ec2:send-spot-instance-interruptions can-Zielen, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Zielsubnetze für aws:network:disrupt-connectivity	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Subnetzen, auf die aws:network:disrupt-connectivity abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Zielsubnetze für aws:network:route-table-disrupt-cross-region-connectivity	Jede unterstützte Region: 6	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.

Name	Standard	Anpas	Beschreibung
Zielaufgaben für aws:ecs:stop-task	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Aufgaben, auf die aws:ecs:stop-task abzielen kann, wenn Sie Ziele mithilfe von Tags identifizieren, pro Experiment.
Zielaufgaben für aws:ecs:task-cpu-stress	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Aufgaben, auf die aws:ecs:task-cpu-stress can abzielen kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Zielaufgaben für aws:ecs:task-io-stress	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Aufgaben, auf die aws:ecs:task-io-stress can abzielen kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Zielaufgaben für aws:ecs:task-kill-process	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Aufgaben, auf die aws:ecs:task-kill-process can abzielen kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.

Name	Standard	Anpas	Beschreibung
Zielaufgaben für aws:ecs:task-network-blackhole-port	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Aufgaben, auf die aws:ecs:task-network-blackhole-port can abzielen kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Zielaufgaben für aws:ecs:task-network-latency	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Aufgaben, auf die aws:ecs:task-network-latency can abzielen kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Zielaufgaben für aws:ecs:task-network-packet-loss	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Aufgaben, auf die aws:ecs:task-network-packet-loss can abzielen kann, wenn Sie Ziele mithilfe von Tags/Parametern identifizieren, pro Experiment.
Ziel TransitGateways für aws:network:transit-gateway-disrupt-cross-region-connectivity	Jede unterstützte Region: 5	Yes (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.

Name	Standard	Anpas	Beschreibung
Zielkontokonfigurationen pro Experimentvorlage	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Zielkontokonfigurationen, die Sie für eine Experimentvorlage in diesem Konto in der aktuellen Region erstellen können.

Ihre Nutzung von AWS FIS unterliegt den folgenden zusätzlichen Einschränkungen:

Name	Einschränkung
Ziele für `aws:elasticache:interrupt-cluster-az-power` Aktionen	Begrenzt auf 10 `aws:elasticache:redis-replicationgroup` Cluster, die pro Tag pro Konto und Region beeinträchtigt sind. Sie können eine Erhöhung beantragen, indem Sie einen Support-Fall in der AWS Support Center Console erstellen.

Dokumentverlauf

In der folgenden Tabelle werden wichtige Aktualisierungen der Dokumentation im AWS Fault Injection Service-Benutzerhandbuch beschrieben.

Änderung	Beschreibung	Datum
Neue Experimentieroption für den Aktionsmodus	Sie können den Aktionsmodus auf einstellen, `skip-all` um vor der Ausführung eines Experiments eine Zielvorschau zu generieren.	13. März 2024
AWS verwaltete Richtlinienaktualisierungen	AWS FIS 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-	30. November 2023

	cross-region-connectivity Aktion jetzt verwenden.	
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
Namensänderung von FIS AWS	Der Dienstname wurde auf AWS Fault Injection Service aktualisiert.	15. November 2023
AWS verwaltete Richtlinienaktualisierungen	AWS FIS hat bestehende verwaltete Richtlinien aktualisiert.	13. November 2023
Neue Szenario-Bibliothek	Sie können jetzt die Funktion der AWS FIS-Szenariobibliothek verwenden.	7. November 2023
Neuer Experimentplaner	Sie können jetzt die AWS FIS-Funktion zur Planung von Experimenten verwenden.	7. November 2023

AWS verwaltete Richtlinienaktualisierungen	AWS FIS hat bestehende verwaltete Richtlinien aktualisiert.	02. Juni 2023
Neue Aktionen	Sie können die neuen aws:ecs:task und aws:eks:pod Aktionen verwenden.	01. Juni 2023
AWS verwaltete Richtlinienaktualisierungen	AWS FIS hat bestehende verwaltete Richtlinien aktualisiert.	01. Juni 2023
Neues vorkonfiguriertes SSM-Dokument	Sie können das folgende vorkonfigurierte SSM-Dokument verwenden: -Run-Disk-Fill. AWSFIS	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, anzuhalten.	27. Januar 2023
Neue Aktion	Sie können die aws:network:disrupt-connectivity Aktion verwenden, um bestimmte Arten von Datenverkehr in die Zielsubnetze 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 Versuchsvorlagen 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, -Run-Network-Blackhold-Port, -Run-Network-Latency-SOURCES, -Run-Network-Packet-Loss und -Run-Network-Packet-Loss-SOURCES AWSFIS. AWSFIS AWSFIS AWSFIS	4. November 2021
Neue Aktion	Sie können die aws:ec2:send-spot-instance-interruptions Aktion verwenden, um eine Benachrichtigung über eine Unterbrechung der Spot-Instance 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.