aws

Benutzerhandbuch

Application Auto Scaling

Application Auto Scaling: Benutzerhandbuch

Table of Contents

Was ist Application Auto Scaling? 1
Merkmale von Application Auto Scaling 2
Arbeiten Sie mit Application Auto Scaling 2
Erste Schritte 4
Weitere Informationen 6
Dienste, die mit Application Auto Scaling zusammenarbeiten 7
Amazon AppStream 2.0 9
Servicegebundene Rolle 10
Dienstauftraggeber 10
Registrierung von AppStream 2.0-Flotten als skalierbare Ziele mit Application Auto Scaling 10
Zugehörige Ressourcen 11
Amazon Aurora 11
Servicegebundene Rolle 12
Dienstauftraggeber 12
Registrierung von Aurora DB-Clustern als skalierbare Ziele mit Application Auto Scaling 12
Zugehörige Ressourcen 13
Amazon Comprehend 13
Servicegebundene Rolle 14
Dienstauftraggeber 14
Registrierung von Amazon Comprehend Ressourcen als skalierbare Ziele mit Application Auto Scaling 14
Zugehörige Ressourcen 16
Amazon DynamoDB 16
Servicegebundene Rolle 16
Dienstauftraggeber 16
Registrieren von DynamoDB-Ressourcen als skalierbare Ziele mit Application Auto Scaling 17
Zugehörige Ressourcen 19
Amazon ECS 20
Servicegebundene Rolle 20
Dienstauftraggeber 20
Registrierung von ECS-Diensten als skalierbare Ziele mit Application Auto Scaling 20
Zugehörige Ressourcen 21

Amazon ElastiCache ... 22
Servicegebundene Rolle ... 22
Dienstauftraggeber ... 22
Registrierung ElastiCache für Redis-Replikationsgruppen als skalierbare Ziele mit Application Auto Scaling ... 23
Zugehörige Ressourcen ... 24
Amazon Keyspaces (für Apache Cassandra) ... 24
Servicegebundene Rolle ... 25
Dienstauftraggeber ... 25
Registrierung von Amazon Keyspaces-Tabellen als skalierbare Ziele mit Application Auto Scaling ... 25
Zugehörige Ressourcen ... 27
AWS Lambda ... 27
Servicegebundene Rolle ... 27
Dienstauftraggeber ... 27
Registrieren von Lambda-Funktionen als skalierbare Ziele mit Application Auto Scaling ... 28
Zugehörige Ressourcen ... 29
Amazon Managed Streaming for Apache Kafka (MSK) ... 29
Servicegebundene Rolle ... 29
Dienstauftraggeber ... 29
Registrierung von Amazon MSK-Cluster-Speicher als skalierbare Ziele mit Application Auto Scaling ... 30
Zugehörige Ressourcen ... 31
Amazon Neptune ... 31
Servicegebundene Rolle ... 31
Dienstauftraggeber ... 31
Registrierung von Neptune-Clustern als skalierbare Ziele mit Application Auto Scaling ... 32
Zugehörige Ressourcen ... 33
Amazon SageMaker ... 33
Servicegebundene Rolle ... 33
Dienstauftraggeber ... 33
Registrieren von SageMaker Endpunktvarianten als skalierbare Ziele mit Application Auto Scaling ... 34
Registrieren der bereitgestellten Gleichzeitigkeit von Serverless-Endpunkten als skalierbare Ziele mit Application Auto Scaling ... 35
Registrieren von Inferenzkomponenten als skalierbare Ziele mit Application Auto Scaling ... 36

Zugehörige Ressourcen ... 37
Amazon EC2-Spot-Flotte ... 37
Servicegebundene Rolle ... 37
Dienstauftraggeber ... 38
Registrierung von Spot Fleets als skalierbare Ziele mit Application Auto Scaling ... 38
Zugehörige Ressourcen ... 39
Benutzerdefinierte Ressourcen ... 39
Servicegebundene Rolle ... 39
Dienstauftraggeber ... 40
Registrierung von benutzerdefinierten Ressourcen als skalierbare Ziele mit Application Auto Scaling ... 40
Zugehörige Ressourcen ... 41
Einrichten ... 42
Melden Sie sich an für AWS ... 42
Richten Sie ein AWS CLI ... 43
Verwenden AWS CloudShell ... 44
Konfigurieren der Skalierung mit AWS CloudFormation ... 46
Application Auto Scaling und AWS CloudFormation Vorlagen ... 46
Beispielvorlagen-Snippets ... 47
Weitere Informationen über AWS CloudFormation ... 47
Geplante Skalierung ... 48
So funktioniert die geplante Skalierung ... 49
Funktionsweise ... 49
Überlegungen ... 49
Häufig verwendete Befehle ... 50
Zugehörige Ressourcen ... 51
Einschränkungen ... 51
Verwenden von Cron-Ausdrücken ... 52
Beispiel für geplante Aktionen ... 55
Erstellen einer geplanten Aktion, die nur einmal ausgeführt wird ... 55
Erstellen einer geplanten Aktion, die in einem wiederkehrenden Intervall ausgeführt wird ... 57
Erstellen einer geplanten Aktion, die nach einem wiederkehrenden Zeitplan ausgeführt wird ... 58
Erstellen einer einmaligen geplanten Aktion, die eine Zeitzone angibt ... 58
Erstellen einer wiederkehrenden geplanten Aktion, die eine Zeitzone angibt ... 59
Verwalten der geplanten Skalierung ... 60

Anzeige der Skalierungsaktivitäten für einen bestimmten Service ... 61
Beschreiben aller geplanten Aktionen für einen bestimmten Dienst ... 62
Beschreiben einer oder mehrerer geplanter Aktionen für ein skalierbares Ziel ... 64
Ausschalten der geplanten Skalierung für ein skalierbares Ziel ... 65
Löschen einer geplanten Aktion ... 66
Tutorial: Erste Schritte mit der geplanten Skalierung mit AWS CLI ... 67
Schritt 1: Registrieren Sie Ihr skalierbares Ziel ... 68
Schritt 2: Erstellen Sie zwei geplante Aktionen ... 69
Schritt 3: Ansicht der Skalierungsaktivitäten ... 72
Schritt 4: Nächste Schritte ... 76
Schritt 5: Bereinigen ... 76
Skalierungsrichtlinien für die Ziel-Nachverfolgung ... 78
Wie funktioniert die Zielverfolgung ... 79
Funktionsweise ... 79
Auswahl von Metriken ... 81
Definieren des Zielwerts ... 82
Ruhephasen definieren ... 82
Überlegungen ... 84
Mehrere Skalierungsrichtlinien ... 84
Häufig verwendete Befehle ... 85
Zugehörige Ressourcen ... 86
Einschränkungen ... 86
Erstellen einer Zielverfolgungs-Skalierungsrichtlinie ... 87
Registrieren eines skalierbaren Ziels ... 87
Erstellen einer Zielverfolgungs-Skalierungsrichtlinie ... 88
Beschreiben Sie die Zielverfolgungs-Skalierungsrichtlinien ... 90
Löschen einer Zielverfolgungs-Skalierungsrichtlinie ... 92
Verwenden von Metrikberechnungen ... 92
Beispiel: Amazon-SQS-Warteschlangenrückstand pro Aufgabe ... 93
Einschränkungen ... 98
Richtlinien zur schrittweisen Skalierung ... 99
Wie funktioniert Step Scaling ... 100
Funktionsweise ... 100
Schrittweise Anpassungen ... 101
Skalierungsanpassungstypen ... 103
Ruhephase ... 105

Häufig verwendete Befehle ... 106
Überlegungen ... 106
Zugehörige Ressourcen ... 51
Einschränkungen ... 107
Erstellen Sie eine Skalierungsrichtlinie ... 107
Registrieren eines skalierbaren Ziels ... 108
Erstellen Sie eine Skalierungsrichtlinie ... 108
Erstellen eines Alarms, der die Skalierungsrichtlinie auslöst ... 112
Beschreiben Sie Richtlinien für die Stufenskalierung ... 113
Löschen einer Stufenskalierungsrichtlinie ... 115
Tutorial: Auto Scaling zur Bewältigung eines hohen Workloads konfigurieren ... 116
Voraussetzungen ... 117
Schritt 1: Registrieren Sie Ihr skalierbares Ziel ... 118
Schritt 2: Richten Sie geplante Aktionen entsprechend Ihren Anforderungen ein ... 119
Schritt 3: Hinzufügen einer Skalierungsrichtlinie für die Zielverfolgung ... 122
Schritt 4: Nächste Schritte ... 124
Schritt 5: Bereinigen ... 125
Skalierung unterbrechen ... 128
Skalierung von Aktivitäten ... 128
Unterbrechen und Fortsetzen von Skalierungsaktivitäten ... 129
Ausgesetzte Skalierungsaktivitäten anzeigen ... 132
Wiederaufnahme der Skalierungsaktivitäten ... 133
Skalierung von Aktivitäten ... 134
Abrufen von Skalierungsaktivitäten nach skalierbarem Ziel ... 134
Einbeziehung nicht skalierter Aktivitäten ... 135
Verstehen von nicht skalierten Ursachencodes ... 137
Überwachen ... 140
AWS CloudTrail ... 141
Informationen von Application Auto Scaling in CloudTrail ... 142
Grundlegendes zu Application-Auto-Scaling-Protokolldateieinträgen ... 143
... 143
Zugehörige Ressourcen ... 144
Amazon CloudWatch ... 144
CloudWatch-Dashboards erstellen ... 144
CloudWatch-Alarm erstellen ... 146
Ressourcennutzung mit CloudWatch überwachen ... 148

Amazon EventBridge 164
Application Auto Scaling-Ereignisse 164
AWS Health Dashboard 168
Unterstütze Markierungen 170
Beispiel für eine Markierung 170
Tags für Sicherheit 171
Steuern des Zugriffs auf Tags 172
Sicherheit 174
Datenschutz 175
Identitäts- und Zugriffsverwaltung 176
Zugriffskontrolle 176
Wie Application Auto Scaling mit IAM funktioniert 177
AWS verwaltete Richtlinien 183
Service-verknüpfte Rollen 193
Beispiele für identitätsbasierte Richtlinien 199
Fehlerbehebung 211
Validierung von Berechtigungen für API-Aufrufe auf Zielressourcen 212
VPC-Endpunkte (AWS PrivateLink) 215
Erstellen eines Schnittstellen-VPC-Endpunkts 215
Erstellen Sie eine VPC-Endpunktrichtlinie 215
Ausfallsicherheit 216
Sicherheit der Infrastruktur 217
Compliance-Validierung 217
Kontingente 219
Dokumentverlauf 221
..... ccxxxiii

Was ist Application Auto Scaling?

Application Auto Scaling ist ein Webservice für Entwickler und Systemadministratoren, die eine Lösung zur automatischen Skalierung ihrer skalierbaren Ressourcen für einzelne AWS Services außerhalb von Amazon EC2 benötigen. Mit Application Auto Scaling können Sie die automatische Skalierung für die folgenden Ressourcen konfigurieren:

- AppStream 2.0-Flotten
- Aurora-Replikate
- Amazon Comprehend-Dokumentklassifizierungs- und Entitätserkennungs-Endpunkte
- DynamoDB-Tabellen und globale sekundäre Indizes
- Amazon-ECS-Dienstleistungen
- ElastiCache für Redis-Cluster (Replikationsgruppen)
- Amazon EMR-Cluster
- Amazon Keyspaces-Tabellen (für Apache Cassandra)
- Lambda-Funktion bereitgestellte Gleichzeitigkeit
- Amazon Managed Streaming for Apache Kafka (MSK) Broker-Speicher
- Amazon Neptune-Cluster
- SageMaker Endpunkt-Varianten
- SageMaker Inferenzkomponenten
- SageMaker Serverlos bereitgestellte Parallelität
- Spot-Flottenanforderungen
- Von Ihren eigenen Anwendungen und Services bereitgestellte benutzerdefinierte Ressourcen. Weitere Informationen finden Sie im Repository. GitHub

Die regionale Verfügbarkeit der oben aufgeführten AWS Dienste finden Sie in der Regionstabelle „“.

Informationen zur Skalierung Ihrer Flotte von Amazon EC2-Instances mithilfe von Auto Scaling-Gruppen finden Sie im Amazon EC2 Auto Scaling User Guide.

Merkmale von Application Auto Scaling

Application Auto Scaling ermöglicht Ihnen die automatische Skalierung Ihrer skalierbaren Ressourcen entsprechend den von Ihnen definierten Bedingungen.

- Skalierung der Zielverfolgung — Skalieren Sie eine Ressource auf der Grundlage eines Zielwerts für eine bestimmte CloudWatch Metrik.
- Schrittweise Skalierung – Skaliert eine Ressource auf der Grundlage einer Reihe von Skalierungsanpassungen, die je nach Ausmaß der Alarmüberschreitung variieren.
- Geplante Skalierung – Skalieren Sie eine Ressource nur einmalig oder nach einem wiederkehrenden Zeitplan.

Arbeiten Sie mit Application Auto Scaling

Sie können die Skalierung mit den folgenden Schnittstellen konfigurieren, abhängig von der Ressource, die Sie skalieren:

- AWS Management Console – Stellt eine Weboberfläche bereit, mit der Sie die Skalierung konfigurieren können. Wenn Sie sich für ein AWS Konto angemeldet haben, greifen Sie auf Application Auto Scaling zu, indem Sie sich bei der anmelden AWS Management Console. Öffnen Sie dann die Service-Konsole für eine der in der Einführung aufgeführten Ressourcen. Stellen Sie sicher, dass Sie die Konsole in derselben AWS-Region Weise öffnen wie die Ressource, mit der Sie arbeiten möchten.

  > **Note**
  >
  > Der Konsolenzugriff ist nicht für alle Ressourcen verfügbar. Weitere Informationen finden Sie unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

- AWS Command Line Interface (AWS CLI) — Stellt Befehle für eine Vielzahl von AWS-Services Befehlen bereit und wird unter Windows, MacOS und Linux unterstützt. Um zu beginnen, sehen Sie sich Richten Sie das ein AWS CLI an. Weitere Informationen finden Sie unter application-autoscaling in der AWS CLI Befehlsreferenz.
- AWS Tools for Windows PowerShell— Stellt Befehle für eine breite Palette von AWS Produkten für Benutzer bereit, die in der PowerShell Umgebung Skripts erstellen. Informationen zu den ersten Schritten finden Sie im AWS Tools for Windows PowerShell -Benutzerhandbuch. Weitere Informationen finden Sie in der AWS Tools for PowerShell Cmdlet-Referenz.

- AWS SDKs — Stellt sprachspezifische API-Operationen bereit und kümmert sich um viele Verbindungsdetails, wie z. B. die Berechnung von Signaturen, die Bearbeitung von Wiederholungsversuchen von Anfragen und die Behandlung von Fehlern. Weitere Informationen finden Sie unter AWS -SDKs.
- HTTPS-API – Bietet API-Aktionen auf niedriger Ebene, die Sie mithilfe von HTTPS-Anforderungen aufrufen. Weitere Informationen finden Sie unter Aktionen in der Application Auto Scaling API-Referenz.
- AWS CloudFormation— Unterstützt die Konfiguration der Skalierung mithilfe einer Vorlage. CloudFormation Weitere Informationen finden Sie unter Application-Auto-Scaling-Ressourcen mit AWS CloudFormation erstellen.

Um programmgesteuert eine Verbindung zu einem herzustellen AWS-Service, verwenden Sie einen Endpunkt. Informationen zu Endpunkten für Aufrufe von Application Auto Scaling finden Sie unter Application Auto Scaling Scaling-Endpunkte und Kontingente in den Allgemeine AWS-Referenz

Erste Schritte mit Application Auto Scaling

In diesem Thema werden die wichtigsten Konzepte erläutert, die Ihnen helfen, Application Auto Scaling kennenzulernen und zu nutzen.

Skalierbares Ziel

Eine Entität, die Sie erstellen, um die Ressource anzugeben, die Sie skalieren möchten. Jedes skalierbare Ziel wird eindeutig durch einen Service-Namespace, eine Ressourcen-ID und eine skalierbare Dimension identifiziert, die eine Kapazitätsdimension des zugrunde liegenden Dienstes darstellt. Ein Amazon ECS-Service unterstützt beispielsweise die automatische Skalierung der Anzahl seiner Aufgaben, eine DynamoDB-Tabelle unterstützt das Auto Scaling der Lese- und Schreibkapazität der Tabelle und ihrer globalen sekundären Indizes, und ein Aurora-Cluster unterstützt die Skalierung der Anzahl seiner Replikate.

> **Tip**
>
> Jedes skalierbare Ziel hat auch eine minimale und maximale Kapazität. Die Skalierungsrichtlinien gehen nie über oder unter den Minimal-/Maximalbereich.
> Sie können Out-of-Band-Änderungen direkt an der zugrunde liegenden Ressource vornehmen, die außerhalb dieses Bereichs liegen und von denen Application Auto Scaling nichts weiß. Wenn jedoch eine Skalierungsrichtlinie aufgerufen oder die `RegisterScalableTarget`-API aufgerufen wird, ruft Application Auto Scaling die aktuelle Kapazität ab und vergleicht sie mit der minimalen und maximalen Kapazität. Liegt sie außerhalb des Minimal- und Maximalbereichs, wird die Kapazität so aktualisiert, dass sie dem festgelegten Minimum und Maximum entspricht.

Skalieren in

Wenn Application Auto Scaling die Kapazität für ein skalierbares Ziel automatisch verringert, skaliert das skalierbare Ziel nach innen. Wenn Skalierungsrichtlinien festgelegt sind, kann das skalierbare Ziel nicht unter seiner Mindestkapazität abskaliert werden.

Horizontale Skalierung

Wenn Application Auto Scaling automatisch die Kapazität für ein skalierbares Ziel erhöht, skaliert das skalierbare Ziel nach aussen. Wenn Skalierungsrichtlinien festgelegt sind, kann das skalierbare Ziel nicht über seine maximale Kapazität aufskaliert werden.

Skalierungsrichtlinie

Eine Skalierungsrichtlinie weist Application Auto Scaling an, eine bestimmte CloudWatch-Metrik zu verfolgen. Anschließend wird festgelegt, welche Skalierungsmaßnahme zu ergreifen ist, wenn die Metrik einen bestimmten Schwellenwert über- oder unterschreitet. Sie könnten beispielsweise eine Skalierung vornehmen, wenn die CPU-Auslastung in Ihrem Cluster zu steigen beginnt, und eine Skalierung vornehmen, wenn sie wieder sinkt.

Die Metriken, die für das Auto Scaling verwendet werden, werden vom Zieldienst veröffentlicht, aber Sie können auch Ihre eigene Metrik in CloudWatch veröffentlichen und sie dann mit einer Skalierungsrichtlinie verwenden.

Ein Abkühlungszeitraum zwischen den Skalierungsaktivitäten ermöglicht es der Ressource, sich zu stabilisieren, bevor eine weitere Skalierungsaktivität beginnt. Application Auto Scaling wertet die Metriken während der Abkühlungsphase weiter aus. Nach Ablauf des Abkühlungszeitraums leitet die Skalierungsrichtlinie bei Bedarf eine weitere Skalierungsaktivität ein. Wenn während des Abkühlungszeitraums aufgrund des aktuellen Metrikwerts eine größere Skalierung erforderlich ist, nimmt die Skalierungsrichtlinie sofort eine Skalierung vor.

Geplante Aktion

Geplante Aktionen skalieren Ressourcen automatisch zu einem bestimmten Datum und einer bestimmten Uhrzeit. Sie funktionieren, indem sie die minimale und maximale Kapazität für ein skalierbares Ziel ändern, und können daher verwendet werden, um nach einem Zeitplan zu skalieren, indem die minimale Kapazität hoch oder die maximale Kapazität niedrig eingestellt wird. Sie können zum Beispiel geplante Aktionen verwenden, um eine Anwendung zu skalieren, die an Wochenenden keine Ressourcen verbraucht, indem Sie die Kapazität am Freitag verringern und am darauffolgenden Montag erhöhen.

Sie können auch geplante Aktionen verwenden, um die minimalen und maximalen Werte im Laufe der Zeit zu optimieren, um sich an Situationen anzupassen, in denen ein höherer Datenverkehr als normal erwartet wird, z. B. bei Marketingkampagnen oder saisonalen Schwankungen. Auf diese Weise können Sie die Leistung in Zeiten verbessern, in denen Sie aufgrund der zunehmenden Nutzung eine höhere Skalierung vornehmen müssen, und die Kosten in Zeiten senken, in denen Sie weniger Ressourcen benötigen.

Weitere Informationen

AWS -Services, die Sie mit Application Auto Scaling verwenden können — In diesem Abschnitt werden die Dienste vorgestellt, die Sie skalieren können, und Sie können das Auto Scaling einrichten, indem Sie ein skalierbares Ziel registrieren. Außerdem wird jede der mit dem IAM-Dienst verknüpften Rollen beschrieben, die Application Auto Scaling für den Zugriff auf Ressourcen im Zieldienst erstellt.

Skalierungsrichtlinien für die Ziel-Nachverfolgung — Eine der wichtigsten Funktionen von Application Auto Scaling ist die Nachverfolgung von Skalierungsrichtlinien für das Ziel. Erfahren Sie, wie Zielverfolgungsrichtlinien automatisch die gewünschte Kapazität anpassen, um die Auslastung auf der Grundlage Ihrer konfigurierten Metrik- und Zielwerte konstant zu halten. So können Sie beispielsweise die Zielverfolgung so konfigurieren, dass die durchschnittliche CPU-Auslastung für Ihre Spot-Flotte bei 50 Prozent bleibt. Application Auto Scaling startet oder beendet dann EC2-Instanzen nach Bedarf, um die aggregierte CPU-Auslastung über alle Server hinweg auf 50 Prozent zu halten.

AWS -Services, die Sie mit Application Auto Scaling verwenden können

Application Auto Scaling lässt sich in andere - AWS Services integrieren, sodass Sie Skalierungsfunktionen hinzufügen können, um die Anforderungen Ihrer Anwendung zu erfüllen. Die automatische Skalierung ist eine optionale Funktion des Services, die standardmäßig in fast allen Fällen deaktiviert ist.

In der folgenden Tabelle sind die AWS Services aufgeführt, die Sie mit Application Auto Scaling verwenden können, einschließlich Informationen zu unterstützten Methoden für die Konfiguration von Auto Scaling. Sie können Application Auto Scaling auch mit benutzerdefinierten Ressourcen verwenden.

Konsolenzugriff - Sie können einen kompatiblen AWS Dienst zum Starten der automatischen Skalierung konfigurieren, indem Sie eine Skalierungsrichtlinie in der Konsole des Zieldienstes konfigurieren.

CLI-Zugriff - Sie können einen kompatiblen AWS -Dienst so konfigurieren, dass die automatische Skalierung über die AWS CLI.

SDK-Zugriff – Sie können einen kompatiblen AWS Service konfigurieren, um die automatische Skalierung mit den AWS SDKs zu starten.

CloudFormation Zugriff – Sie können einen kompatiblen AWS Service so konfigurieren, dass die automatische Skalierung mithilfe einer - AWS CloudFormation Stack-Vorlage gestartet wird. Weitere Informationen finden Sie unter Application-Auto-Scaling-Ressourcen mit AWS CloudFormation erstellen.

AWS - Service	Konsolenzugriff1	CLI-Zugang	SDK-Zugang	CloudFormation -Zugriff
AppStream 2.0	Ja	Ja	Ja	Ja

AWS - Service	Konsolenzugriff[1]	CLI-Zugang	SDK-Zugang	CloudFormation -Zugriff
Aurora	Ja	Ja	Ja	Ja
Amazon Comprehend	Nein	Ja	Ja	Ja
Amazon DynamoDB	Ja	Ja	Ja	Ja
Amazon ECS	Ja	Ja	Ja	Ja
Amazon ElastiCache	Ja	Ja	Ja	Ja
Amazon EMR	Ja	Ja	Ja	Ja
Amazon Keyspaces	Ja	Ja	Ja	Ja
Lambda	Nein	Ja	Ja	Ja

AWS - Service	Konsolenzugriff1	CLI-Zugang	SDK-Zugang	CloudFormation -Zugriff
Amazon MSK	Ja	Ja	Ja	Ja
Amazon Neptune	Nein	Ja	Ja	Ja
SageMaker	Ja	Ja	Ja	Ja
Spot Fleet	Ja	Ja	Ja	Ja
Benutzerdefinierte Ressourcen	Nein	Ja	Ja	Ja

1 Konsolenzugriff für die Konfiguration von Skalierungsrichtlinien. Die meisten -Services unterstützen die Konfiguration der geplanten Skalierung über die Konsole nicht. Derzeit bieten nur Amazon AppStream 2.0 ElastiCacheund Spot-Flotte Konsolenzugriff für geplante Skalierung.

Amazon AppStream 2.0 und Application Auto Scaling

Sie können AppStream 2.0-Flotten mithilfe von Zielverfolgungs-Skalierungsrichtlinien, Stufenskalierungsrichtlinien und geplanter Skalierung skalieren.

Verwenden Sie die folgenden Informationen, um AppStream 2.0 mit Application Auto Scaling zu integrieren.

Serviceverknüpfte Rolle für AppStream 2.0 erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn AppStream 2.0-Ressourcen als skalierbare Ziele mit Application Auto Scaling registriert werden. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_AppStreamFleet`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `appstream.application-autoscaling.amazonaws.com`

Registrierung von AppStream 2.0-Flotten als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für eine AppStream 2.0-Flotte erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Wenn Sie Auto Scaling mit der AppStream 2.0-Konsole konfigurieren, registriert AppStream 2.0 automatisch ein skalierbares Ziel für Sie.

Wenn Sie Auto Scaling mit der AWS CLI oder einem der AWS SDKs konfigurieren möchten, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den register-scalable-target Befehl für eine AppStream 2.0-Flotte auf. Im folgenden Beispiel wird die gewünschte Kapazität einer Flotte mit dem Namen `sample-fleet` registriert,

mit einer Mindestkapazität von einer Flotten-Instance und einer Höchstkapazität von fünf Flotten-Instances.

```
aws application-autoscaling register-scalable-target \
   --service-namespace appstream \
   --scalable-dimension appstream:fleet:DesiredCapacity \
   --resource-id fleet/sample-fleet \
   --min-capacity 1 \
   --max-capacity 5
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation zusätzliche nützliche Informationen zur Skalierung Ihrer AppStream 2.0-Ressourcen:

Fleet Auto Scaling für AppStream 2.0 im Amazon AppStream 2.0 Administration Guide

Amazon Aurora und Application Auto Scaling

Sie können Aurora-DB-Cluster mithilfe von Zielverfolgungs-Skalierungsrichtlinien, Stufenskalierungsrichtlinien und geplanter Skalierung skalieren.

Verwenden Sie die folgenden Informationen, um Aurora mit Application Auto Scaling zu integrieren.

Service-verknüpfte Rolle für Aurora erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Sie Aurora-Ressourcen als skalierbare Ziele mit Application Auto Scaling registrieren. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_RDSCluster`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `rds.application-autoscaling.amazonaws.com`

Registrierung von Aurora DB-Clustern als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für einen Aurora-Cluster erstellen können. Ein skalierbares Ziel ist eine Ressource, die Application Auto Scaling aus- und einskalieren kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Wenn Sie die automatische Skalierung über die Aurora-Konsole konfigurieren, registriert Aurora automatisch ein skalierbares Ziel für Sie.

Wenn Sie Auto Scaling mit der AWS CLI oder einem der AWS SDKs konfigurieren möchten, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den Befehl register-scalable-target für einen Aurora-Cluster auf. Im folgenden Beispiel wird die Anzahl der Aurora-Replikate in einem Cluster mit dem Namen `my-db-cluster` registriert, mit einer Mindestkapazität von einem Aurora-Replikat und einer Höchstkapazität von acht Aurora-Replikaten.

```
aws application-autoscaling register-scalable-target \
  --service-namespace rds \
  --scalable-dimension rds:cluster:ReadReplicaCount \
  --resource-id cluster:my-db-cluster \
  --min-capacity 1 \
  --max-capacity 8
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer Aurora-Ressourcen:

Verwendung von Amazon Aurora Auto Scaling mit Aurora-Replikaten im Amazon RDS-Benutzerhandbuch

Amazon Comprehend und Application Auto Scaling

Sie können Amazon Comprehend Dokumentenklassifizierung und Entity Recognizer Endpunkte mit Hilfe von Zielverfolgungs-Skalierungsrichtlinien und geplanter Skalierung skalieren.

Die folgenden Informationen helfen Ihnen bei der Integration von Amazon Comprehend mit Application Auto Scaling.

Service-verknüpfte Rolle für Amazon Comprehend erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Sie Amazon Comprehend-Ressourcen als skalierbare Ziele mit Application Auto Scaling registrieren. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_ComprehendEndpoint`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `comprehend.application-autoscaling.amazonaws.com`

Registrierung von Amazon Comprehend Ressourcen als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für einen Amazon Comprehend Document Classification oder Entity Recognizer Endpunkt erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Um Auto Scaling mit der AWS CLI oder einem der AWS SDKs zu konfigurieren, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den Befehl register-scalable-target für einen Endpunkt der Dokumentenklassifizierung auf. Das folgende Beispiel registriert die gewünschte Anzahl von Inferenzeinheiten, die vom Modell für einen Dokumentenklassifikator-Endpunkt verwendet werden sollen, unter Verwendung des ARN des Endpunkts, mit einer Mindestkapazität von einer Inferenzeinheit und einer Höchstkapazität von drei Inferenzeinheiten.

```
aws application-autoscaling register-scalable-target \
  --service-namespace comprehend \
  --scalable-dimension comprehend:document-classifier-endpoint:DesiredInferenceUnits
 \
  --resource-id arn:aws:comprehend:us-west-2:123456789012:document-classifier-
endpoint/EXAMPLE \
  --min-capacity 1 \
  --max-capacity 3
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Rufen Sie den Befehl `register-scalable-target` für einen Entity Recognizer Endpunkt auf. Das folgende Beispiel registriert die gewünschte Anzahl von Inferenzeinheiten, die vom Modell für einen Entity Recognizer unter Verwendung der ARN des Endpunkts verwendet werden sollen, mit einer Mindestkapazität von einer Inferenzeinheit und einer Höchstkapazität von drei Inferenzeinheiten.

```
aws application-autoscaling register-scalable-target \
  --service-namespace comprehend \
  --scalable-dimension comprehend:entity-recognizer-endpoint:DesiredInferenceUnits \
  --resource-id arn:aws:comprehend:us-west-2:123456789012:entity-recognizer-
endpoint/EXAMPLE \
  --min-capacity 1 \
  --max-capacity 3
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer Amazon Comprehend-Ressourcen:

Automatische Skalierung mit Endpunkten im Amazon Comprehend Developer Guide

Amazon DynamoDB und Application Auto Scaling

Sie können DynamoDB-Tabellen und globale sekundäre Indizes mithilfe von Zielverfolgungs-Skalierungsrichtlinien und geplanter Skalierung skalieren.

Die folgenden Informationen helfen Ihnen bei der Integration von DynamoDB mit Application Auto Scaling.

Service-verknüpfte Rolle für DynamoDB erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn DynamoDB-Ressourcen als skalierbare Ziele mit Application Auto Scaling registriert werden. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_DynamoDBTable`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `dynamodb.application-autoscaling.amazonaws.com`

Registrieren von DynamoDB-Ressourcen als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für eine DynamoDB-Tabelle oder einen globalen sekundären Index erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Wenn Sie die automatische Skalierung über die DynamoDB-Konsole konfigurieren, registriert DynamoDB automatisch ein skalierbares Ziel für Sie.

Wenn Sie Auto Scaling mit der AWS CLI oder einem der AWS SDKs konfigurieren möchten, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den register-scalable-target Befehl für die Schreibkapazität einer Tabelle auf. Das folgende Beispiel registriert die bereitgestellte Schreibkapazität einer Tabelle mit dem Namen `my-table`, mit einer Mindestkapazität von fünf Schreibkapazitätseinheiten und einer Höchstkapazität von 10 Schreibkapazitätseinheiten.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace dynamodb \
    --scalable-dimension dynamodb:table:WriteCapacityUnits \
    --resource-id table/my-table \
    --min-capacity 5 \
    --max-capacity 10
  ```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

  Rufen Sie den register-scalable-target Befehl für die Lesekapazität einer Tabelle auf. Das folgende Beispiel registriert die bereitgestellte Lesekapazität einer Tabelle mit dem Namen `my-table`,

mit einer Mindestkapazität von fünf Lesekapazitätseinheiten und einer Höchstkapazität von 10 Leseeinheiten.

```
aws application-autoscaling register-scalable-target \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits \
  --resource-id table/my-table \
  --min-capacity 5 \
  --max-capacity 10
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Rufen Sie den register-scalable-target Befehl für die Schreibkapazität eines globalen sekundären Index auf. Das folgende Beispiel registriert die bereitgestellte Schreibkapazität eines globalen sekundären Index mit dem Namen `my-table-index`, mit einer Mindestkapazität von fünf Schreibkapazitätseinheiten und einer Höchstkapazität von 10 Schreibkapazitätseinheiten.

```
aws application-autoscaling register-scalable-target \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:index:WriteCapacityUnits \
  --resource-id table/my-table/index/my-table-index \
  --min-capacity 5 \
  --max-capacity 10
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Rufen Sie den register-scalable-target Befehl für die Lesekapazität eines globalen sekundären Index auf. Das folgende Beispiel registriert die bereitgestellte Lesekapazität eines global

sekundären Index mit dem Namen `my-table-index`, mit einer Mindestkapazität von fünf Lesekapazitätseinheiten und einer Höchstkapazität von 10 Lesekapazitätseinheiten.

```
aws application-autoscaling register-scalable-target \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:index:ReadCapacityUnits \
  --resource-id table/my-table/index/my-table-index \
  --min-capacity 5 \
  --max-capacity 10
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation zusätzliche nützliche Informationen zum Skalieren Ihrer DynamoDB-Ressourcen:

- Verwaltung der Durchsatzkapazität mit DynamoDB Auto Scaling im Amazon DynamoDB Developer Leitfaden
- Auswerten der Auto-Scaling-Einstellungen Ihrer Tabelle im Amazon-DynamoDB-Entwicklerhandbuch
- So konfigurieren Sie AWS CloudFormation Auto Scaling für DynamoDB-Tabellen und -Indizes mit im AWS Blog

Sie finden auch ein Tutorial für geplante Skalierung in Tutorial: Erste Schritte mit der geplanten Skalierung mit AWS CLI. In diesem Tutorial lernen Sie die grundlegenden Schritte zur Konfiguration der Skalierung, damit Ihre DynamoDB-Tabelle zu geplanten Zeiten skaliert wird.

Amazon ECS und Application Auto Scaling

Sie können ECS-Services mithilfe von Zielverfolgungs-Skalierungsrichtlinien, Stufenskalierungsrichtlinien und geplanter Skalierung skalieren.

Die folgenden Informationen helfen Ihnen bei der Integration von Amazon ECS mit Application Auto Scaling.

Serviceverknüpfte Rolle für Amazon ECS erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Sie Amazon-ECS-Ressourcen als skalierbare Ziele mit Application Auto Scaling registrieren. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_ECSService`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `ecs.application-autoscaling.amazonaws.com`

Registrierung von ECS-Diensten als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für einen Amazon ECS-Service erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Wenn Sie die automatische Skalierung über die Amazon ECS-Konsole konfigurieren, dann registriert Amazon ECS automatisch ein skalierbares Ziel für Sie.

Wenn Sie Auto Scaling mit der AWS CLI oder einem der AWS SDKs konfigurieren möchten, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den Befehl register-scalable-target für einen Amazon ECS-Service auf. Das folgende Beispiel registriert ein skalierbares Ziel für einen Service namens `sample-app-service`, der auf dem `default` Cluster läuft, mit einer minimalen Taskanzahl von einem Task und einer maximalen Taskanzahl von 10 Tasks.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace ecs \
    --scalable-dimension ecs:service:DesiredCount \
    --resource-id service/default/sample-app-service \
    --min-capacity 1 \
    --max-capacity 10
  ```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
  target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer Amazon-ECS-Ressourcen:

- Service Auto Scaling im Amazon Elastic Container Service-Entwicklerhandbuch
- Konfigurieren von Service Auto Scaling im Leitfaden zu bewährten Methoden für Amazon Elastic Container Service

> **Note**
>
> Anweisungen zum Aussetzen von Aufskalierungsprozessen während Amazon-ECS-Bereitstellungen finden Sie in der folgenden Dokumentation: Service Auto Scaling und Bereitstellungen im Amazon Elastic Container Service-Entwicklerhandbuch

ElastiCache für Redis und Application Auto Scaling

Sie können ElastiCache für Redis-Replikationsgruppen mithilfe von Zielverfolgungs-Skalierungsrichtlinien und geplanter Skalierung skalieren.

Verwenden Sie die folgenden Informationen, um die Integration ElastiCache mit Application Auto Scaling zu erleichtern.

Dienstverknüpfte Rolle für ElastiCache erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Sie ElastiCache Ressourcen als skalierbare Ziele mit Application Auto Scaling registrieren. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `elasticache.application-autoscaling.amazonaws.com`

Registrierung ElastiCache für Redis-Replikationsgruppen als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für eine ElastiCache Replikationsgruppe erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Wenn Sie Auto Scaling über die ElastiCache Konsole konfigurieren, registriert ElastiCache automatisch ein skalierbares Ziel für Sie.

Wenn Sie Auto Scaling mit der AWS CLI oder einem der AWS SDKs konfigurieren möchten, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den register-scalable-target Befehl für eine ElastiCache Replikationsgruppe auf. Im folgenden Beispiel wird die gewünschte Anzahl von Knotengruppen für eine Replikationsgruppe mit dem Namen `mycluster` registriert, mit einer Mindestkapazität von einem und einer Höchstkapazität von fünf.

  ```
  aws application-autoscaling register-scalable-target \
     --service-namespace elasticache \
     --scalable-dimension elasticache:replication-group:NodeGroups \
     --resource-id replication-group/mycluster \
     --min-capacity 1 \
     --max-capacity 5
  ```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
  target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

  Das folgende Beispiel registriert die gewünschte Anzahl von Replikaten pro Knotengruppe für eine Replikationsgruppe mit dem Namen 1, mit einer Mindestkapazität von `mycluster` und einer Höchstkapazität von 5.

```
aws application-autoscaling register-scalable-target \
   --service-namespace elasticache \
   --scalable-dimension elasticache:replication-group:Replicas \
   --resource-id replication-group/mycluster \
   --min-capacity 1 \
   --max-capacity 5
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer ElastiCache Ressourcen:

Auto Scaling ElastiCache für Redis-Cluster im Amazon ElastiCache for Redis-Benutzerhandbuch

Amazon Keyspaces (für Apache Cassandra) und Application Auto Scaling

Sie können Amazon Keyspaces-Tabellen mithilfe von Zielverfolgungs-Skalierungsrichtlinien und geplanter Skalierung skalieren.

Die folgenden Informationen helfen Ihnen bei der Integration von Amazon Keyspaces mit Application Auto Scaling.

Service-verknüpfte Rolle für Amazon Keyspaces erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Sie Amazon Keyspaces-Ressourcen als skalierbare Ziele mit Application Auto Scaling registrieren. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_CassandraTable`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `cassandra.application-autoscaling.amazonaws.com`

Registrierung von Amazon Keyspaces-Tabellen als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für eine Amazon Keyspaces-Tabelle erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Wenn Sie die automatische Skalierung über die Amazon Keyspaces-Konsole konfigurieren, dann registriert Amazon Keyspaces automatisch ein skalierbares Ziel für Sie.

Wenn Sie Auto Scaling mit der AWS CLI oder einem der AWS SDKs konfigurieren möchten, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den register-scalable-target Befehl für eine Amazon Keyspaces-Tabelle auf. Das folgende Beispiel registriert die bereitgestellte Schreibkapazität einer Tabelle mit dem

Namen `mytable`, mit einer Mindestkapazität von fünf Schreibkapazitätseinheiten und einer Höchstkapazität von 10 Schreibkapazitätseinheiten.

```
aws application-autoscaling register-scalable-target \
  --service-namespace cassandra \
  --scalable-dimension cassandra:table:WriteCapacityUnits \
  --resource-id keyspace/mykeyspace/table/mytable \
  --min-capacity 5 \
  --max-capacity 10
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Das folgende Beispiel registriert die bereitgestellte Lesekapazität einer Tabelle mit dem Namen `mytable`, mit einer Mindestkapazität von fünf Lesekapazitätseinheiten und einer Höchstkapazität von 10 Lesekapazitätseinheiten.

```
aws application-autoscaling register-scalable-target \
  --service-namespace cassandra \
  --scalable-dimension cassandra:table:ReadCapacityUnits \
  --resource-id keyspace/mykeyspace/table/mytable \
  --min-capacity 5 \
  --max-capacity 10
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer Amazon Keyspaces-Ressourcen:

Verwalten der Durchsatzkapazität mit Amazon Keyspaces Auto Scaling im Entwicklerhandbuch für Amazon Keyspaces (für Apache Cassandra)

AWS Lambda und Application Auto Scaling

Sie können AWS Lambda bereitgestellte Gleichzeitigkeit mithilfe von Zielverfolgungs-Skalierungsrichtlinien und geplanter Skalierung skalieren.

Die folgenden Informationen helfen Ihnen bei der Integration von Lambda mit Application Auto Scaling.

Dienstverknüpfte Rolle für Lambda erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Lambda-Ressourcen als skalierbare Ziele mit Application Auto Scaling registriert werden. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `lambda.application-autoscaling.amazonaws.com`

Registrieren von Lambda-Funktionen als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für eine Lambda-Funktion erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Um Auto Scaling mit der AWS CLI oder einem der AWS SDKs zu konfigurieren, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den register-scalable-target-Befehl für eine Lambda-Funktion auf. Im folgenden Beispiel wird die bereitgestellte Gleichzeitigkeit für einen Alias mit dem Namen `BLUE` für eine Funktion mit dem Namen `my-function` mit einer Mindestkapazität von 0 und einer Höchstkapazität von 100 registriert.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace lambda \
    --scalable-dimension lambda:function:ProvisionedConcurrency \
    --resource-id function:my-function:BLUE \
    --min-capacity 0 \
    --max-capacity 100
  ```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer Lambda-Funktionen:

- Konfigurieren der bereitgestellten Gleichzeitigkeit im -AWS Lambda Entwicklerhandbuch
- Planen von Lambda Provisioned Concurrency für wiederkehrende Spitzenauslastung im AWS Blog

Amazon Managed Streaming for Apache Kafka (MSK) und Application Auto Scaling

Sie können den Amazon MSK-Clusterspeicher mithilfe von Zielverfolgungs-Skalierungsrichtlinien skalieren. Gibt an, ob die Herunterskalierung durch die Richtlinie für die Ziel-Nachverfolgung deaktiviert ist.

Die folgenden Informationen helfen Ihnen bei der Integration von Amazon MSK mit Application Auto Scaling.

Service-gebundene Rolle für Amazon MSK erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Sie Amazon-MSK-Ressourcen als skalierbare Ziele mit Application Auto Scaling registrieren. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_KafkaCluster`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `kafka.application-autoscaling.amazonaws.com`

Registrierung von Amazon MSK-Cluster-Speicher als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie eine Skalierungsrichtlinie für die Größe des Speichervolumens pro Broker eines Amazon MSK Clusters erstellen können. Ein skalierbares Ziel ist eine Ressource, die Application Auto Scaling aufskalieren oder abskalieren kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Wenn Sie die automatische Skalierung über die Amazon MSK-Konsole konfigurieren, dann registriert Amazon MSK automatisch ein skalierbares Ziel für Sie.

Wenn Sie Auto Scaling mit der AWS CLI oder einem der AWS SDKs konfigurieren möchten, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den Befehl register-scalable-target für einen Amazon MSK-Cluster auf. Das folgende Beispiel registriert die Größe des Speichervolumens pro Broker eines Amazon MSK Clusters mit einer Mindestkapazität von 100 GiB und einer Höchstkapazität von 800 GiB.

  ```
  aws application-autoscaling register-scalable-target \
     --service-namespace kafka \
     --scalable-dimension kafka:broker-storage:VolumeSize \
     --resource-id arn:aws:kafka:us-east-1:123456789012:cluster/demo-
  cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5 \
     --min-capacity 100 \
     --max-capacity 800
  ```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
  target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

> **Note**
>
> Wenn ein Amazon MSK-Cluster das skalierbare Ziel ist, ist scale in deaktiviert und kann nicht aktiviert werden.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer Amazon-MSK-Ressourcen:

Automatische Skalierung im Entwicklerhandbuch für Amazon Managed Streaming für Apache Kafka

Amazon Neptune und Application Auto Scaling

Sie können Neptune-Funktionen mithilfe von Zielverfolgungs-Skalierungsrichtlinien und geplanter Skalierung skalieren.

Die folgenden Informationen helfen Ihnen bei der Integration von Neptune mit Application Auto Scaling.

Serviceverknüpfte Rolle für Neptune erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Neptune-Ressourcen als skalierbare Ziele mit Application Auto Scaling registriert werden. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_NeptuneCluster`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen

autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `neptune.application-autoscaling.amazonaws.com`

Registrierung von Neptune-Clustern als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für einen Neptune-Cluster erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Um Auto Scaling mit der AWS CLI oder einem der AWS SDKs zu konfigurieren, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den register-scalable-target Befehl für einen Neptune-Cluster auf. Im folgenden Beispiel wird die gewünschte Kapazität einer Clusters mit dem Namen `mycluster` registriert, mit einer Mindestkapazität von einer Flotten-Instance und einer Höchstkapazität von acht Flotten-Instances.

```
aws application-autoscaling register-scalable-target \
  --service-namespace neptune \
  --scalable-dimension neptune:cluster:ReadReplicaCount \
  --resource-id cluster:mycluster \
  --min-capacity 1 \
  --max-capacity 8
```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer Neptune-Ressourcen:

Automatisches Skalieren der Anzahl der Replikate in einem Amazon Neptune-DB-Cluster im Benutzerhandbuch für Neptune

Amazon SageMaker und Application Auto Scaling

Sie können SageMaker Endpunktvarianten, bereitgestellte Parallelität für Serverless-Endpunkte und Inferenzkomponenten mithilfe von Zielverfolgungs-Skalierungsrichtlinien, Stufenskalierungsrichtlinien und geplanter Skalierung skalieren.

Verwenden Sie die folgenden Informationen, um die Integration SageMaker mit Application Auto Scaling zu erleichtern.

Dienstverknüpfte Rolle für SageMaker erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Sie SageMaker Ressourcen als skalierbare Ziele mit Application Auto Scaling registrieren. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `sagemaker.application-autoscaling.amazonaws.com`

Registrieren von SageMaker Endpunktvarianten als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für ein SageMaker Modell (Variante) erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Wenn Sie Auto Scaling über die SageMaker Konsole konfigurieren, registriert SageMaker automatisch ein skalierbares Ziel für Sie.

Wenn Sie Auto Scaling mit der AWS CLI oder einem der AWS SDKs konfigurieren möchten, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den register-scalable-target Befehl für eine Produktvariante auf. Das folgende Beispiel registriert die gewünschte Anzahl von Instances für eine Produktvariante namens `my-variant`, die auf dem Endpunkt `my-endpoint` ausgeführt wird, mit einer Mindestkapazität von einer Instance und einer Höchstkapazität von acht Instances.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:variant:DesiredInstanceCount \
    --resource-id endpoint/my-endpoint/variant/my-variant \
    --min-capacity 1 \
    --max-capacity 8
  ```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
  target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Registrieren der bereitgestellten Gleichzeitigkeit von Serverless-Endpunkten als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert auch ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für die bereitgestellte Gleichzeitigkeit von Serverless-Endpunkten erstellen können.

Wenn Sie Auto Scaling über die SageMaker Konsole konfigurieren, registriert SageMaker automatisch ein skalierbares Ziel für Sie.

Verwenden Sie andernfalls eine der folgenden Methoden, um das skalierbare Ziel zu registrieren:

- AWS CLI:

  Rufen Sie den register-scalable-target Befehl für eine Produktvariante auf. Das folgende Beispiel registriert die bereitgestellte Gleichzeitigkeit für eine Produktvariante namens `my-variant`, die auf dem Endpunkt `my-endpoint` ausgeführt wird, mit einer Mindestkapazität von eins und einer Höchstkapazität von zehn.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:variant:DesiredProvisionedConcurrency \
    --resource-id endpoint/my-endpoint/variant/my-variant \
    --min-capacity 1 \
    --max-capacity 10
  ```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Registrieren von Inferenzkomponenten als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für Inferenzkomponenten erstellen können.

- AWS CLI:

  Rufen Sie den register-scalable-target Befehl für eine Inferenzkomponente auf. Im folgenden Beispiel wird die gewünschte Kopienanzahl für eine Inferenzkomponente namens `my-inference-component` registriert, mit einer Mindestkapazität von null Kopien und einer Höchstkapazität von drei Kopien.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:inference-component:DesiredCopyCount \
    --resource-id inference-component/my-inference-component \
    --min-capacity 0 \
    --max-capacity 3
  ```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie weitere nützliche Informationen zum Skalieren Ihrer SageMaker Ressourcen im Amazon- SageMaker Entwicklerhandbuch:

- Automatische Skalierung von Amazon- SageMaker Modellen
- Automatische Skalierung der bereitgestellten Gleichzeitigkeit für einen Serverless-Endpunkt
- Festlegen von Auto-Scaling-Richtlinien für Endpunktbereitstellungen mit mehreren Modellen
- Auto Skalieren eines asynchronen Endpunkts

> **Note**
>
> Im Jahr 2023 wurden neue Inferenzfunktionen SageMaker eingeführt, die auf Echtzeit-Inferenzendpunkten basieren. Sie erstellen einen SageMaker Endpunkt mit einer Endpunktkonfiguration, die den Instance-Typ und die anfängliche Instance-Anzahl für den Endpunkt definiert. Erstellen Sie dann eine Inferenzkomponente, bei der es sich um ein SageMaker Hosting-Objekt handelt, mit dem Sie ein Modell auf einem Endpunkt bereitstellen können. Informationen zur Skalierung von Inferenzkomponenten finden Sie unter Amazon SageMaker fügt neue Inferenzfunktionen hinzu, um die Bereitstellungskosten und Latenz des Grundlagenmodells zu senken, und Reduzieren Sie die Kosten für die Modellbereitstellung um durchschnittlich 50 % mithilfe der neuesten Funktionen von Amazon SageMaker im - AWS Blog.

Amazon EC2 Spot-Flotte und Application Auto Scaling

Sie können Spot Fleets mithilfe von Zielverfolgungs-Skalierungsrichtlinien, Stufenskalierungsrichtlinien und geplanter Skalierung skalieren.

Die folgenden Informationen helfen Ihnen bei der Integration von Spot Fleet mit Application Auto Scaling.

Serviceverknüpfte Rolle für Spot Fleet erstellt

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Sie Spot-Flottenressourcen als skalierbare Ziele mit Application Auto Scaling registrieren. Mit dieser Rolle

kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `ec2.application-autoscaling.amazonaws.com`

Registrierung von Spot Fleets als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling erfordert ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für ein Spot Fleet erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Wenn Sie die automatische Skalierung über die Spot Fleet-Konsole konfigurieren, registriert Spot Fleet automatisch ein skalierbares Ziel für Sie.

Wenn Sie Auto Scaling mit der AWS CLI oder einem der AWS SDKs konfigurieren möchten, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den Befehl register-scalable-target für eine Spot-Flotte auf. Das folgende Beispiel registriert die Zielkapazität einer Spot-Flotte anhand ihrer Anfrage-ID, mit einer Mindestkapazität von zwei Instances und einer Höchstkapazität von 10 Instances.

```
aws application-autoscaling register-scalable-target \
  --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --min-capacity 2 \
```

```
  --max-capacity 10
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer Spot-Flotte:

Automatische Skalierung für Spot Fleet im Amazon EC2-Benutzerhandbuch

Benutzerdefinierte Ressourcen und Application Auto Scaling

Sie können benutzerdefinierte Ressourcen mithilfe von Zielverfolgungs-Skalierungsrichtlinien, Stufenskalierungsrichtlinien und geplanter Skalierung skalieren.

Die folgenden Informationen helfen Ihnen bei der Integration benutzerdefinierter Ressourcen in Application Auto Scaling.

Für benutzerdefinierte Ressourcen erstellte serviceverknüpfte Rolle

Die folgende serviceverknüpfte Rolle wird automatisch in Ihrem erstellt AWS-Konto , wenn Sie benutzerdefinierte Ressourcen als skalierbare Ziele mit Application Auto Scaling registrieren. Mit dieser Rolle kann Application Auto Scaling unterstützte Operationen innerhalb Ihres Kontos durchführen. Weitere Informationen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_CustomResource`

Von der dienstgebundenen Rolle verwendeter Hauptdienst

Die im vorigen Abschnitt beschriebene dienstgebundene Rolle kann nur vom Hauptdienst übernommen werden, der durch die für die Rolle definierten vertrauenswürdigen Beziehungen autorisiert ist. Die von Application Auto Scaling verwendete dienstgebundene Rolle gewährt Zugriff auf den folgenden Hauptdienst:

- `custom-resource.application-autoscaling.amazonaws.com`

Registrierung von benutzerdefinierten Ressourcen als skalierbare Ziele mit Application Auto Scaling

Application Auto Scaling benötigt ein skalierbares Ziel, bevor Sie Skalierungsrichtlinien oder geplante Aktionen für eine benutzerdefinierte Ressource erstellen können. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann. Skalierbare Ziele werden eindeutig durch die Kombination von Ressourcen-ID, skalierbarer Dimension und Namespace identifiziert.

Um Auto Scaling mit der AWS CLI oder einem der AWS SDKs zu konfigurieren, können Sie die folgenden Optionen verwenden:

- AWS CLI:

  Rufen Sie den register-scalable-target-Befehl für eine benutzerdefinierte Ressource auf. Im folgenden Beispiel wird eine benutzerdefinierte Ressource als skalierbares Ziel registriert, mit einer gewünschten Mindestanzahl von einer Kapazitätseinheit und einer gewünschten Höchstanzahl von 10 Kapazitätseinheiten. Die Datei `custom-resource-id.txt` enthält eine Zeichenfolge, die die Ressourcen-ID identifiziert, die den Pfad zu der benutzerdefinierten Ressource über Ihren Amazon API Gateway-Endpunkt darstellt.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace custom-resource \
    --scalable-dimension custom-resource:ResourceType:Property \
    --resource-id file://~/custom-resource-id.txt \
    --min-capacity 1 \
    --max-capacity 10
  ```

  Inhalt von `custom-resource-id.txt`:

```
https://example.execute-api.us-west-2.amazonaws.com/prod/
scalableTargetDimensions/1-23456789
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Rufen Sie den Vorgang RegisterScalableTarget auf und geben Sie `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity`, und `MaxCapacity` als Parameter an.

Zugehörige Ressourcen

Wenn Sie gerade erst mit Application Auto Scaling beginnen, finden Sie in der folgenden Dokumentation weitere nützliche Informationen zur Skalierung Ihrer benutzerdefinierten Ressourcen:

GitHub -Repository

Einrichten, um Application Auto Scaling zu verwenden

Führen Sie die Aufgaben in diesem Abschnitt aus, um Application Auto Scaling zum ersten Mal einzurichten:

Themen

- Melden Sie sich an für AWS
- Richten Sie das ein AWS CLI
- Wird verwendet AWS CloudShell , um mit Application Auto Scaling von der Befehlszeile aus zu arbeiten

Melden Sie sich an für AWS

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

Verwenden von Application Auto Scaling in AWS-Regionen

Application Auto Scaling ist in mehreren AWS-Regionen. Ein globales System AWS-Konto ermöglicht es Ihnen, mit Ressourcen in den meisten Regionen zu arbeiten. Wenn Sie Application Auto Scaling mit Ressourcen in der Region China verwenden, beachten Sie, dass Sie ein separates Amazon Web Services (China) Konto haben müssen. Darüber hinaus gibt es einige Unterschiede bei der Implementierung von Application Auto Scaling. Weitere Informationen zur Verwendung von

Application Auto Scaling in den chinesischen Regionen finden Sie unter Application Auto Scaling in China.

Nachdem Sie Ihre eingerichtet haben AWS-Konto, fahren Sie mit dem nächsten Thema fort:Richten Sie das ein AWS CLI.

Richten Sie das ein AWS CLI

Das AWS Command Line Interface (AWS CLI) ist ein einheitliches Entwicklertool für die Verwaltung von AWS Diensten, einschließlich Application Auto Scaling. Befolgen Sie die Schritte zum Herunterladen und Konfigurieren der AWS CLI.

Um das einzurichten AWS CLI

1. Laden Sie Version 1 oder 2 von AWS CLI herunter, installieren und konfigurieren Sie sie. Dieselbe Application Auto Scaling Scaling-Funktionalität ist in Version 1 und 2 verfügbar. Eine Anleitung finden Sie unter den folgenden Themen im AWS Command Line Interface - Benutzerhandbuch:

   AWS CLI Version 1

   - Installation, Aktualisierung und Deinstallation von AWS CLI
   - Konfigurieren von AWS CLI

   AWS CLI Version 2

   - Installation oder Aktualisierung der neuesten Version von AWS CLI
   - Schnelleinrichtung

   > **Note**
   >
   > Für CLI-Zugriff benötigen Sie eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel. Verwenden Sie möglichst temporäre Anmeldeinformationen anstelle langfristiger Zugriffsschlüssel. Temporäre Anmeldeinformationen bestehen aus einer Zugriffsschlüssel-ID, einem geheimen Zugriffsschlüssel und einem Sicherheits-Token, das angibt, wann die Anmeldeinformationen ablaufen. Um die Sicherheit Ihres zu erhöhen AWS-Konto, empfehlen wir Ihnen dringend, nicht die mit Ihrem AWS-Konto Root-Benutzer verknüpften Zugangsdaten zu verwenden. Weitere Informationen finden

Sie unter Programmgesteuerter Zugriff in der Allgemeine AWS-Referenz und unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

2. Um zu überprüfen, ob das AWS CLI Profil korrekt konfiguriert ist, führen Sie den folgenden Befehl in einem Befehlsfenster aus.

```
aws configure
```

Wenn das Profil korrekt konfiguriert wurde, sollte die Ausgabe ähnlich wie folgt aussehen.

```
AWS Access Key ID [****************52FQ]:
AWS Secret Access Key [****************xgyZ]:
Default region name [us-east-1]:
Default output format [json]:
```

3. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob die Application Auto Scaling Scaling-Befehle für installiert AWS CLI sind.

```
aws application-autoscaling help
```

Wird verwendet AWS CloudShell , um mit Application Auto Scaling von der Befehlszeile aus zu arbeiten

AWS CloudShell ermöglicht es Ihnen, die Installation AWS CLI in Ihrer Entwicklungsumgebung zu überspringen und sie AWS Management Console stattdessen in der zu verwenden. Sie vermeiden nicht nur die Installation, sondern müssen auch keine Anmeldeinformationen konfigurieren und keine Region angeben. Ihre AWS Management Console Sitzung bietet diesen Kontext für die AWS CLI. Sie können es verwenden AWS CloudShell , wenn es unterstützt wird AWS-Regionen.

Sie können AWS CLI Befehle für Dienste mit Ihrer bevorzugten Shell (Bash- PowerShell, oder Z-Shell) ausführen.

Sie können AWS CloudShell von der aus AWS Management Console mit einer der folgenden beiden Methoden starten:

- Wählen Sie das AWS CloudShell Symbol in der Navigationsleiste der Konsole. Es befindet sich rechts neben dem Suchfeld.

- Verwenden Sie das Suchfeld in der Navigationsleiste der Konsole, um nach der CloudShellOption zu suchen CloudShellund diese dann auszuwählen.

Beim ersten AWS CloudShell Start in einem neuen Browserfenster wird ein Begrüßungsfenster mit einer Liste der wichtigsten Funktionen angezeigt. Nachdem Sie dieses Panel geschlossen haben, werden Statusaktualisierungen bereitgestellt, während die Shell Ihre Konsolenanmeldeinformationen konfiguriert und weiterleitet. Wenn die Eingabeaufforderung angezeigt wird, ist die Shell für die Interaktion bereit.

Weitere Informationen zu diesem Service finden Sie im AWS CloudShell -Benutzerhandbuch.

Application-Auto-Scaling-Ressourcen mit AWS CloudFormation erstellen

Application Auto Scaling ist in integriert, einem Service AWS CloudFormation, der Ihnen hilft, Ihre - AWS Ressourcen zu modellieren und einzurichten, sodass Sie weniger Zeit für die Erstellung und Verwaltung Ihrer Ressourcen und Infrastruktur aufwenden müssen. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen beschreibt, und stellt diese Ressourcen für Sie AWS CloudFormation bereit und konfiguriert sie.

Wenn Sie verwenden AWS CloudFormation, können Sie Ihre Vorlage wiederverwenden, um Ihre Application Auto Scaling-Ressourcen konsistent und wiederholt einzurichten. Beschreiben Sie Ihre Ressourcen einmal und stellen Sie dann dieselben Ressourcen immer wieder in mehreren AWS-Konten und Regionen bereit.

Application Auto Scaling und AWS CloudFormation Vorlagen

Um Ressourcen für Application Auto Scaling und damit verbundene Dienste bereitzustellen und zu konfigurieren, müssen Sie AWS CloudFormation Templates verstehen. Vorlagen sind formatierte Textdateien in JSON oder YAML. Diese Vorlagen beschreiben die Ressourcen, die Sie in Ihren AWS CloudFormation Stacks bereitstellen möchten. Wenn Sie mit JSON oder YAML nicht vertraut sind, können Sie AWS CloudFormation Designer verwenden, um Ihnen den Einstieg in AWS CloudFormation Vorlagen zu erleichtern. Weitere Informationen finden Sie unter Was ist AWS CloudFormation -Designer? im AWS CloudFormation -Benutzerhandbuch.

Wenn Sie eine Stack-Vorlage für Application Auto Scaling-Ressourcen erstellen, müssen Sie die folgenden Angaben machen:

- Einen Namespace für den Zieldienst (z. B. **`appstream`**). Informationen zum Abrufen von Service-Namespaces finden Sie in der -AWS::ApplicationAutoScaling::ScalableTargetReferenz.
- Eine skalierbare Dimension, die mit der Zielressource verbunden ist (z. B. **`appstream:fleet:DesiredCapacity`**). Informationen zum Abrufen skalierbarer Dimensionen finden Sie in der -AWS::ApplicationAutoScaling::ScalableTargetReferenz.
- Eine Ressourcen-ID für die Zielressource (z. B. **`fleet/sample-fleet`**). Weitere Informationen zur Syntax und Beispiele für spezifische Ressourcen-IDs finden Sie in der -AWS::ApplicationAutoScaling::ScalableTargetReferenz.

- Eine mit einem Service verknüpfte Rolle für die Zielressource (z. B. **`arn:aws:iam::012345678910:role/aws-service-role/appstream.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_AppStreamFleet`**). In der Tabelle ARN-Referenz für serviceverknüpfte Rollen finden Sie ARNs für Rollen.

Weitere Informationen zu Application-Auto-Scaling-Ressourcen finden Sie in der Application Auto Scaling-Referenz im AWS CloudFormation -Benutzerhandbuch.

Beispielvorlagen-Snippets

In den folgenden Abschnitten des AWS CloudFormation -Benutzerhandbuchs finden Sie Beispielausschnitte, die in AWS CloudFormation Vorlagen aufgenommen werden sollen:

- Beispiele für Skalierungsrichtlinien und geplante Aktionen finden Sie unter Konfigurieren von Application Auto Scaling-Ressourcen mit AWS CloudFormation.
- Weitere Beispiele für Skalierungsrichtlinien finden Sie unter AWS::ApplicationAutoScaling::ScalingPolicy.

Weitere Informationen über AWS CloudFormation

Weitere Informationen zu finden Sie AWS CloudFormationin den folgenden Ressourcen:

- AWS CloudFormation
- AWS CloudFormation Benutzerhandbuch
- AWS CloudFormation API Reference
- AWS CloudFormation -Benutzerhandbuch für die Befehlszeilenschnittstelle

Geplante Skalierung

Mit der geplanten Skalierung können Sie eine automatische Skalierung für Ihre Anwendung auf der Grundlage vorhersehbarer Laständerungen einrichten, indem Sie geplante Aktionen erstellen, mit denen die Kapazität zu bestimmten Zeiten erhöht oder verringert wird. So können Sie Ihre Anwendung proaktiv skalieren, um sie an vorhersehbare Laständerungen anzupassen.

Beispiel: Angenommen, Sie haben ein regelmäßiges wöchentliches Datenverkehrsmuster, bei dem die Last in der Wochenmitte steigt und gegen Ende der Woche sinkt. Sie können in Application Auto Scaling einen Skalierungsplan konfigurieren, der sich an diesem Muster orientiert:

- Am Mittwochmorgen erhöht eine geplante Aktion die Kapazität, indem die zuvor festgelegte Mindestkapazität des skalierbaren Ziels erhöht wird.
- Am Freitagabend verringert eine weitere geplante Aktion die Kapazität, indem die zuvor festgelegte maximale Kapazität des skalierbaren Ziels verringert wird.

Mit diesen geplanten Skalierungsaktionen können Sie Kosten und Leistung optimieren. Ihre Anwendung verfügt über ausreichend Kapazität, um die Hauptverkehrsspitzen unter der Woche zu bewältigen, ohne dass zu anderen Zeiten unnötige Kapazitäten bereitgestellt werden.

Sie können geplante Skalierung und Skalierungsrichtlinien zusammen verwenden, um die Vorteile von proaktiven und reaktiven Skalierungsansätzen zu nutzen. Nachdem eine geplante Skalierungsaktion ausgeführt wurde, kann die Skalierungsrichtlinie weiterhin Entscheidungen darüber treffen, ob die Kapazität weiter skaliert werden soll. So können Sie sicherstellen, dass Sie über eine ausreichende Kapazität verfügen, um die Last für Ihre Anwendung zu bewältigen. Während sich Ihre Anwendung an die Nachfrage anpasst, muss die aktuelle Kapazität innerhalb der minimalen und maximalen Kapazität liegen, die durch Ihre geplante Aktion festgelegt wurde.

Themen

- So funktioniert die geplante Skalierung
- Planung der wiederkehrenden Skalierungsaktionen mit Cron-Ausdrücken
- Beispiel für geplante Aktionen für Application Auto Scaling
- Verwalten der geplanten Skalierung von Application Auto Scaling
- Tutorial: Erste Schritte mit der geplanten Skalierung mit AWS CLI

So funktioniert die geplante Skalierung

In diesem Thema wird beschrieben, wie die geplante Skalierung funktioniert, und es werden die wichtigsten Überlegungen vorgestellt, die Sie verstehen müssen, um sie effektiv nutzen zu können.

Inhalt

- Funktionsweise
- Überlegungen
- Häufig verwendete Befehle zur Erstellung, Verwaltung und Löschung von geplanten Aktionen
- Zugehörige Ressourcen
- Einschränkungen

Funktionsweise

Um die geplante Skalierung zu nutzen, erstellen Sie geplante Aktionen, die Application Auto Scaling anweisen, zu bestimmten Zeiten Skalierungsaktivitäten durchzuführen. Wenn Sie eine geplante Aktion erstellen, geben Sie das skalierbare Ziel, den Zeitpunkt der Skalierungsaktivität sowie eine Mindest- und eine Maximalkapazität an. Sie können geplante Aktionen erstellen, die nur einmal skalieren oder wiederholt geplant ausgeführt werden.

Zum festgelegten Zeitpunkt skaliert Application Auto Scaling auf der Grundlage der neuen Kapazitätswerte, indem die aktuelle Kapazität mit der festgelegten Mindest- und Höchstkapazität verglichen wird.

- Wenn die aktuelle Kapazität geringer ist als die angegebene Mindestkapazität, wird Application Auto Scaling auf die angegebene Mindestkapazität erweitert (erhöht).
- Wenn die aktuelle Kapazität größer als die angegebene Maximalkapazität ist, skaliert Application Auto Scaling auf die angegebene Maximalkapazität hinein (verringert die Kapazität).

Überlegungen

Beachten Sie bei der Erstellung einer geplanten Aktion Folgendes:

- Eine geplante Aktion setzt die `MinCapacity` und `MaxCapacity` zum angegebenen Zeitpunkt und Datum auf den Wert, der durch die geplante Aktion angegeben ist. Die Anfrage kann optional

nur eine dieser Größen enthalten. Sie können beispielsweise eine geplante Aktion erstellen, bei der nur die minimale Kapazität angegeben ist. In einigen Fällen müssen Sie jedoch beide Größen einbeziehen, um sicherzustellen, dass die neue Mindestkapazität nicht größer als die maximale Kapazität ist oder die neue maximale Kapazität nicht unter der Mindestkapazität liegt.

- Standardmäßig befinden sich die wiederkehrenden Zeitpläne in UTC (Coordinated Universal Time). Sie können die Zeitzone ändern, wenn sie Ihrer örtlichen Zeitzone oder einer Zeitzone in einem anderen Teil Ihres Netzwerks entsprechen soll. Wenn Sie eine Zeitzone angeben, in der die Sommerzeit gilt, wird die Aktion automatisch an die Sommerzeit angepasst. Weitere Informationen finden Sie unter Planung der wiederkehrenden Skalierungsaktionen mit Cron-Ausdrücken.
- Sie können die geplante Skalierung für ein skalierbares Ziel vorübergehend deaktivieren. Dadurch können Sie verhindern, dass geplante Aktionen aktiv sind, ohne sie löschen zu müssen. Sie können die geplante Skalierung dann fortsetzen, wenn Sie sie erneut verwenden möchten. Weitere Informationen finden Sie unter Die Skalierung von Application Auto Scaling unterbrechen und wiederaufnehmen.
- Die Reihenfolge, in der geplante Aktionen ausgeführt werden, ist für dasselbe skalierbare Ziel garantiert, jedoch nicht für geplante Aktionen über skalierbare Ziele hinweg.
- Um eine geplante Aktion erfolgreich abzuschließen, muss sich die angegebene Ressource im Zielservice in einem skalierbaren Zustand befinden. Ist dies nicht der Fall, schlägt die Anforderung fehl und gibt eine Fehlermeldung zurück, z. B. `Resource Id [ActualResourceId] is not scalable. Reason: The status of all DB instances must be 'available' or 'incompatible-parameters'`.
- Aufgrund der verteilten Natur von Application Auto Scaling und den Zieldiensten kann die Verzögerung zwischen dem Zeitpunkt der Auslösung der geplanten Aktion und dem Zeitpunkt, zu dem der Zieldienst die Skalierungsaktion honoriert, einige Sekunden betragen. Da geplante Aktionen in der Reihenfolge ausgeführt werden, in der sie angegeben wurden, kann die Ausführung von geplanten Aktionen mit nahe beieinander liegenden Startzeiten länger dauern.

Häufig verwendete Befehle zur Erstellung, Verwaltung und Löschung von geplanten Aktionen

Zu den häufig verwendeten Befehlen für die Arbeit mit der Zeitplan-Skalierung gehören:

- register-scalable-targetum Ressourcen als skalierbare Ziele zu registrieren AWS oder anzupassen (eine Ressource, die Application Auto Scaling skalieren kann) und die Skalierung auszusetzen und wieder aufzunehmen.

- put-scheduled-actionum geplante Aktionen für ein vorhandenes skalierbares Ziel hinzuzufügen oder zu ändern.
- describe-scaling-activitiesum Informationen über Skalierungsaktivitäten in einer AWS Region zurückzugeben.
- describe-scheduled-actionsum Informationen über geplante Aktionen in einer AWS Region zurückzugeben.
- delete-scheduled-actionum eine geplante Aktion zu löschen.

Zugehörige Ressourcen

Ein detailliertes Beispiel für die Verwendung der geplanten Skalierung finden Sie im Blogbeitrag Scheduling AWS Lambda Provisioned Concurrency for recurriring peak usage auf dem AWS Compute-Blog.

Eine Anleitung zur Erstellung von geplanten Aktionen unter Verwendung von Beispiel- AWS -Ressourcen finden Sie unter Tutorial: Erste Schritte mit der geplanten Skalierung mit AWS CLI.

Informationen zum Erstellen von geplanten Aktionen für Auto Scaling finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling im Benutzerhandbuch für Amazon EC2 Auto Scaling.

Einschränkungen

Die folgenden Einschränkungen gelten für die Verwendung der geplanten Skalierung:

- Die Namen der geplanten Aktionen müssen pro skalierbarem Ziel eindeutig sein.
- Application Auto Scaling bietet keine Präzision zweiter Ebene in Zeitplanausdrücken. Die feinste Zeitauflösung bei Verwendung eines Cron-Ausdrucks ist 1 Minute.
- Das skalierbare Ziel kann nicht ein Amazon MSK-Cluster sein. Geplante Skalierung wird für Amazon MSK nicht unterstützt.
- Der Konsolenzugriff zum Anzeigen, Hinzufügen, Aktualisieren oder Entfernen von geplanten Aktionen für skalierbare Ressourcen hängt von der verwendeten Ressource ab. Weitere Informationen finden Sie unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

Planung der wiederkehrenden Skalierungsaktionen mit Cron-Ausdrücken

> **Important**
>
> Für Hilfe zu Cron-Ausdrücken für Amazon EC2 Auto Scaling lesen Sie das Thema Regelmäßige Zeitpläne im Benutzerhandbuch zu Amazon EC2 Auto Scaling. Mit Amazon EC2 Auto Scaling verwenden Sie die herkömmliche Cron-Syntax anstelle der benutzerdefinierten Cron-Syntax, die Application Auto Scaling verwendet.

Mithilfe eines Cron-Ausdrucks können Sie geplante Aktionen erstellen, die nach einem wiederkehrenden Zeitplan ausgeführt werden.

Um einen wiederkehrenden Zeitplan zu erstellen, geben Sie einen Cron-Ausdruck und eine Zeitzone an, um zu beschreiben, wann diese geplante Aktion wiederholt werden soll. Die unterstützten Zeitzonenwerte sind die kanonischen Namen der von Joda-Time unterstützten IANA-Zeitzonen (z. B. `Etc/GMT+9` oder `Pacific/Tahiti`). Sie können optional ein Datum und eine Uhrzeit für die Startzeit, die Endzeit oder beides angeben. Ein Beispielbefehl, der das verwendet, um eine geplante Aktion AWS CLI zu erstellen, finden Sie unter. Erstellen einer wiederkehrenden geplanten Aktion, die eine Zeitzone angibt

Der unterstützte Cron-Ausdruck besteht aus sechs Feldern, getrennt durch Leerzeichen: [Minute] [Stunde] [Tag_des_Monats] [Monat_des_Jahres] [Wochentag] [Jahr]. Beispielsweise konfiguriert der Cron-Ausdruck `30 6 ? * MON *` eine geplante Aktion, die jeden Montag um 6:30 Uhr wiederholt wird. Das Sternchen wird als Platzhalter verwendet, um alle Werte für ein Feld abzugleichen.

Weitere Informationen zur Cron-Syntax für geplante Aktionen von Application Auto Scaling finden Sie unter Referenz zu Cron-Ausdrücken im EventBridge Amazon-Benutzerhandbuch.

Wählen Sie bei der Erstellung eines wiederkehrenden Zeitplans Ihre Start- und Endzeiten sorgfältig aus. Beachten Sie Folgendes:

- Wenn Sie eine Startzeit angeben, führt Application Auto Scaling die Aktion zu dieser Zeit aus und führt die Aktion dann auf der Grundlage der angegebenen Wiederholung aus.
- Wenn Sie eine Endzeit angeben, wird die Aktion nach dieser Zeit nicht mehr wiederholt. Application Auto Scaling merkt sich keine früheren Werte und kehrt nach der Endzeit zu diesen früheren Werten zurück.

- Die Start- und Endzeit müssen in UTC festgelegt werden, wenn Sie die SDKs AWS CLI oder die AWS SDKs verwenden, um eine geplante Aktion zu erstellen oder zu aktualisieren.

Beispiele

Sie können sich auf die folgende Tabelle beziehen, wenn Sie einen wiederkehrenden Zeitplan für ein skalierbares Application-Auto-Scaling-Ziel erstellen. Die folgenden Beispiele zeigen die korrekte Syntax für die Verwendung von Application Auto Scaling zum Erstellen oder Aktualisieren einer geplanten Aktion.

Minuten	Stunden	Tag des Monats	Monat	Wochentag	Jahr	Bedeutung
0	10	*	*	?	*	Ausführung jeden Tag um 10:00 Uhr (UTC)
15	12	*	*	?	*	Ausführung jeden Tag um 12:15 Uhr (UTC)
0	18	?	*	MO-FR	*	Ausführung jeden Montag bis Freitag um 18:00 Uhr (UTC)
0	8	1	*	?	*	Lauf um 8:00 Uhr (UTC) am 1. Tag

Minuten	Stunden	Tag des Monats	Monat	Wochentag	Jahr	Bedeutung
						eines jeden Monats
0/15	*	*	*	?	*	Ausführung alle 15 Minuten
0/10	*	?	*	MO-FR	*	Ausführung alle 10 Minuten von Montag bis Freitag
0/5	8-17	?	*	MO-FR	*	Ausführung alle 5 Minuten von Montag bis Freitag zwischen 08:00 Uhr und 17:55 Uhr (UTC)

Exception

Sie können auch einen Cron-Ausdruck mit einem Zeichenfolgenwert erstellen, der sieben Felder enthält. In diesem Fall können Sie die ersten drei Felder verwenden, um den Zeitpunkt anzugeben, zu dem eine geplante Aktion ausgeführt werden soll, einschließlich der Sekunden. Der vollständige Cron-Ausdruck hat die folgenden durch Leerzeichen getrennten Felder: [Sekunden] [Minuten] [Stunden] [Tag_des_Monats] [Monat] [Tag_des_Woche] [Jahr]. Dieser Ansatz garantiert jedoch nicht, dass die geplante Aktion genau in der von Ihnen angegebenen Sekunde ausgeführt wird.

Außerdem kann es sein, dass einige Service-Konsolen das Sekundenfeld in einem Cron-Ausdruck nicht unterstützen.

Beispiel für geplante Aktionen für Application Auto Scaling

Die folgenden Beispiele zeigen, wie Sie geplante Aktionen mit dem AWS CLI put-scheduled-actionBefehl erstellen. Wenn Sie die neue Kapazität angeben, können Sie eine Mindestkapazität, eine Maximalkapazität oder beides angeben.

Der Kürze halber werden in den Beispielen in diesem Thema CLI-Befehle für einige der Dienste erläutert, die in Application Auto Scaling integriert sind. Um ein anderes skalierbares Ziel anzugeben, geben Sie seinen Namespace in `--service-namespace`, seine skalierbare Dimension in `--scalable-dimension` und seine Ressourcen-ID in `--resource-id` an. Weitere Informationen und Beispiele für die einzelnen Services finden Sie in den Themen unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

Denken Sie bei der Verwendung von daran AWS CLI, dass Ihre Befehle in der für Ihr Profil AWS-Region konfigurierten Version ausgeführt werden. Wenn Sie die Befehle in einer anderen Region ausführen möchten, ändern Sie entweder die Standardregion für Ihr Profil, oder verwenden Sie den `--region`-Parameter mit dem Befehl.

Inhalt

- Erstellen einer geplanten Aktion, die nur einmal ausgeführt wird
- Erstellen einer geplanten Aktion, die in einem wiederkehrenden Intervall ausgeführt wird
- Erstellen einer geplanten Aktion, die nach einem wiederkehrenden Zeitplan ausgeführt wird
- Erstellen einer einmaligen geplanten Aktion, die eine Zeitzone angibt
- Erstellen einer wiederkehrenden geplanten Aktion, die eine Zeitzone angibt

Erstellen einer geplanten Aktion, die nur einmal ausgeführt wird

Um Ihr skalierbares Ziel nur einmal zu einem bestimmten Datum und einer bestimmten Uhrzeit automatisch zu skalieren, verwenden Sie die Option `--schedule "at(`*`yyyy-mm-ddThh:mm:ss`*`)"`.

Example Beispiel: Einmalige horizontale Skalierung nach oben

Nachfolgend ein Beispiel für die Erstellung einer geplanten Aktion zum Abbau von Kapazitäten zu einem bestimmten Datum und einer bestimmten Uhrzeit.

Wenn der für `MinCapacity` angegebene Wert zu dem für `--schedule` angegebenen Datum und Zeitpunkt (22:00 Uhr UTC am 31. März 2021) über der aktuellen Kapazität liegt, skaliert Application Auto Scaling auf `MinCapacity`.

Linux, macOS oder Unix

```
aws application-autoscaling put-scheduled-action --service-namespace custom-resource \
  --scalable-dimension custom-resource:ResourceType:Property \
  --resource-id file://~/custom-resource-id.txt \
  --scheduled-action-name scale-out \
  --schedule "at(2021-03-31T22:00:00)" \
  --scalable-target-action MinCapacity=3
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace custom-resource --
scalable-dimension custom-resource:ResourceType:Property --resource-id file://~/custom-
resource-id.txt --scheduled-action-name scale-out --schedule "at(2021-03-31T22:00:00)"
 --scalable-target-action MinCapacity=3
```

> **Note**
>
> Wenn diese eingeplante Aktion ausgeführt wird und die maximale Kapazität unter dem für die minimale Kapazität angegebenen Wert liegt, müssen Sie eine neue minimale und maximale Kapazität angeben und nicht nur die minimale Kapazität.

Example Beispiel: Einmalige horizontale Skalierung nach unten

Nachfolgend ein Beispiel für die Erstellung einer geplanten Aktion zur Kapazitätsanpassung zu einem bestimmten Datum und einer bestimmten Uhrzeit.

Wenn der für `MaxCapacity` angegebene Wert zu dem für `--schedule` angegebenen Datum und Zeitpunkt (22:30 Uhr UTC am 31. März 2021) unter der aktuellen Kapazität liegt, skaliert Application Auto Scaling automatisch ab auf `MaxCapacity`.

Linux, macOS oder Unix

```
aws application-autoscaling put-scheduled-action --service-namespace custom-resource \
  --scalable-dimension custom-resource:ResourceType:Property \
```

```
  --resource-id file://~/custom-resource-id.txt \
  --scheduled-action-name scale-in \
  --schedule "at(2021-03-31T22:30:00)" \
  --scalable-target-action MinCapacity=0,MaxCapacity=0
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace custom-resource --
scalable-dimension custom-resource:ResourceType:Property --resource-id file://~/custom-
resource-id.txt --scheduled-action-name scale-in --schedule "at(2021-03-31T22:30:00)"
 --scalable-target-action MinCapacity=0,MaxCapacity=0
```

Erstellen einer geplanten Aktion, die in einem wiederkehrenden Intervall ausgeführt wird

Um eine Skalierung in einem wiederkehrenden Intervall zu planen, verwenden Sie die Option `--schedule "rate(value unit)"`. Der Wert muss eine positive ganze Zahl sein. Die Einheit kann `minute`, `minutes`, `hour`, `hours`, `day` oder `days` sein. Weitere Informationen finden Sie unter Bewertungsausdrücke im Amazon CloudWatch Events-Benutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine geplante Aktion, die einen Ratenausdruck verwendet.

Wenn der für `MinCapacity` angegebene Wert im angegebenen Zeitplan (alle 5 Stunden, beginnend am 30. Januar 2021 um 12:00 Uhr UTC und endend am 31. Januar 2021 um 22:00 Uhr UTC) über der aktuellen Kapazität liegt, skaliert Application Auto Scaling automatisch auf `MinCapacity`. Wenn der für `MaxCapacity` angegebene Wert unter der aktuellen Kapazität liegt, skaliert Application Auto Scaling auf `MaxCapacity`.

Linux, macOS oder Unix

```
aws application-autoscaling put-scheduled-action --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service \
  --scheduled-action-name my-recurring-action \
  --schedule "rate(5 hours)" \
  --start-time 2021-01-30T12:00:00 \
  --end-time 2021-01-31T22:00:00 \
  --scalable-target-action MinCapacity=3,MaxCapacity=10
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace ecs --scalable-
dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-service
 --scheduled-action-name my-recurring-action --schedule "rate(5 hours)" --start-
time 2021-01-30T12:00:00 --end-time 2021-01-31T22:00:00 --scalable-target-action
 MinCapacity=3,MaxCapacity=10
```

Erstellen einer geplanten Aktion, die nach einem wiederkehrenden Zeitplan ausgeführt wird

Planen Sie eine Skalierung im Rahmen eines sich wiederholenden Zeitplans unter Verwendung der `--schedule "cron(fields)"`-Option. Weitere Informationen finden Sie unter Planung der wiederkehrenden Skalierungsaktionen mit Cron-Ausdrücken.

Im Folgenden finden Sie ein Beispiel für eine geplante Aktion, die einen Cron-Ausdruck verwendet.

Wenn der für `MinCapacity` angegebene Wert im angegebenen Zeitplan (täglich um 9:00 Uhr UTC) über der aktuellen Kapazität liegt, skaliert Application Auto Scaling auf `MinCapacity`. Wenn der für `MaxCapacity` angegebene Wert unter der aktuellen Kapazität liegt, skaliert Application Auto Scaling auf `MaxCapacity`.

Linux, macOS oder Unix

```
aws application-autoscaling put-scheduled-action --service-namespace appstream \
  --scalable-dimension appstream:fleet:DesiredCapacity \
  --resource-id fleet/sample-fleet \
  --scheduled-action-name my-recurring-action \
  --schedule "cron(0 9 * * ? *)" \
  --scalable-target-action MinCapacity=10,MaxCapacity=50
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace appstream --
scalable-dimension appstream:fleet:DesiredCapacity --resource-id fleet/sample-fleet --
scheduled-action-name my-recurring-action --schedule "cron(0 9 * * ? *)" --scalable-
target-action MinCapacity=10,MaxCapacity=50
```

Erstellen einer einmaligen geplanten Aktion, die eine Zeitzone angibt

Geplante Aktionen sind standardmäßig auf die UTC-Zeitzone eingestellt. Um eine andere Zeitzone anzugeben, fügen Sie die Option `--timezone` ein und geben den kanonischen Namen für

die Zeitzone an (z. B. `America/New_York`). Weitere Informationen finden Sie unter https://www.joda.org/joda-time/timezones.html. Dort finden Sie Informationen zu den IANA-Zeitzonen, die bei Anrufen put-scheduled-actionunterstützt werden.

Im folgenden Beispiel wird die Option `--timezone` bei der Erstellung einer geplanten Aktion zur Skalierung der Kapazität zu einem bestimmten Datum und einer bestimmten Uhrzeit verwendet.

Wenn der für `MinCapacity` angegebene Wert zu dem für `--schedule` angegebenen Datum und Zeitpunkt (17:00 Uhr Ortszeit am 31. Januar 2021) über der aktuellen Kapazität liegt, skaliert Application Auto Scaling auf `MinCapacity` ab. Wenn der für `MaxCapacity` angegebene Wert unter der aktuellen Kapazität liegt, skaliert Application Auto Scaling auf `MaxCapacity`.

Linux, macOS oder Unix

```
aws application-autoscaling put-scheduled-action --service-namespace comprehend \
  --scalable-dimension comprehend:document-classifier-endpoint:DesiredInferenceUnits \
  --resource-id arn:aws:comprehend:us-west-2:123456789012:document-classifier-endpoint/
EXAMPLE \
  --scheduled-action-name  my-one-time-action \
  --schedule "at(2021-01-31T17:00:00)" --timezone "America/New_York" \
  --scalable-target-action MinCapacity=1,MaxCapacity=3
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace comprehend --
scalable-dimension comprehend:document-classifier-endpoint:DesiredInferenceUnits
 --resource-id arn:aws:comprehend:us-west-2:123456789012:document-classifier-
endpoint/EXAMPLE --scheduled-action-name  my-one-time-action --schedule
 "at(2021-01-31T17:00:00)" --timezone "America/New_York" --scalable-target-action
 MinCapacity=1,MaxCapacity=3
```

Erstellen einer wiederkehrenden geplanten Aktion, die eine Zeitzone angibt

Es folgt ein Beispiel, bei dem die `--timezone`-Option verwendet wird, wenn Sie eine wiederkehrende geplante Aktion zum Skalieren der Kapazität erstellen. Weitere Informationen finden Sie unter Planung der wiederkehrenden Skalierungsaktionen mit Cron-Ausdrücken.

Wenn der für `MinCapacity` angegebene Wert im angegebenen Zeitplan (jeden Montag bis Freitag um 18:00 Uhr Ortszeit) über der aktuellen Kapazität liegt, skaliert Application Auto Scaling auf `MinCapacity` ab. Wenn der für `MaxCapacity` angegebene Wert unter der aktuellen Kapazität liegt, skaliert Application Auto Scaling auf `MaxCapacity`.

Linux, macOS oder Unix

```
aws application-autoscaling put-scheduled-action --service-namespace lambda \
  --scalable-dimension lambda:function:ProvisionedConcurrency \
  --resource-id function:my-function:BLUE \
  --scheduled-action-name my-recurring-action \
  --schedule "cron(0 18 ? * MON-FRI *)" --timezone "Etc/GMT+9" \
  --scalable-target-action MinCapacity=10,MaxCapacity=50
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace lambda
 --scalable-dimension lambda:function:ProvisionedConcurrency --resource-
id function:my-function:BLUE --scheduled-action-name my-recurring-action --schedule
 "cron(0 18 ? * MON-FRI *)" --timezone "Etc/GMT+9" --scalable-target-action
 MinCapacity=10,MaxCapacity=50
```

Verwalten der geplanten Skalierung von Application Auto Scaling

AWS CLI Dazu gehören mehrere andere Befehle, mit denen Sie Ihre geplanten Aktionen verwalten können.

Der Kürze halber werden in den Beispielen in diesem Thema CLI-Befehle für einige der Dienste erläutert, die in Application Auto Scaling integriert sind. Um ein anderes skalierbares Ziel anzugeben, geben Sie seinen Namespace in `--service-namespace`, seine skalierbare Dimension in `--scalable-dimension` und seine Ressourcen-ID in `--resource-id` an. Weitere Informationen und Beispiele für die einzelnen Services finden Sie in den Themen unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

Denken Sie bei der Verwendung von daran AWS CLI, dass Ihre Befehle in der für Ihr Profil AWS-Region konfigurierten Version ausgeführt werden. Wenn Sie die Befehle in einer anderen Region ausführen möchten, ändern Sie entweder die Standardregion für Ihr Profil, oder verwenden Sie den `--region`-Parameter mit dem Befehl.

Inhalt

- Anzeige der Skalierungsaktivitäten für einen bestimmten Service
- Beschreiben aller geplanten Aktionen für einen bestimmten Dienst
- Beschreiben einer oder mehrerer geplanter Aktionen für ein skalierbares Ziel
- Ausschalten der geplanten Skalierung für ein skalierbares Ziel

- Löschen einer geplanten Aktion

Anzeige der Skalierungsaktivitäten für einen bestimmten Service

Verwenden Sie den describe-scaling-activitiesBefehl, um die Skalierungsaktivitäten für alle skalierbaren Ziele in einem angegebenen Service-Namespace anzuzeigen.

Das folgende Beispiel ruft die Skalierungsaktivitäten ab, die mit dem Service-Namespace dynamodb verbunden sind.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scaling-activities --service-namespace dynamodb
```

Windows

```
aws application-autoscaling describe-scaling-activities --service-namespace dynamodb
```

Wird der Befehl erfolgreich ausgeführt, wird Ihnen eine Ausgabe ähnlich der folgenden angezeigt.

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 10.",
            "ResourceId": "table/my-table",
            "ActivityId": "4d1308c0-bbcf-4514-a673-b0220ae38547",
            "StartTime": 1561574415.086,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574449.51,
            "Cause": "maximum capacity was set to 10",
            "StatusMessage": "Successfully set write capacity units to 10. Change
 successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 5 and max capacity to 10",
            "ResourceId": "table/my-table",
            "ActivityId": "f2b7847b-721d-4e01-8ef0-0c8d3bacc1c7",
            "StartTime": 1561574414.644,
            "ServiceNamespace": "dynamodb",
```

```
            "Cause": "scheduled action name my-second-scheduled-action was triggered",
            "StatusMessage": "Successfully set min capacity to 5 and max capacity to
 10",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 15.",
            "ResourceId": "table/my-table",
            "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
            "StartTime": 1561574108.904,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574140.255,
            "Cause": "minimum capacity was set to 15",
            "StatusMessage": "Successfully set write capacity units to 15. Change
 successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 15 and max capacity to 20",
            "ResourceId": "table/my-table",
            "ActivityId": "3250fd06-6940-4e8e-bb1f-d494db7554d2",
            "StartTime": 1561574108.512,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-first-scheduled-action was triggered",
            "StatusMessage": "Successfully set min capacity to 15 and max capacity to
 20",
            "StatusCode": "Successful"
        }
    ]
}
```

Um diesen Befehl so zu ändern, dass er nur die Skalierungsaktivitäten für eines Ihrer skalierbaren Ziele abruft, fügen Sie die Option `--resource-id` hinzu.

Beschreiben aller geplanten Aktionen für einen bestimmten Dienst

Verwenden Sie den Befehl, um die geplanten Aktionen für alle skalierbaren Ziele in einem angegebenen Service-Namespace zu beschreiben. describe-scheduled-actions

Das folgende Beispiel ruft die geplanten Aktionen ab, die mit dem Service-Namespace `ec2` verbunden sind.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scheduled-actions --service-namespace ec2
```

Windows

```
aws application-autoscaling describe-scheduled-actions --service-namespace ec2
```

Ist der Befehl erfolgreich, wird eine Ausgabe zurückgegeben, die wie folgt aussehen sollte.

```
{
    "ScheduledActions": [
        {
            "ScheduledActionName": "my-one-time-action",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:493a6261-fbb9-432d-855d-3c302c14bdb9:resource/ec2/
spot-fleet-request/sfr-107dc873-0802-4402-a901-37294EXAMPLE:scheduledActionName/my-one-
time-action",
            "ServiceNamespace": "ec2",
            "Schedule": "at(2021-01-31T17:00:00)",
            "Timezone": "America/New_York",
            "ResourceId": "spot-fleet-request/sfr-107dc873-0802-4402-
a901-37294EXAMPLE",
            "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
            "ScalableTargetAction": {
                "MaxCapacity": 1
            },
            "CreationTime": 1607454792.331
        },
        {
            "ScheduledActionName": "my-recurring-action",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:493a6261-fbb9-432d-855d-3c302c14bdb9:resource/ec2/
spot-fleet-request/sfr-107dc873-0802-4402-a901-37294EXAMPLE:scheduledActionName/my-
recurring-action",
            "ServiceNamespace": "ec2",
            "Schedule": "rate(5 minutes)",
            "ResourceId": "spot-fleet-request/sfr-107dc873-0802-4402-
a901-37294EXAMPLE",
            "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
            "StartTime": 1604059200.0,
            "EndTime": 1612130400.0,
```

```
            "ScalableTargetAction": {
                "MinCapacity": 3,
                "MaxCapacity": 10
            },
            "CreationTime": 1607454949.719
        },
        {
            "ScheduledActionName": "my-one-time-action",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:4bce34c7-bb81-4ecf-b776-5c726efb1567:resource/ec2/
spot-fleet-request/sfr-40edeb7b-9ae7-44be-bef2-5c4c8EXAMPLE:scheduledActionName/my-one-
time-action",
            "ServiceNamespace": "ec2",
            "Schedule": "at(2020-12-08T9:36:00)",
            "Timezone": "America/New_York",
            "ResourceId": "spot-fleet-request/sfr-40edeb7b-9ae7-44be-
bef2-5c4c8EXAMPLE",
            "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
            "ScalableTargetAction": {
                "MinCapacity": 1,
                "MaxCapacity": 3
            },
            "CreationTime": 1607456031.391
        }
    ]
}
```

Beschreiben einer oder mehrerer geplanter Aktionen für ein skalierbares Ziel

Um Informationen über die geplanten Aktionen für ein bestimmtes skalierbares Ziel abzurufen, fügen Sie die `--resource-id` Option hinzu, wenn Sie geplante Aktionen mit dem describe-scheduled-actionsBefehl beschreiben.

Wenn Sie die Option `--scheduled-action-names` hinzufügen und den Namen einer geplanten Aktion als Wert angeben, gibt der Befehl nur die geplante Aktion zurück, deren Name übereinstimmt, wie im folgenden Beispiel gezeigt.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scheduled-actions --service-namespace ec2 \
  --resource-id spot-fleet-request/sfr-40edeb7b-9ae7-44be-bef2-5c4c8EXAMPLE \
```

```
  --scheduled-action-names my-one-time-action
```

Windows

```
aws application-autoscaling describe-scheduled-actions --service-namespace ec2 --
resource-id spot-fleet-request/sfr-40edeb7b-9ae7-44be-bef2-5c4c8EXAMPLE --scheduled-
action-names my-one-time-action
```

Es folgt eine Beispielausgabe.

```
{
    "ScheduledActions": [
        {
            "ScheduledActionName": "my-one-time-action",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:4bce34c7-bb81-4ecf-b776-5c726efb1567:resource/ec2/
spot-fleet-request/sfr-40edeb7b-9ae7-44be-bef2-5c4c8EXAMPLE:scheduledActionName/my-one-
time-action",
            "ServiceNamespace": "ec2",
            "Schedule": "at(2020-12-08T9:36:00)",
            "Timezone": "America/New_York",
            "ResourceId": "spot-fleet-request/sfr-40edeb7b-9ae7-44be-
bef2-5c4c8EXAMPLE",
            "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
            "ScalableTargetAction": {
                "MinCapacity": 1,
                "MaxCapacity": 3
            },
            "CreationTime": 1607456031.391
        }
    ]
}
```

Wenn mehr als ein Wert für die Option `--scheduled-action-names` angegeben wird, werden alle geplanten Aktionen, deren Namen übereinstimmen, in die Ausgabe aufgenommen.

Ausschalten der geplanten Skalierung für ein skalierbares Ziel

Sie können die geplante Skalierung vorübergehend deaktivieren, ohne Ihre geplanten Aktionen zu löschen. Weitere Informationen finden Sie unter Die Skalierung von Application Auto Scaling unterbrechen und wiederaufnehmen.

Unterbrechen Sie die geplante Skalierung auf einem skalierbaren Ziel, indem Sie den register-scalable-targetBefehl mit der `--suspended-state` Option verwenden und den Wert des `ScheduledScalingSuspended` Attributs angeben`true`, wie im folgenden Beispiel gezeigt.

Linux, macOS oder Unix

```
aws application-autoscaling register-scalable-target --service-namespace rds \
  --scalable-dimension rds:cluster:ReadReplicaCount --resource-id cluster:my-db-cluster
 \
  --suspended-state '{"ScheduledScalingSuspended": true}'
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace rds --
scalable-dimension rds:cluster:ReadReplicaCount --resource-id cluster:my-db-cluster --
suspended-state "{\"ScheduledScalingSuspended\": true}"
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Um die geplante Skalierung fortzusetzen, führen Sie diesen Befehl erneut aus und geben dabei `false` als Wert des Attributs `ScheduledScalingSuspended` an.

Löschen einer geplanten Aktion

Wenn Sie mit einer geplanten Aktion fertig sind, können Sie sie mit dem delete-scheduled-actionBefehl löschen.

Linux, macOS oder Unix

```
aws application-autoscaling delete-scheduled-action --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-37294EXAMPLE \
  --scheduled-action-name my-recurring-action
```

Windows

```
aws application-autoscaling delete-scheduled-action --service-namespace ec2 --scalable-dimension ec2:spot-fleet-request:TargetCapacity --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-37294EXAMPLE --scheduled-action-name my-recurring-action
```

Wenn dieser Befehl erfolgreich war, kehrt er zur Eingabeaufforderung zurück.

Tutorial: Erste Schritte mit der geplanten Skalierung mit AWS CLI

Das folgende Tutorial zeigt Ihnen, wie Sie mit der geplanten Skalierung beginnen können, indem es Ihnen hilft, geplante Aktionen AWS CLI zu erstellen, die eine DynamoDB-Beispieltabelle mit dem Namen skalieren. `TestTable` Wenn Sie noch keine `TestTable`-Tabelle in DynamoDB haben, die Sie für das Testen verwenden, können Sie jetzt eine erstellen, indem Sie den Befehl create-table verwenden, der in Schritt 1: Erstellen einer DynamoDB-Tabelle im Amazon DynamoDB Developer Guide gezeigt wird.

Denken Sie bei der Verwendung von daran AWS CLI, dass Ihre Befehle in der AWS Region ausgeführt werden, die für Ihr Profil konfiguriert ist. Wenn Sie die Befehle in einer anderen Region ausführen möchten, ändern Sie entweder die Standardregion für Ihr Profil, oder verwenden Sie den `--region`-Parameter mit dem Befehl.

> **Note**
>
> Im Rahmen dieses Tutorials können AWS Gebühren anfallen. Bitte überwachen Sie die Nutzung Ihres kostenlosen Kontingents und stellen Sie sicher, dass Sie die Kosten verstehen, die mit der Anzahl der Lese- und Schreibkapazitätseinheiten verbunden sind, die Ihre DynamoDB-Datenbanken nutzen.

Inhalt

- Schritt 1: Registrieren Sie Ihr skalierbares Ziel
- Schritt 2: Erstellen Sie zwei geplante Aktionen
- Schritt 3: Ansicht der Skalierungsaktivitäten
- Schritt 4: Nächste Schritte
- Schritt 5: Bereinigen

Schritt 1: Registrieren Sie Ihr skalierbares Ziel

Beginnen Sie mit der Registrierung Ihrer DynamoDB-Tabelle als skalierbares Ziel mit Application Auto Scaling.

So registrieren Sie Ihr skalierbares Ziel mit Application Auto Scaling

1. Verwenden Sie zunächst den describe-scalable-targetsBefehl, um zu überprüfen, ob DynamoDB-Ressourcen bereits registriert sind. Auf diese Weise können Sie überprüfen, dass die `TestTable`-Tabelle nicht registriert ist, falls es sich nicht um eine neue Tabelle handelt.

   Linux, macOS oder Unix

   ```
   aws application-autoscaling describe-scalable-targets \
     --service-namespace dynamodb
   ```

   Windows

   ```
   aws application-autoscaling describe-scalable-targets --service-namespace dynamodb
   ```

   Wenn es keine skalierbaren Ziele gibt, ist dies die Antwort.

   ```
   {
       "ScalableTargets": []
   }
   ```

2. Verwenden Sie den folgenden register-scalable-targetBefehl, um die Schreibkapazität Ihrer aufgerufenen DynamoDB-Tabelle zu registrieren. `TestTable` Legen Sie eine gewünschte Mindestkapazität von 5 Schreibkapazitätseinheiten und eine gewünschte maximale Kapazität von 10 Schreibkapazitätseinheiten fest.

   Linux, macOS oder Unix

   ```
   aws application-autoscaling register-scalable-target \
     --service-namespace dynamodb \
     --scalable-dimension dynamodb:table:WriteCapacityUnits \
     --resource-id table/TestTable \
     --min-capacity 5 --max-capacity 10
   ```

   Windows

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb
 --scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/
TestTable --min-capacity 5 --max-capacity 10
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-
id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Schritt 2: Erstellen Sie zwei geplante Aktionen

Bei Application Auto Scaling können Sie den Zeitpunkt festlegen, zu dem eine Skalierungsaktion durchgeführt werden soll. Sie geben das skalierbare Ziel, den Zeitplan und die Mindest- und maximale Kapazität an. Zum angegebenen Zeitpunkt aktualisiert Application Auto Scaling den minimalen und maximalen Wert für das skalierbare Ziel. Wenn die aktuelle Kapazität außerhalb dieses Bereichs liegt, führt dies zu einer Skalierungsaktion.

Das Planen von Updates für die Mindest- und maximale Kapazität ist auch nützlich, wenn Sie eine Skalierungsrichtlinie erstellen. Eine Skalierungsrichtlinie ermöglicht eine dynamische Skalierung Ihrer Ressourcen basierend auf der aktuellen Ressourcenauslastung. Geeignete Werte für die Mindest- und maximale Kapazität werden häufig als Orientierungspunkte für Skalierungsrichtlinien verwendet.

In dieser Übung werden zwei einmalige Aktionen für die horizontale Skalierung erstellt.

Erstellen und Anzeigen geplanter Aktionen

1. Verwenden Sie den folgenden put-scheduled-actionBefehl, um die erste geplante Aktion zu erstellen.

   Der Befehl at in `--schedule`-Zeitplänen plant die Aktion zur einmaligen Ausführung an einem bestimmten Zeitpunkt in der Zukunft. Stunden werden im 24-Stunden-Format in UTC angegeben. Planen Sie die Ausführung der Aktion in ungefähr 5 Minuten ab jetzt.

   Zum angegebenen Datum und zur angegebenen Uhrzeit aktualisiert Application Auto Scaling die Werte `MinCapacity` und `MaxCapacity`. Angenommen, die Tabelle hat derzeit 5

Schreibkapazitätseinheiten, skaliert Application Auto Scaling auf `MinCapacity`, um die Tabelle in den neuen gewünschten Bereich von 15-20 Schreibkapazitätseinheiten zu bringen.

Linux, macOS oder Unix

```
aws application-autoscaling put-scheduled-action \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:WriteCapacityUnits \
  --resource-id table/TestTable \
  --scheduled-action-name my-first-scheduled-action \
  --schedule "at(2019-05-20T17:05:00)" \
  --scalable-target-action MinCapacity=15,MaxCapacity=20
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace dynamodb
 --scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/
TestTable --scheduled-action-name my-first-scheduled-action --schedule
 "at(2019-05-20T17:05:00)" --scalable-target-action MinCapacity=15,MaxCapacity=20
```

Dieser Befehl gibt keine Ausgabe zurück, wenn er nicht erfolgreich ist.

2. Verwenden Sie den folgenden put-scheduled-actionBefehl, um die zweite geplante Aktion zu erstellen, die Application Auto Scaling für die Skalierung verwendet.

   Planen Sie die Ausführung der Aktion in ungefähr 10 Minuten ab jetzt.

   Zum angegebenen Datum und zur angegebenen Uhrzeit aktualisiert Application Auto Scaling die Tabellen `MinCapacity` und `MaxCapacity` und skaliert auf `MaxCapacity`, um die Tabelle auf den ursprünglich gewünschten Bereich von 5-10 Schreibkapazitätseinheiten zurückzuführen.

   Linux, macOS oder Unix

   ```
   aws application-autoscaling put-scheduled-action \
     --service-namespace dynamodb \
     --scalable-dimension dynamodb:table:WriteCapacityUnits \
     --resource-id table/TestTable \
     --scheduled-action-name my-second-scheduled-action \
     --schedule "at(2019-05-20T17:10:00)" \
     --scalable-target-action MinCapacity=5,MaxCapacity=10
   ```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace dynamodb
 --scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/
TestTable --scheduled-action-name my-second-scheduled-action --schedule
 "at(2019-05-20T17:10:00)" --scalable-target-action MinCapacity=5,MaxCapacity=10
```

3. (Optional) Rufen Sie mit dem folgenden describe-scheduled-actionsBefehl eine Liste der geplanten Aktionen für den angegebenen Dienst-Namespace ab.

   Linux, macOS oder Unix

   ```
   aws application-autoscaling describe-scheduled-actions \
     --service-namespace dynamodb
   ```

   Windows

   ```
   aws application-autoscaling describe-scheduled-actions --service-namespace dynamodb
   ```

   Es folgt eine Beispielausgabe.

   ```
   {
       "ScheduledActions": [
           {
               "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
               "Schedule": "at(2019-05-20T18:35:00)",
               "ResourceId": "table/TestTable",
               "CreationTime": 1561571888.361,
               "ScheduledActionARN": "arn:aws:autoscaling:us-
   east-1:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
   dynamodb/table/TestTable:scheduledActionName/my-first-scheduled-action",
               "ScalableTargetAction": {
                   "MinCapacity": 15,
                   "MaxCapacity": 20
               },
               "ScheduledActionName": "my-first-scheduled-action",
               "ServiceNamespace": "dynamodb"
           },
           {
               "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
               "Schedule": "at(2019-05-20T18:40:00)",
   ```

```
            "ResourceId": "table/TestTable",
            "CreationTime": 1561571946.021,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
east-1:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/TestTable:scheduledActionName/my-second-scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 5,
                "MaxCapacity": 10
            },
            "ScheduledActionName": "my-second-scheduled-action",
            "ServiceNamespace": "dynamodb"
        }
    ]
}
```

Schritt 3: Ansicht der Skalierungsaktivitäten

In diesem Schritt zeigen Sie die Skalierungsaktivitäten an, die durch die geplanten Aktionen ausgelöst wurden, und überprüfen, ob DynamoDB die Schreibkapazität der Tabelle geändert hat.

Ansehen der Skalierungsaktivitäten

1. Warten Sie auf die von Ihnen gewählte Zeit und überprüfen Sie mithilfe des folgenden describe-scaling-activitiesBefehls, ob Ihre geplanten Aktionen funktionieren.

   Linux, macOS oder Unix

   ```
   aws application-autoscaling describe-scaling-activities \
     --service-namespace dynamodb
   ```

   Windows

   ```
   aws application-autoscaling describe-scaling-activities --service-
   namespace dynamodb
   ```

   Im Folgenden finden Sie eine Beispielausgabe für die erste geplante Aktion, während die geplante Aktion ausgeführt wird.

   Skalierungsaktivitäten werden nach Erstellungsdatum angeordnet, wobei die neuesten Skalierungsaktivitäten zuerst zurückgegeben werden.

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 15.",
            "ResourceId": "table/TestTable",
            "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
            "StartTime": 1561574108.904,
            "ServiceNamespace": "dynamodb",
            "Cause": "minimum capacity was set to 15",
            "StatusMessage": "Successfully set write capacity units to 15. Waiting
 for change to be fulfilled by dynamodb.",
            "StatusCode": "InProgress"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 15 and max capacity to 20",
            "ResourceId": "table/TestTable",
            "ActivityId": "3250fd06-6940-4e8e-bb1f-d494db7554d2",
            "StartTime": 1561574108.512,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-first-scheduled-action was
 triggered",
            "StatusMessage": "Successfully set min capacity to 15 and max capacity
 to 20",
            "StatusCode": "Successful"
        }
    ]
}
```

Im Folgenden finden Sie eine Beispielausgabe, nachdem beide geplanten Aktionen ausgeführt wurden.

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 10.",
            "ResourceId": "table/TestTable",
            "ActivityId": "4d1308c0-bbcf-4514-a673-b0220ae38547",
            "StartTime": 1561574415.086,
            "ServiceNamespace": "dynamodb",
```

```
            "EndTime": 1561574449.51,
            "Cause": "maximum capacity was set to 10",
            "StatusMessage": "Successfully set write capacity units to 10. Change
successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 5 and max capacity to 10",
            "ResourceId": "table/TestTable",
            "ActivityId": "f2b7847b-721d-4e01-8ef0-0c8d3bacc1c7",
            "StartTime": 1561574414.644,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-second-scheduled-action was
triggered",
            "StatusMessage": "Successfully set min capacity to 5 and max capacity
to 10",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 15.",
            "ResourceId": "table/TestTable",
            "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
            "StartTime": 1561574108.904,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574140.255,
            "Cause": "minimum capacity was set to 15",
            "StatusMessage": "Successfully set write capacity units to 15. Change
successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 15 and max capacity to 20",
            "ResourceId": "table/TestTable",
            "ActivityId": "3250fd06-6940-4e8e-bb1f-d494db7554d2",
            "StartTime": 1561574108.512,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-first-scheduled-action was
triggered",
            "StatusMessage": "Successfully set min capacity to 15 and max capacity
to 20",
            "StatusCode": "Successful"
```

```
        }
    ]
}
```

2. Nachdem Sie die geplanten Aktionen erfolgreich ausgeführt haben, öffnen Sie die DynamoDB-Konsole und wählen die Tabelle aus, mit der Sie arbeiten möchten. Zeigen Sie die Schreibkapazitätseinheiten auf der Registerkarte Kapazität an. Nach Ausführung der zweiten Skalierungsaktion sollte die Zahl der Schreibkapazitätseinheiten von 15 auf 10 skaliert worden sein.

   Sie können auch die aktuelle Schreibkapazität der Tabelle mit dem DynamoDB-Befehl describe-table überprüfen. Fügen Sie die Option `--query` ein, um die Ausgabe zu filtern. Weitere Informationen zu den Ausgabefilterfunktionen von finden Sie unter Steuern der Befehlsausgabe von AWS CLI im AWS Command Line Interface Benutzerhandbuch. AWS CLI

   Linux, macOS oder Unix

   ```
   aws dynamodb describe-table --table-name TestTable \
     --query 'Table.[TableName,TableStatus,ProvisionedThroughput]'
   ```

   Windows

   ```
   aws dynamodb describe-table --table-name TestTable --query "Table.
   [TableName,TableStatus,ProvisionedThroughput]"
   ```

   Es folgt eine Beispielausgabe.

   ```
   [
       "TestTable",
       "ACTIVE",
       {
           "NumberOfDecreasesToday": 1,
           "WriteCapacityUnits": 10,
           "LastIncreaseDateTime": 1561574133.264,
           "ReadCapacityUnits": 5,
           "LastDecreaseDateTime": 1561574435.607
       }
   ]
   ```

Schritt 4: Nächste Schritte

Wenn Sie versuchen möchten, sowohl mit geplanter Skalierung als auch mit einer Skalierungsrichtlinie zu skalieren, führen Sie die Schritte unter Tutorial: Auto Scaling zur Bewältigung eines hohen Workloads konfigurieren aus.

Schritt 5: Bereinigen

Nach Abschluss der Einstiegsübungen können Sie die zugehörigen Ressourcen wie folgt bereinigen.

So löschen Sie die geplanten Aktionen

Mit dem folgenden delete-scheduled-actionBefehl wird eine angegebene geplante Aktion gelöscht. Sie können diesen Schritt überspringen, wenn Sie die geplante Aktion in der Zukunft verwenden möchten.

Linux, macOS oder Unix

```
aws application-autoscaling delete-scheduled-action \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:WriteCapacityUnits \
  --resource-id table/TestTable \
  --scheduled-action-name my-second-scheduled-action
```

Windows

```
aws application-autoscaling delete-scheduled-action --service-namespace dynamodb --
scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/TestTable --
scheduled-action-name my-second-scheduled-action
```

So melden Sie das skalierbare Ziel ab

Verwenden Sie den folgenden deregister-scalable-targetBefehl, um die Registrierung des skalierbaren Ziels aufzuheben. Mit diesem Befehl werden alle Skalierungsrichtlinien, die Sie erstellt haben, und alle geplanten Aktionen, die Sie noch nicht gelöscht haben, gelöscht. Sie können diesen Schritt überspringen, wenn das skalierbare Ziel für eine zukünftige Verwendung registriert bleiben soll.

Linux, macOS oder Unix

```
aws application-autoscaling deregister-scalable-target \
```

```
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:WriteCapacityUnits \
  --resource-id table/TestTable
```

Windows

```
aws application-autoscaling deregister-scalable-target --service-namespace dynamodb --
scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/TestTable
```

So löschen Sie die DynamoDB-Tabelle

Verwenden Sie den folgenden delete-table-Befehl, um die Tabelle zu löschen, die Sie in diesem Lernprogramm verwendet haben. Sie können diesen Schritt überspringen, wenn Sie die Tabelle für die spätere Verwendung behalten möchten.

Linux, macOS oder Unix

```
aws dynamodb delete-table --table-name TestTable
```

Windows

```
aws dynamodb delete-table --table-name TestTable
```

Skalierungsrichtlinien für die Ziel-Nachverfolgung

Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung skaliert Ihre Anwendung automatisch basierend auf einem Zielmetrikwert. Auf diese Weise kann Ihre Anwendung ohne manuelles Eingreifen eine optimale Leistung und Kosteneffizienz aufrechterhalten.

Bei der Ziel-Nachverfolgung wählen Sie eine Metrik und einen Zielwert aus, der die ideale durchschnittliche Auslastung oder den idealen Durchsatz für Ihre Anwendung darstellt. Application Auto Scaling erstellt und verwaltet die CloudWatch Alarme, die Skalierungsereignisse auslösen, wenn die Metrik vom Ziel abweicht. Dies ist vergleichbar mit der Art und Weise, wie ein Thermostat eine Zieltemperatur aufrechterhält.

Ein Beispiel: Angenommen, Sie verfügen über eine Anwendung, die derzeit in der Spot-Flotte ausgeführt wird, und die CPU-Auslastung der Flotte soll bei etwa 50 Prozent bleiben, wenn sich die Anwendungslast ändert. Auf diese Weise erlangen Sie zusätzliche Kapazität für Datenverkehrsspitzen, ohne übermäßig viele Ressourcen im Leerlauf zu verwalten.

Hierzu können Sie eine Skalierungsrichtlinie für die Zielverfolgung erstellen, die eine durchschnittliche CPU-Auslastung von 50 Prozent vorsieht. Dann skaliert Application Auto Scaling auf (erhöht die Kapazität), wenn die CPU 50 Prozent überschreitet, um die erhöhte Auslastung zu bewältigen. Wenn die CPU-Auslastung unter 50 Prozent sinkt, wird abskaliert (die Kapazität verringert), um die Kosten in Zeiten geringer Auslastung zu optimieren.

Richtlinien zur Zielverfolgung machen die manuelle Definition von CloudWatch Alarmen und Skalierungsanpassungen überflüssig. Application Auto Scaling erledigt dies automatisch auf der Grundlage des von Ihnen festgelegten Ziels.

Richtlinien für die Ziel-Nachverfolgung können entweder auf vordefinierten oder benutzerdefinierten Metriken basieren:

- Vordefinierte Metriken – von Application Auto Scaling bereitgestellte Metriken wie die durchschnittliche CPU-Auslastung oder die durchschnittliche Anzahl von Anfragen pro Ziel.
- Benutzerdefinierte Metriken — Sie können Metriken verwenden, um Metriken zu kombinieren, bestehende Metriken zu nutzen oder Ihre eigenen benutzerdefinierten Metriken zu CloudWatch verwenden, die unter veröffentlicht wurden.

Wählen Sie eine Metrik, die sich umgekehrt proportional zu einer Änderung der Kapazität Ihres skalierbaren Ziels ändert. Wenn Sie also die Kapazität verdoppeln, sinkt die Metrik um 50 Prozent. Auf diese Weise können die Metrikdaten genau proportionale Skalierungsereignisse auslösen.

Themen

- So funktioniert die Skalierung der Zielverfolgung
- Erstellen Sie eine Skalierungsrichtlinie für die Zielverfolgung mithilfe der AWS CLI
- Erstellen einer Skalierungsrichtlinie für Zielnachverfolgung für Application Auto Scaling mit Metrikberechnungen

So funktioniert die Skalierung der Zielverfolgung

In diesem Thema wird beschrieben, wie die Skalierung der Zielverfolgung funktioniert, und es werden die wichtigsten Elemente einer Skalierungsrichtlinie für die Zielverfolgung vorgestellt.

Inhalt

- Funktionsweise
- Auswahl von Metriken
- Definieren des Zielwerts
- Ruhephasen definieren
- Überlegungen
- Mehrere Skalierungsrichtlinien
- Häufig verwendete Befehle zur Erstellung, Verwaltung und Löschung von Skalierungsrichtlinien
- Zugehörige Ressourcen
- Einschränkungen

Funktionsweise

Um die Skalierung der Zielverfolgung zu verwenden, erstellen Sie eine Skalierungsrichtlinie für die Zielverfolgung und geben Folgendes an:

- CloudWatch Metrik — Eine zu verfolgende Metrik, z. B. die durchschnittliche CPU-Auslastung oder die durchschnittliche Anzahl von Anfragen pro Ziel.

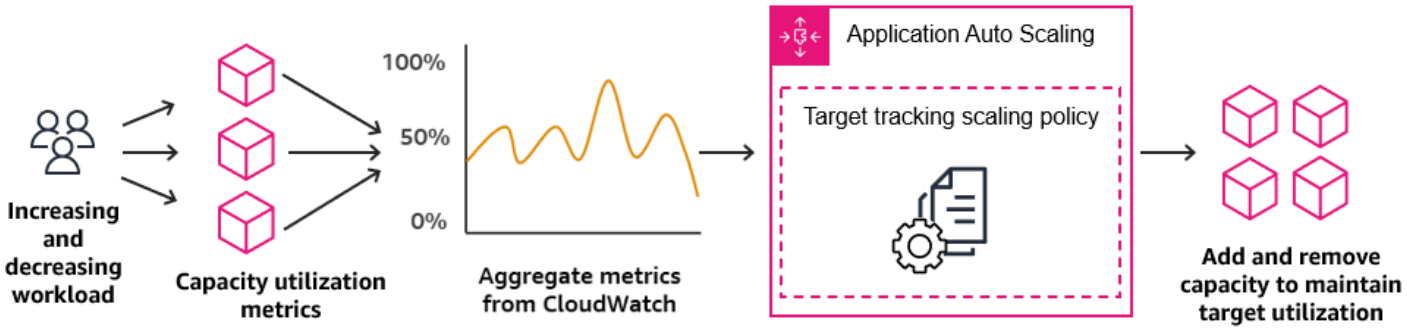
- Zielwert – der Zielwert für die Metrik, z. B. 50 Prozent CPU-Auslastung oder 1 000 Anfragen pro Ziel pro Minute.

Application Auto Scaling erstellt und verwaltet die CloudWatch Alarme, die die Skalierungsrichtlinie aufrufen, und berechnet die Skalierungsanpassung auf der Grundlage der Metrik und des Zielwerts. Es wird so viel Kapazität wie erforderlich hinzugefügt oder entfernt, damit die Metrik auf oder nahe an dem Zielwert gehalten wird.

Wenn die Metrik über dem Zielwert liegt, skaliert Application Auto Scaling auf, indem Kapazität hinzugefügt wird, um die Differenz zwischen dem Metrikwert und dem Zielwert zu verringern. Wenn die Metrik unter dem Zielwert liegt, skaliert Application Auto Scaling ab, indem Kapazität entfernt wird.

Zwischen den Skalierungsaktivitäten liegen Ruhephasen, um schnelle Kapazitätsschwankungen zu vermeiden. Sie können die Ruhephasen für Ihre Richtlinie optional konfigurieren.

Das folgende Diagramm zeigt einen Überblick über die Funktionsweise einer Zielverfolgungsrichtlinie, wenn die Einrichtung abgeschlossen ist.



Hinweis: Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung ist aggressiver beim Hinzufügen von Kapazität bei steigender Auslastung als beim Entfernen von Kapazität bei sinkender Auslastung. Wenn zum Beispiel die in der Richtlinie angegebene Metrik ihren Zielwert erreicht, geht die Richtlinie davon aus, dass Ihre Anwendung bereits stark belastet ist. Daher reagiert sie, indem sie so schnell wie möglich Kapazität proportional zum metrischen Wert hinzufügt. Je höher die Metrik, desto mehr Kapazität wird hinzugefügt.

Wenn die Metrik unter den Zielwert fällt, geht die Richtlinie davon aus, dass die Auslastung letztendlich wieder steigt. In diesem Fall verlangsamt sie die Skalierung, indem sie Kapazitäten nur dann entfernt, wenn die Auslastung einen Schwellenwert überschreitet, der weit genug unter dem Zielwert liegt (in der Regel um mehr als 10 %), damit die Auslastung als verlangsamt angesehen werden kann. Mit diesem vorsichtigeren Verhalten soll sichergestellt werden, dass Kapazitäten erst dann entfernt werden, wenn die Anwendung nicht mehr so häufig aufgerufen wird wie zuvor.

Auswahl von Metriken

Sie können Skalierungsrichtlinien zur Zielverfolgung mit vordefinierten oder benutzerdefinierten Metriken erstellen.

Wenn Sie eine Skalierungsrichtlinie zur Zielverfolgung mit einem vordefinierten Metriktyp erstellen, wählen Sie eine Metrik aus der Liste der vordefinierten Metriken in Vordefinierte Metriken für Skalierungsrichtlinien für die Zielverfolgung aus.

Berücksichtigen Sie die folgenden Aspekte, wenn Sie eine Metrik auswählen:

- Nicht alle benutzerdefinierten Metriken funktionieren für die Zielverfolgung. Die Metrik muss eine gültige Auslastungsmetrik sein und beschreiben, wie ausgelastet ein skalierbares Ziel ist. Der Metrikwert muss proportional zur Kapazität des skalierbaren Ziels steigen oder fallen, damit die metrischen Daten zur proportionalen Skalierung des skalierbaren Ziels verwendet werden können.
- Um die Metrik `ALBRequestCountPerTarget` zu verwenden, müssen Sie den Parameter `ResourceLabel` angeben, um die Zielgruppe zu identifizieren, die der Metrik zugeordnet ist.
- Wenn eine Metrik echte Werte von 0 ausgibt CloudWatch (z. B.`ALBRequestCountPerTarget`), kann Application Auto Scaling auf 0 skalieren, wenn über einen längeren Zeitraum kein Datenverkehr zu Ihrer Anwendung erfolgt. Damit Ihr skalierbares Ziel auf 0 skaliert wird, wenn keine Anforderungen an es weitergeleitet werden, muss die Mindestkapazität des skalierbaren Ziels auf 0 gesetzt werden.
- Anstatt neue Metriken zur Verwendung in Ihrer Skalierungsrichtlinie zu veröffentlichen, können Sie mit metrischer Mathematik bestehende Metriken kombinieren. Weitere Informationen finden Sie unter Erstellen einer Skalierungsrichtlinie für Zielnachverfolgung für Application Auto Scaling mit Metrikberechnungen.
- Informationen dazu, ob der von Ihnen verwendete Service die Angabe einer benutzerdefinierten Metrik in der Konsole des Service unterstützt, finden Sie in der Dokumentation für den betreffenden Service.
- Wir empfehlen, Metriken zu verwenden, die in einminütigen Intervallen verfügbar sind, damit Sie schneller auf Änderungen der Auslastung reagieren können. Die Zielverfolgung wertet Metriken für alle vordefinierten und benutzerdefinierten Metriken aus, die mit einer Granularität von einer Minute aggregiert sind, aber die zugrunde liegende Metrik veröffentlicht die Daten möglicherweise weniger häufig. So werden beispielsweise alle Amazon-EC2-Metriken standardmäßig in Fünf-Minuten-Intervallen gesendet, können aber auch auf eine Minute konfiguriert werden (bekannt als detaillierte Überwachung). Diese Entscheidung liegt bei den einzelnen Services. Die meisten versuchen, das kleinstmögliche Intervall zu verwenden.

Definieren des Zielwerts

Wenn Sie eine Skalierungsrichtlinie für die Zielverfolgung erstellen, müssen Sie einen Zielwert angeben. Der Zielwert stellt die optimale durchschnittliche Auslastung oder den idealen durchschnittlichen Durchsatz für Ihre Anwendung dar. Für eine kosteneffiziente Ressourcennutzung sollte der Zielwert auf einen möglichst hohen Wert mit einem angemessenen Puffer für unerwartete Datenverkehrserhöhungen festgelegt werden. Wenn Ihre Anwendung optimal für einen normalen Datenverkehrsfluss aufskaliert wird, sollte der tatsächliche Metrikwert dem Zielwert entsprechen oder knapp darunter liegen.

Wenn eine Skalierungsrichtlinie auf dem Durchsatz basiert, z. B. der Anzahl der Anfragen pro Ziel für einen Application Load Balancer, dem Netzwerk-E/A oder anderen Zählmetriken, stellt der Zielwert den optimalen durchschnittlichen Durchsatz einer einzelnen Einheit (z. B. eines einzelnen Ziels aus Ihrer Zielgruppe für Application Load Balancer) für einen Zeitraum von einer Minute dar.

Ruhephasen definieren

In Ihrer Skalierungsrichtlinie für die Zielverfolgung können Sie optional Ruhephasen definieren.

Eine Ruhephase ist die Zeitspanne, die die Skalierungsrichtlinie warten muss, bis eine vorherige Skalierungsaktivität wirksam wird.

Es gibt zwei Arten von Ruhephasen:

- Mit der Abkühlungsphase der Aufskalierung wird beabsichtigt, kontinuierlich (aber nicht übermäßig) zu skalieren. Nachdem Application Auto Scaling unter Verwendung einer Skalierungsrichtlinie erfolgreich aufskaliert wurde, wird die Berechnung der Ruhezeit gestartet. Eine Skalierungsrichtlinie erhöht die gewünschte Kapazität nicht erneut, es sei denn, es wird eine größere Aufskalierung ausgelöst oder die Ruhephase endet. Während die Scale-Out-Ruhephase wirksam ist, wird die durch die initiierende horizontale Skalierung nach oben (Scale-Out) hinzugefügte Kapazität als Teil der gewünschten Kapazität für die nächste horizontale Skalierung nach oben berechnet.
- Mit der Ruhephase für die Abskalierung ist beabsichtigt, die Abskalierung konservativ durchzuführen, um die Verfügbarkeit Ihrer Anwendung zu schützen, sodass Abskalierungsaktivitäten blockiert werden, bis die Ruhephase für die Abskalierung abgelaufen ist. Wenn jedoch ein anderer Alarm während der Abkühlphase nach einer Abskalier-Aktivität eine Aufskalier-Aktivität auslöst, wird das Ziel durch Application Auto Scaling sofort abskaliert. In diesem Fall wird die Ruhephase für die Abskalierung angehalten und nicht abgeschlossen.

Jede Ruhephase wird in Sekunden gemessen und gilt nur für Skalierungsrichtlinien-bezogene Skalierungen. Wenn eine geplante Aktion während einer Ruhephase zum geplanten Zeitpunkt beginnt, kann sie umgehend eine Skalierung auslösen, ohne das Ablaufen der Ruhephase abzuwarten.

Sie können mit den Standardwerten beginnen, die später optimiert werden können. So kann es beispielsweise erforderlich sein, die Ruhephase zu verlängern, um zu verhindern, dass Ihre Skalierungsrichtlinie zur Ziel-Nachverfolgung zu aggressiv auf Änderungen reagiert, die über kurze Zeiträume auftreten.

Standardwerte

Application Auto Scaling bietet einen Standardwert von 600 für ElastiCache Replikationsgruppen und einen Standardwert von 300 für die folgenden skalierbaren Ziele:

- AppStream 2.0-Flotten
- Aurora-DB-Cluster
- ECS-Services
- Neptune-Cluster
- SageMaker Endpunkt-Varianten
- SageMaker Inferenzkomponenten
- SageMaker Serverlos bereitgestellte Parallelität
- Spot Flotten
- Benutzerdefinierte Ressourcen

Für alle anderen skalierbaren Ziele ist der Standardwert 0 oder NULL:

- Amazon Comprehend-Dokumentklassifizierungs- und Entitätserkennungs-Endpunkte
- DynamoDB-Tabellen und globale sekundäre Indizes
- Amazon Keyspace-Tabellen
- Parallelität per Lambda
- Amazon MSK-Broker-Speicher

NULL-Werte werden genauso behandelt wie Nullwerte, wenn Application Auto Scaling die Ruhephase auswertet.

Sie können jeden der Standardwerte, einschließlich NULL-Werte, aktualisieren, um Ihre eigenen Ruhephasen festzulegen.

Überlegungen

Bei der Arbeit mit Skalierungsrichtlinien für die Zielverfolgung ist Folgendes zu beachten:

- Erstellen, bearbeiten oder löschen Sie keine CloudWatch Alarme, die mit einer Skalierungsrichtlinie für die Zielverfolgung verwendet werden. Application Auto Scaling erstellt und verwaltet die CloudWatch Alarme, die mit Ihren Skalierungsrichtlinien für die Zielverfolgung verknüpft sind, und löscht sie, wenn sie nicht mehr benötigt werden.
- Wenn der Metrik Datenpunkte fehlen, führt dies dazu, dass der CloudWatch Alarmstatus auf `INSUFFICIENT_DATA` geändert wird. In diesem Fall kann Application Auto Scaling das skalierbare Ziel erst wieder skalieren, wenn neue Datenpunkte gefunden wurden. Informationen zum Erstellen von Alarmen bei unzureichenden Daten finden Sie unter CloudWatch-Alarmen überwachen.
- Wenn die Metrik konstruktionsbedingt nur spärlich gemeldet wird, kann metrische Mathematik hilfreich sein. Um beispielsweise die neuesten Werte zu verwenden, verwenden Sie die Funktion `FILL(m1,REPEAT)`, wobei `m1` die Metrik ist.
- Möglicherweise werden Lücken zwischen den Datenpunkten für den Zielwert und die aktuelle Metrik angezeigt. Dies liegt daran, dass Application Auto Scaling immer konservativ vorgeht, indem sie auf- oder abrundet, wenn sie bestimmt, wie viel Kapazität hinzugefügt oder entfernt werden soll. Dadurch wird verhindert, dass zu wenig Kapazität hinzufügt oder zu viel Kapazität entfernt wird. Bei einem skalierbaren Ziel mit geringer Kapazität können die tatsächlichen metrischen Datenpunkte jedoch scheinbar weit vom Zielwert entfernt sein.

  Bei einem skalierbaren Ziel mit größerer Kapazität führt das Hinzufügen oder Entfernen von Kapazität zu einem geringeren Abstand zwischen dem Zielwert und den tatsächlichen metrischen Datenpunkten.
- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung geht davon aus, dass sie eine horizontale Skalierung nach oben vornehmen soll, wenn die angegebene Metrik über dem Zielwert liegt. Sie können keine Skalierungsrichtlinie für die Ziel-Nachverfolgung verwenden, um eine horizontale Skalierung nach oben vorzunehmen, wenn die angegebene Metrik unter dem Zielwert liegt.

Mehrere Skalierungsrichtlinien

Sie können mehrere Skalierungsrichtlinien für die Ziel-Nachverfolgung für ein skalierbares Ziel besitzen, vorausgesetzt, dass diese alle verschiedene Metriken verwenden. Die Absicht von

Application Auto Scaling ist es, der Verfügbarkeit immer Vorrang einzuräumen. Daher unterscheidet sich das Verhalten von Application Auto Scaling, je nachdem, ob die Ziel-Tracking-Richtlinien für die Skalierung nach außen oder nach innen bereit sind. Sofern Richtlinien für die Ziel-Nachverfolgung für die horizontale Skalierung nach oben bereit sind, findet eine horizontale Skalierung des skalierbaren Ziels nach oben statt. Eine horizontale Skalierung nach unten wird jedoch nur vorgenommen, wenn alle Richtlinien für die Ziel-Nachverfolgung (mit aktivierter horizontaler Skalierung nach unten) zur horizontalen Skalierung nach unten bereit sind.

Wenn mehrere Richtlinien das skalierbare Ziel gleichzeitig zum Auf- oder Abskalieren anweisen, erfolgt die Skalierung durch Application Auto Scaling auf Grundlage der Richtlinie, die die größte Kapazität für die Ab- und Aufskalierung bereitstellt. Das bietet Ihnen eine größere Flexibilität für verschiedene Szenarien und stellt sicher, dass immer ausreichend Kapazität zur Verarbeitung Ihrer Workloads vorhanden ist.

Sie können den Abskalierungsteil einer Skalierungsrichtlinie für die Zielnachverfolgung deaktivieren, um eine andere Methode für die Abskalierung zu verwenden als für die Aufskalierung. Sie können z. B. eine Schrittskalierungsrichtlinie für die Skalierung nach unten verwenden, während Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung für die Skalierung nach oben verwenden.

Sie sollten bei der Verwendung von Zielverfolgungs-Skalierungsrichtlinien mit Schrittskalierungsrichtlinien jedoch vorsichtig sein, da Konflikte zwischen diesen Richtlinien zu unerwünschtem Verhalten führen können. Wenn beispielsweise die Schrittskalierungsrichtlinie eine Abwärtsskalierungsaktivität initiiert, bevor die Zielverfolgungsrichtlinie abwärts skaliert werden kann, wird die Abwärtsskalierungsaktivität nicht blockiert. Nach Abschluss der herunterskalierenden Aktivität könnte die Zielverfolgungsrichtlinie das skalierbare Ziel anweisen, erneut zu skalieren.

Für zyklisch anfallende Verarbeitungslasten haben Sie außerdem die Möglichkeit, Kapazitätsänderungen in einem Zeitplan mithilfe der geplanten Skalierung zu automatisieren. Für jede geplante Aktion können ein neuer Kapazitätsmindestwert und ein neuer Kapazitätshöchstwert festgelegt werden. Diese Werte bilden die Grenzen der Skalierungsrichtlinie. Die Kombination aus geplanter Skalierung und Skalierung zur Zielnachverfolgung kann dazu beitragen, die Auswirkungen eines starken Anstiegs des Auslastungsgrades zu verringern, wenn die Kapazität sofort benötigt wird.

Häufig verwendete Befehle zur Erstellung, Verwaltung und Löschung von Skalierungsrichtlinien

Zu den häufig verwendeten Befehlen für die Arbeit mit Skalierungsrichtlinien gehören:

- register-scalable-targetum Ressourcen als skalierbare Ziele zu registrieren AWS oder anzupassen (eine Ressource, die Application Auto Scaling skalieren kann) und die Skalierung auszusetzen und wieder aufzunehmen.
- put-scaling-policyum Skalierungsrichtlinien für ein vorhandenes skalierbares Ziel hinzuzufügen oder zu ändern.
- describe-scaling-activitiesum Informationen über Skalierungsaktivitäten in einer AWS Region zurückzugeben.
- describe-scaling-policiesum Informationen über Skalierungsrichtlinien in einer AWS Region zurückzugeben.
- delete-scaling-policyum eine Skalierungsrichtlinie zu löschen.

Zugehörige Ressourcen

Informationen zum Erstellen von Skalierungsrichtlinien für die Ziel-Nachverfolgung für Auto-Scaling-Gruppen finden Sie unter Skalierungsrichtlinien für die Ziel-Nachverfolgung für Amazon EC2 Auto Scaling im Benutzerhandbuch für Amazon EC2 Auto Scaling.

Einschränkungen

Bei der Verwendung von Zielverfolgungs-Skalierungsrichtlinien gibt es folgende Einschränkungen:

- Das skalierbare Ziel kann nicht ein Amazon EMR-Cluster sein. Zielverfolgungs-Skalierungsrichtlinien werden für Amazon EMR nicht unterstützt.
- Wenn ein Amazon MSK-Cluster das skalierbare Ziel ist, ist scale in deaktiviert und kann nicht aktiviert werden.
- Sie können die `RegisterScalableTarget` oder `PutScalingPolicy` API-Operationen nicht verwenden, um einen AWS Auto Scaling Skalierungsplan zu aktualisieren. Informationen zur Verwendung von Skalierungsplänen finden Sie in der Dokumentation AWS Auto Scaling.
- Der Konsolenzugriff zum Anzeigen, Hinzufügen, Aktualisieren oder Entfernen von Skalierungsrichtlinien für die Ziel-Nachverfolgung auf skalierbaren Ressourcen hängt von der verwendeten Ressource ab. Weitere Informationen finden Sie unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

Erstellen Sie eine Skalierungsrichtlinie für die Zielverfolgung mithilfe der AWS CLI

Sie können eine Skalierungsrichtlinie für die Zielverfolgung für Application Auto Scaling erstellen, indem Sie die AWS CLI für die folgenden Konfigurationsaufgaben verwenden.

1. Registrieren eines skalierbaren Ziels
2. Fügen Sie dem skalierbaren Ziel eine Skalierungsrichtlinie für die Ziel-Nachverfolgung hinzu.

Der Kürze halber zeigen die Beispiele in diesem Thema CLI-Befehle für eine Amazon EC2 Spot Fleet. Um ein anderes skalierbares Ziel anzugeben, geben Sie seinen Namespace in `--service-namespace`, seine skalierbare Dimension in `--scalable-dimension` und seine Ressourcen-ID in `--resource-id` an. Weitere Informationen und Beispiele für die einzelnen Services finden Sie in den Themen unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

Denken Sie bei der Verwendung von daran AWS CLI, dass Ihre Befehle in der für Ihr Profil AWS-Region konfigurierten Version ausgeführt werden. Wenn Sie die Befehle in einer anderen Region ausführen möchten, ändern Sie entweder die Standardregion für Ihr Profil, oder verwenden Sie den `--region`-Parameter mit dem Befehl.

Inhalt

- Registrieren eines skalierbaren Ziels
- Erstellen einer Zielverfolgungs-Skalierungsrichtlinie
- Beschreiben Sie die Zielverfolgungs-Skalierungsrichtlinien
- Löschen einer Zielverfolgungs-Skalierungsrichtlinie

Registrieren eines skalierbaren Ziels

Wenn Sie dies noch nicht getan haben, registrieren Sie das skalierbare Ziel. Verwenden Sie den register-scalable-targetBefehl, um eine bestimmte Ressource im Zieldienst als skalierbares Ziel zu registrieren. Im folgenden Beispiel wird eine Spot Fleet-Anfrage mit Application Auto Scaling registriert. Application Auto Scaling kann die Anzahl der Instanzen in der Spot Fleet auf ein Minimum von 2 und ein Maximum von 10 Instanzen skalieren. Ersetzen Sie jedes *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Linux, macOS oder Unix

```
aws application-autoscaling register-scalable-target --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --min-capacity 2 --max-capacity 10
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace ec2 --
scalable-dimension ec2:spot-fleet-request:TargetCapacity --resource-id spot-fleet-
request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --min-capacity 2 --max-capacity 10
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Erstellen einer Zielverfolgungs-Skalierungsrichtlinie

Um eine Skalierungsrichtlinie für die Zielverfolgung zu erstellen, können Sie die folgenden Beispiele verwenden, um Ihnen den Einstieg zu erleichtern.

So erstellen Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung

1. Verwenden Sie den folgenden `cat` Befehl, um einen Zielwert für Ihre Skalierungsrichtlinie und eine vordefinierte Metrikspezifikation in einer JSON-Datei mit dem Namen `config.json` in Ihrem Home-Verzeichnis zu speichern. Im Folgenden finden Sie ein Beispiel für eine Konfiguration zur Zielverfolgung, mit der die durchschnittliche CPU-Auslastung bei 50 Prozent gehalten wird.

   ```
   $ cat ~/config.json
   {
     "TargetValue": 50.0,
     "PredefinedMetricSpecification":
       {
         "PredefinedMetricType": "EC2SpotFleetRequestAverageCPUUtilization"
       }
   }
   ```

Weitere Informationen finden Sie PredefinedMetricSpecificationin der API-Referenz für Application Auto Scaling.

Alternativ können Sie eine benutzerdefinierte Metrik für die Skalierung verwenden, indem Sie eine benutzerdefinierte Metrikspezifikation erstellen und Werte für jeden Parameter von hinzufügen CloudWatch. Im Folgenden finden Sie ein Beispiel für eine Konfiguration zur Zielverfolgung, bei der die durchschnittliche Auslastung der angegebenen Metrik bei 100 belassen wird.

```
$ cat ~/config.json
{
   "TargetValue": 100.0,
   "CustomizedMetricSpecification":{
      "MetricName": "MyUtilizationMetric",
      "Namespace": "MyNamespace",
      "Dimensions": [
         {
            "Name": "MyOptionalMetricDimensionName",
            "Value": "MyOptionalMetricDimensionValue"
         }
      ],
      "Statistic": "Average",
      "Unit": "Percent"
   }
}
```

Weitere Informationen finden Sie CustomizedMetricSpecificationin der API-Referenz für Application Auto Scaling.

2. Verwenden Sie den folgenden put-scaling-policyBefehl zusammen mit der von Ihnen erstellten `config.json` Datei, um eine Skalierungsrichtlinie mit dem Namen zu erstellen`cpu50-target-tracking-scaling-policy`.

   Linux, macOS oder Unix

```
aws application-autoscaling put-scaling-policy --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --policy-name cpu50-target-tracking-scaling-policy --policy-type
 TargetTrackingScaling \
```

```
  --target-tracking-scaling-policy-configuration file://config.json
```

Windows

```
aws application-autoscaling put-scaling-policy --service-namespace ec2 --scalable-
dimension ec2:spot-fleet-request:TargetCapacity --resource-id spot-fleet-request/
sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --policy-name cpu50-target-tracking-
scaling-policy --policy-type TargetTrackingScaling --target-tracking-scaling-
policy-configuration file://config.json
```

Bei Erfolg gibt dieser Befehl die ARNs und Namen der beiden CloudWatch Alarme zurück, die in Ihrem Namen erstellt wurden.

```
{
    "PolicyARN": "arn:aws:autoscaling:region:account-
id:scalingPolicy:policy-id:resource/ec2/spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE:policyName/cpu50-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
            "AlarmName": "TargetTracking-spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE-AlarmLow-1b437334-
d19b-4a63-a812-6c67aaf2910d",
            "AlarmName": "TargetTracking-spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
        }
    ]
}
```

Beschreiben Sie die Zielverfolgungs-Skalierungsrichtlinien

Mit dem folgenden describe-scaling-policiesBefehl können Sie alle Skalierungsrichtlinien für den angegebenen Service-Namespace beschreiben.

```
aws application-autoscaling describe-scaling-policies --service-namespace ec2
```

Sie können die Ergebnisse mithilfe des Parameters `--query` filtern, um nur die Skalierungsrichtlinien zur Ziel-Nachverfolgung zu erhalten. Weitere Informationen über die Syntax von `query`, finden Sie unter Steuerung der Befehlsausgabe vom AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scaling-policies --service-namespace ec2 \
  --query 'ScalingPolicies[?PolicyType==`TargetTrackingScaling`]'
```

Windows

```
aws application-autoscaling describe-scaling-policies --service-namespace ec2 --query
 "ScalingPolicies[?PolicyType==`TargetTrackingScaling`]"
```

Es folgt eine Beispielausgabe.

```
[
    {
        "PolicyARN": "PolicyARN",
        "TargetTrackingScalingPolicyConfiguration": {
            "PredefinedMetricSpecification": {
                "PredefinedMetricType": "EC2SpotFleetRequestAverageCPUUtilization"
            },
            "TargetValue": 50.0
        },
        "PolicyName": "cpu50-target-tracking-scaling-policy",
        "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
        "ServiceNamespace": "ec2",
        "PolicyType": "TargetTrackingScaling",
        "ResourceId": "spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
        "Alarms": [
            {
                "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
                "AlarmName": "TargetTracking-spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca"
            },
```

```
            {
                "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE-AlarmLow-1b437334-
d19b-4a63-a812-6c67aaf2910d",
                "AlarmName": "TargetTracking-spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
            }
        ],
        "CreationTime": 1515021724.807
    }
]
```

Löschen einer Zielverfolgungs-Skalierungsrichtlinie

Wenn Sie mit einer Skalierungsrichtlinie für die Zielverfolgung fertig sind, können Sie sie mit dem delete-scaling-policyBefehl löschen.

Mit dem folgenden Befehl wird die angegebene Skalierungsrichtlinie zur Ziel-Nachverfolgung für die angegebene Spot-Flottenanforderung gelöscht. Außerdem werden die CloudWatch Alarme gelöscht, die Application Auto Scaling in Ihrem Namen erstellt hat.

Linux, macOS oder Unix

```
aws application-autoscaling delete-scaling-policy --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --policy-name cpu50-target-tracking-scaling-policy
```

Windows

```
aws application-autoscaling delete-scaling-policy --service-namespace ec2 --scalable-
dimension ec2:spot-fleet-request:TargetCapacity --resource-id spot-fleet-request/
sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --policy-name cpu50-target-tracking-scaling-
policy
```

Erstellen einer Skalierungsrichtlinie für Zielnachverfolgung für Application Auto Scaling mit Metrikberechnungen

Mithilfe von metrischer Mathematik können Sie mehrere CloudWatch Metriken abfragen und mathematische Ausdrücke verwenden, um neue Zeitreihen auf der Grundlage dieser Metriken zu

erstellen. Sie können die resultierenden Zeitreihen in der CloudWatch Konsole visualisieren und sie zu Dashboards hinzufügen. Weitere Informationen zur metrischen Mathematik finden Sie unter Verwenden von metrischer Mathematik im CloudWatch Amazon-Benutzerhandbuch.

Für metrische mathematische Ausdrücke gelten folgende Überlegungen:

- Sie können jede verfügbare CloudWatch Metrik abfragen. Jede Metrik ist eine eindeutige Kombination aus Metrikname, Namespace und null oder mehr Dimensionen.
- Sie können einen beliebigen arithmetischen Operator (+ - */^), jede statistische Funktion (wie AVG oder SUM) oder eine andere Funktion verwenden, die diese CloudWatch Funktion unterstützt.
- Sie können sowohl Metriken als auch die Ergebnisse anderer mathematischer Ausdrücke in den Formeln des mathematischen Ausdrucks verwenden.
- Alle Ausdrücke, die in einer metrischen Spezifikation verwendet werden, müssen letztendlich eine einzige Zeitreihe ergeben.
- Sie können überprüfen, ob ein metrischer mathematischer Ausdruck gültig ist, indem Sie die CloudWatch Konsole oder die CloudWatch GetMetricDataAPI verwenden.

Themen

- Beispiel: Amazon-SQS-Warteschlangenrückstand pro Aufgabe
- Einschränkungen

Beispiel: Amazon-SQS-Warteschlangenrückstand pro Aufgabe

Um den Rückstand der Amazon SQS-Warteschlange pro Aufgabe zu berechnen, nehmen Sie die ungefähre Anzahl der Nachrichten, die für den Abruf aus der Warteschlange zur Verfügung stehen, und teilen Sie diese Zahl durch die Anzahl der im Service laufenden Amazon ECS-Aufgaben. Weitere Informationen finden Sie im AWS Compute-Blog unter Amazon Elastic Container Service (ECS) Auto Scaling using custom metrics.

Die Logik für den Ausdruck lautet wie folgt:

```
sum of (number of messages in the queue)/(number of tasks that are
currently in the RUNNING state)
```

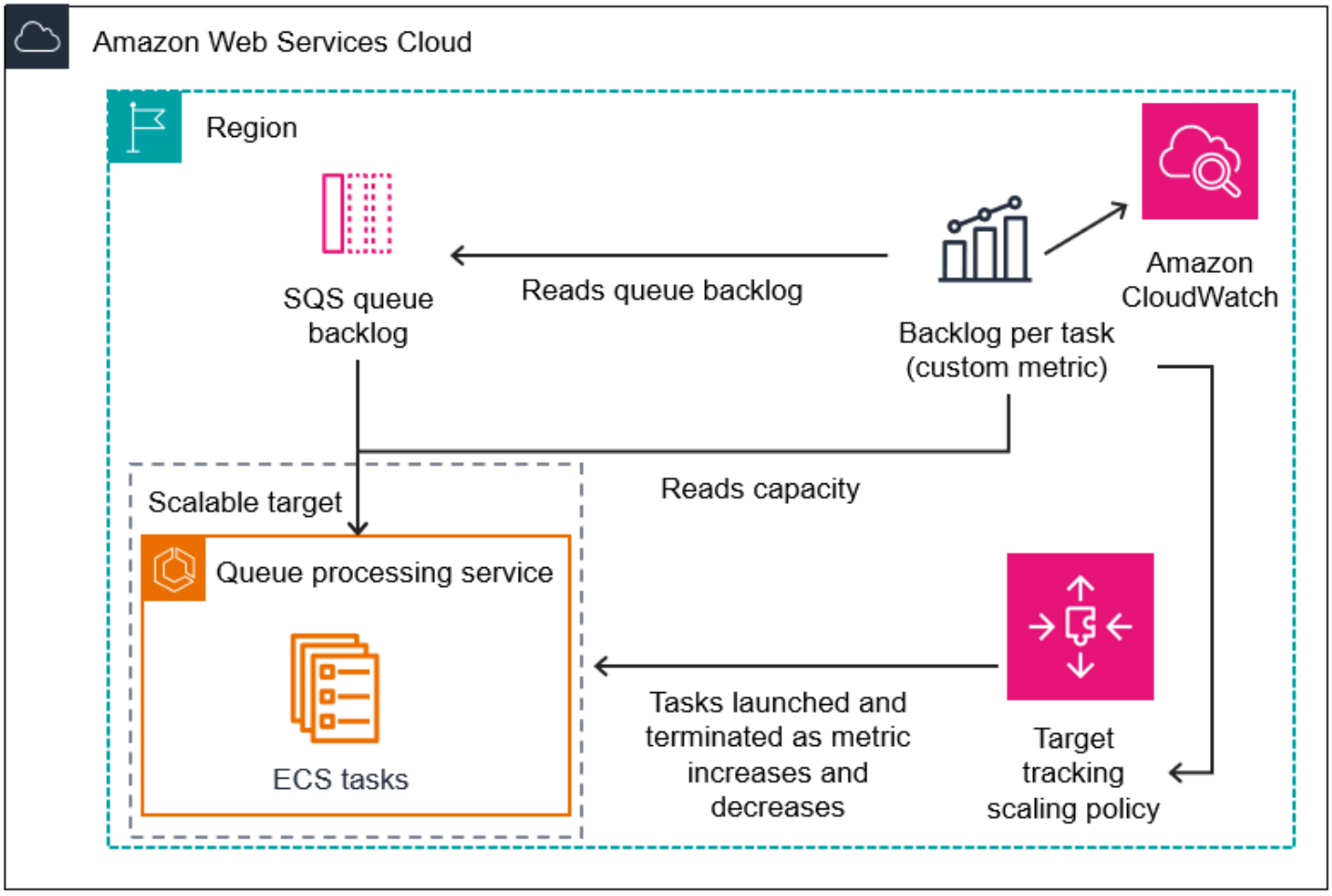
Dann lauten Ihre CloudWatch Metrikinformationen wie folgt.

ID	CloudWatch metrisch	Statistik	Intervall
m1	ApproximateNumberOfMessagesVisible	Summe	1 Minute
m2	RunningTaskCount	Durchschnitt	1 Minute

ID und Ausdruck Ihrer Metrikberechnung lauten wie folgt.

ID	Expression
e1	`(m1)/(m2)`

Das folgende Diagramm veranschaulicht die Architektur dieser Metrik:



So erstellen Sie mithilfe dieser Metrikberechnung eine Skalierungsrichtlinie für die Zielnachverfolgung (AWS CLI)

1. Speichern Sie den metrischen mathematischen Ausdruck als Teil einer benutzerdefinierten Metrikspezifikation in einer JSON-Datei namens `config.json`.

   Das folgende Beispiel hilft Ihnen bei den ersten Schritten. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

```
{
    "CustomizedMetricSpecification": {
        "Metrics": [
            {
                "Label": "Get the queue size (the number of messages waiting to be
 processed)",
                "Id": "m1",
                "MetricStat": {
                    "Metric": {
                        "MetricName": "ApproximateNumberOfMessagesVisible",
                        "Namespace": "AWS/SQS",
                        "Dimensions": [
                            {
                                "Name": "QueueName",
                                "Value": "my-queue"
                            }
                        ]
                    },
                    "Stat": "Sum"
                },
                "ReturnData": false
            },
            {
                "Label": "Get the ECS running task count (the number of currently
 running tasks)",
                "Id": "m2",
                "MetricStat": {
                    "Metric": {
                        "MetricName": "RunningTaskCount",
                        "Namespace": "ECS/ContainerInsights",
                        "Dimensions": [
                            {
                                "Name": "ClusterName",
                                "Value": "my-cluster"
```

```
                            },
                            {
                                "Name": "ServiceName",
                                "Value": "my-service"
                            }
                        ]
                    },
                    "Stat": "Average"
                },
                "ReturnData": false
            },
            {
                "Label": "Calculate the backlog per instance",
                "Id": "e1",
                "Expression": "m1 / m2",
                "ReturnData": true
            }
        ]
    },
    "TargetValue": 100
}
```

Weitere Informationen finden Sie TargetTrackingScalingPolicyConfigurationin der API-Referenz für Application Auto Scaling.

> **Note**
>
> Im Folgenden finden Sie einige zusätzliche Ressourcen, die Ihnen bei der Suche nach Metriknamen, Namespaces, Dimensionen und Statistiken für Metriken helfen können: CloudWatch
>
> - Informationen zu den verfügbaren Metriken für AWS Services finden Sie im CloudWatch Amazon-Benutzerhandbuch unter AWS Services, die CloudWatch Metriken veröffentlichen.
> - Den genauen Metriknamen, den Namespace und die Dimensionen (falls zutreffend) für eine CloudWatch Metrik mit dem finden Sie unter AWS CLIlist-metrics.

2. Um diese Richtlinie zu erstellen, führen Sie den put-scaling-policyBefehl mit der JSON-Datei als Eingabe aus, wie im folgenden Beispiel gezeigt.

```
aws application-autoscaling put-scaling-policy --policy-name sqs-backlog-target-tracking-scaling-policy \
  --service-namespace ecs --scalable-dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-service \
  --policy-type TargetTrackingScaling --target-tracking-scaling-policy-configuration file://config.json
```

Bei Erfolg gibt dieser Befehl den Amazon-Ressourcennamen (ARN) der Richtlinie und die ARNs der beiden in Ihrem Namen erstellten CloudWatch Alarme zurück.

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:
 8784a896-b2ba-47a1-b08c-27301cc499a1:resource/ecs/service/my-cluster/my-
service:policyName/sqs-backlog-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/my-cluster/my-service-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0",
            "AlarmName": "TargetTracking-service/my-cluster/my-service-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/my-cluster/my-service-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4",
            "AlarmName": "TargetTracking-service/my-cluster/my-service-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4"
        }
    ]
}
```

> **Note**
>
> Wenn dieser Befehl einen Fehler auslöst, stellen Sie sicher, dass Sie die AWS CLI lokale Version auf die neueste Version aktualisiert haben.

Einschränkungen

- Die maximale Größe der Anfrage ist 50 KB. Dies ist die gesamte Payload-Größe für die PutScalingPolicyAPI-Anfrage, wenn Sie metrische Mathematik in der Richtliniendefinition verwenden. Wenn Sie diese Grenze überschreiten, lehnt Application Auto Scaling die Anfrage ab.
- Die folgenden Dienste werden bei Verwendung der metrischen Mathematik mit Skalierungsrichtlinien für die Zielverfolgung nicht unterstützt:
  - Amazon Keyspaces (für Apache Cassandra)
  - DynamoDB
  - Amazon EMR
  - Amazon MSK
  - Amazon Neptune

Richtlinien zur schrittweisen Skalierung

Eine schrittweise Skalierungsrichtlinie skaliert die Kapazität Ihrer Anwendung in vordefinierten Schritten auf der Grundlage von CloudWatch Alarmen. Sie können separate Skalierungsrichtlinien definieren, um die Aufskalierung (Erhöhung der Kapazität) und die Abskalierung (Verringerung der Kapazität) zu handhaben, wenn ein Alarmschwellenwert überschritten wird.

Mit Richtlinien zur schrittweisen Skalierung erstellen und verwalten Sie die CloudWatch Alarme, die den Skalierungsprozess auslösen. Wenn ein Alarm ausgelöst wird, initiiert Application Auto Scaling die mit diesem Alarm verbundene Skalierungsrichtlinie.

Die Richtlinie zur schrittweisen Skalierung skaliert die Kapazität anhand einer Reihe von Anpassungen, die als schrittweise Anpassungen bezeichnet werden. Die Größe der Anpassung richtet sich nach dem Ausmaß der Alarmüberschreitung.

- Wenn der Verstoß den ersten Schwellenwert überschreitet, wendet Application Auto Scaling die erste schrittweise Anpassung an.
- Wenn der Verstoß den zweiten Schwellenwert überschreitet, wendet Application Auto Scaling die zweite schrittweise Anpassung an, und so weiter.

Auf diese Weise kann die Skalierungsrichtlinie sowohl auf kleinere als auch auf größere Änderungen der Alarmmetrik angemessen reagieren.

Die Richtlinie reagiert auch während einer laufenden Skalierungsaktivität auf weitere Alarmverstöße. Das bedeutet, dass Application Auto Scaling alle Alarmverstöße auswertet, sobald sie auftreten. Eine Ruhephase dient zum Schutz vor zu hoher Skalierung aufgrund mehrerer schnell aufeinanderfolgender Alarmverstöße.

Wie die Ziel-Nachverfolgung kann auch die schrittweise Skalierung dazu beitragen, die Kapazität Ihrer Anwendung automatisch zu skalieren, wenn sich der Datenverkehr ändert. Richtlinien für die Ziel-Nachverfolgung sind jedoch einfacher zu implementieren und zu verwalten, wenn eine stetige Skalierung erforderlich ist.

Sie können Richtlinien zur schrittweisen Skalierung mit den folgenden skalierbaren Zielen verwenden:

- AppStream 2.0-Flotten
- Aurora-DB-Cluster
- ECS-Services

- EMR-Cluster
- SageMaker Endpunkt-Varianten
- SageMaker Inferenzkomponenten
- SageMaker Serverlos bereitgestellte Parallelität
- Spot Flotten
- Benutzerdefinierte Ressourcen

Themen

- Wie funktioniert Step Scaling
- Eine Stufenskalierungsrichtlinie mit der AWS CLI erstellen

Wie funktioniert Step Scaling

In diesem Thema wird beschrieben, wie Step Scaling funktioniert, und es werden die wichtigsten Elemente einer Step Scaling-Richtlinie vorgestellt.

Inhalt

- Funktionsweise
- Schrittweise Anpassungen
- Skalierungsanpassungstypen
- Ruhephase
- Häufig verwendete Befehle zur Erstellung, Verwaltung und Löschung von Skalierungsrichtlinien
- Überlegungen
- Zugehörige Ressourcen
- Einschränkungen

Funktionsweise

Um Step Scaling zu verwenden, erstellen Sie einen CloudWatch Alarm, der eine Metrik für Ihr skalierbares Ziel überwacht. Sie definieren die Metrik, den Schwellenwert und die Anzahl der Bewertungszeiträume, die einen Alarmverstoß bestimmen. Außerdem erstellen Sie eine Richtlinie zur schrittweisen Skalierung, die definiert, wie die Kapazität skaliert werden soll, wenn der Alarmschwellenwert überschritten wird, und verknüpfen sie mit Ihrem skalierbaren Ziel.

Sie fügen die schrittweisen Anpassungen in der Richtlinie hinzu. Sie können verschiedene schrittweise Anpassungen basierend auf der Größe der Alarmüberschreitung definieren. Beispielsweise:

- Aufskalierung um 10 Kapazitätseinheiten, wenn die Alarmmetrik 60 Prozent erreicht
- Aufskalierung um 30 Kapazitätseinheiten, wenn die Alarmmetrik 75 Prozent erreicht
- Aufskalierung um 40 Kapazitätseinheiten, wenn die Alarmmetrik 85 Prozent erreicht

Wenn der Alarmschwellenwert für die angegebene Anzahl von Auswertungszeiträumen überschritten wird, wendet Application Auto Scaling die in der Richtlinie definierten schrittweisen Anpassungen an. Die Anpassungen können bei weiteren Überschreitungen des Alarms fortgesetzt werden, bis der Alarmstatus `OK` wieder erreicht ist.

Zwischen den Skalierungsaktivitäten liegen Ruhephasen, um schnelle Kapazitätsschwankungen zu vermeiden. Sie können die Ruhephasen für Ihre Richtlinie optional konfigurieren.

Schrittweise Anpassungen

Wenn Sie eine Richtlinie zur schrittweise Skalierung erstellen, geben Sie eine oder mehrere Stufenanpassungen an, die automatisch die Kapazität des Ziels dynamisch basierend auf der Größe der Alarmüberschreitung skalieren. Jede Schrittanpassung gibt Folgendes an:

- Eine Untergrenze für den Metrikwert
- Eine Obergrenze für den Metrikwert
- Den Skalierungswert basierend auf dem Skalierungsanpassungstyp

CloudWatch aggregiert metrische Datenpunkte auf der Grundlage der Statistik für die mit Ihrem CloudWatch Alarm verknüpfte Metrik. Wenn der Alarm ausgelöst wird, wird die entsprechende Skalierungsrichtlinie ausgelöst. Application Auto Scaling wendet den angegebenen Aggregationstyp auf die neuesten metrischen Datenpunkte von an CloudWatch (im Gegensatz zu den metrischen Rohdaten). Dieser aggregierte Metrikwert wird anschließend mit der Ober- und der Untergrenze verglichen, die durch die Schrittanpassungen definiert wurden. Dadurch wird ermittelt, welche Schrittanpassung auszuführen ist.

Sie geben die Ober- und Untergrenzen relativ zum Verletzungsschwellenwert an. Nehmen wir zum Beispiel an, Sie haben einen CloudWatch Alarm ausgelöst und eine Scale-Out-Richtlinie für den Fall festgelegt, dass die Metrik über 50 Prozent liegt. Dann haben Sie einen zweiten Alarm und eine

Abskalierungsrichtlinie für den Fall erstellt, dass die Metrik unter 50 Prozent liegt. Sie haben eine Reihe von schrittweisen Anpassungen mit dem Anpassungstyp `PercentChangeInCapacity` für jede Richtlinie vorgenommen:

Beispiel: Schrittanpassungen für die Richtlinie zur horizontalen Skalierung nach oben

Untergrenze	Obergrenze	Anpassung
0	10	0
10	20	10
20	Null	30

Beispiel: Schrittanpassungen für die Richtlinie zur horizontalen Skalierung nach unten

Untergrenze	Obergrenze	Anpassung
-10	0	0
-20	-10	-10
Null	-20	-30

Dadurch wird die folgende Skalierungskonfiguration erstellt.

```
Metric value

-infinity          30%    40%          60%     70%             infinity
-------------------------------------------------------------------------
          -30%      | -10% | Unchanged  | +10%  |       +30%
-------------------------------------------------------------------------
```

Angenommen, Sie verwenden diese Skalierungskonfiguration für ein skalierbares Ziel mit einer Kapazität von 10. Die folgenden Punkte fassen das Verhalten der Skalierungskonfiguration in Bezug auf die Kapazität des skalierbaren Ziels zusammen:

- Die ursprüngliche Kapazität wird aufrechterhalten, solange der aggregierte Metrikwert größer als 40 und kleiner als 60 ist.

- Wenn der Metrikwert 60 erreicht, erhöht Application Auto Scaling die Kapazität des skalierbaren Ziels um 1 auf 11. Dies basiert auf der zweiten Schrittanpassung der Richtlinie für die horizontale Skalierung nach oben (Erhöhen um 10 Prozent von 10). Nachdem die neue Kapazität hinzugefügt wurde, erhöht Application Auto Scaling die aktuelle Kapazität auf 11. Steigt der metrische Wert auch nach dieser Kapazitätserhöhung auf 70, erhöht Application Auto Scaling die Zielkapazität um 3 auf 14. Dies basiert auf der dritten Schrittanpassung der Richtlinie für die horizontale Skalierung nach oben (Erhöhen um 30 Prozent von 11, 3,3 abgerundet auf 3).
- Wenn der Metrikwert 40 erreicht, verringert Application Auto Scaling die Kapazität des skalierbaren Ziels um 1 auf 13, basierend auf der zweiten Anpassungsstufe der Scale-in-Richtlinie (Entfernen von 10 Prozent von 14, 1,4, abgerundet auf 1). Wenn der metrische Wert auch nach dieser Kapazitätsverringerung auf 30 fällt, verringert Application Auto Scaling die Zielkapazität um 3 auf 10, basierend auf der dritten Anpassungsstufe der Scale-in-Richtlinie (Entfernen von 30 Prozent von 13, 3,9, abgerundet auf 3).

Wenn Sie die Schrittanpassungen für Ihre Skalierungsrichtlinie angeben, beachten Sie Folgendes:

- Die Bereiche der Schrittanpassungen dürfen sich nicht überschneiden oder Lücken aufweisen.
- Nur eine Schrittanpassung darf über einen Nullwert als Untergrenze verfügen (negative Unendlichkeit). Verfügt eine Schrittanpassung über eine negative Untergrenze, muss eine Schrittanpassung mit einem Nullwert als Untergrenze vorhanden sein.
- Nur eine Schrittanpassung darf über einen Nullwert als Obergrenze verfügen (positive Unendlichkeit). Verfügt eine Schrittanpassung über eine positive Obergrenze, muss eine Schrittanpassung mit einem Nullwert als Obergrenze vorhanden sein.
- Ober- und Untergrenze einer Schrittanpassung können nicht gleichzeitig über einen Nullwert verfügen.
- Liegt der Metrikwert oberhalb des Verletzungsschwellenwerts, wird die Untergrenze eingeschlossen und die Obergrenze ausgeschlossen. Liegt der Metrikwert unterhalb des Verletzungsschwellenwerts, wird die Untergrenze ausgeschlossen und die Obergrenze eingeschlossen.

Skalierungsanpassungstypen

Sie können eine Skalierungsrichtlinie definieren, welche die optimale Skalierungsaktion basierend auf dem von Ihnen gewählten Skalierungsanpassungstyp ausführt. Sie können den Anpassungstyp als Prozentsatz der aktuellen Kapazität Ihres skalierbaren Ziels oder in absoluten Zahlen angeben.

Application Auto Scaling unterstützt die folgenden Anpassungstypen für Stufenskalierungsrichtlinien:

- ChangeInCapacity— Erhöht oder verringert die aktuelle Kapazität des skalierbaren Ziels um den angegebenen Wert. Ein positiver Wert erhöht die Kapazität, ein negativer Anpassungswert verringert die Kapazität. Ein Beispiel: Wenn die aktuelle Kapazität 3 ist und die Anpassung 5 beträgt, fügt Application Auto Scaling der Kapazität 5 hinzu, so dass sie insgesamt 8 beträgt.
- ExactCapacity— Ändert die aktuelle Kapazität des skalierbaren Ziels auf den angegebenen Wert. Geben Sie bei diesem Anpassungstyp einen nicht-negativen Wert an. Ein Beispiel: Wenn die aktuelle Kapazität 3 ist und die Anpassung 5 beträgt, ändert Application Auto Scaling die Kapazität auf 5.
- PercentChangeInCapacity— Erhöht oder verringert die aktuelle Kapazität des skalierbaren Ziels um den angegebenen Prozentsatz. Ein positiver Wert erhöht die Kapazität, ein negativer Anpassungswert verringert die Kapazität. Ein Beispiel: Wenn die aktuelle Kapazität 10 ist und die Anpassung 10 Prozent beträgt, fügt Application Auto Scaling 1 zur Kapazität hinzu, so dass sie insgesamt 11 beträgt.

> **Note**
>
> Wenn der resultierende Wert keine ganze Zahl ist, rundet Application Auto Scaling ihn wie folgt:
>
> - Werte größer als 1 werden abgerundet. Beispielsweise wird `12.7` auf `12` gerundet.
> - Werte zwischen 0 und 1 werden auf 1 gerundet. Beispielsweise wird `.67` auf `1` gerundet.
> - Werte zwischen 0 und -1 werden auf -1 gerundet. Beispielsweise wird `-.58` auf `-1` gerundet.
> - Werte kleiner als -1 werden aufgerundet. Beispielsweise wird `-6.67` auf `-6` gerundet.

Mit PercentChangeInCapacitykönnen Sie auch den Mindestbetrag für die Skalierung mithilfe des `MinAdjustmentMagnitude` Parameters angeben. Angenommen, Sie erstellen eine Richtlinie zum Hinzufügen von 25 Prozent und geben an, dass mindestens 2 hinzugefügt werden sollen. Hat das skalierbare Ziel eine Kapazität von 4 und wird die Skalierungsrichtlinie ausgeführt, ist 25 Prozent von 4 gleich 1. Da Sie jedoch eine Mindestschrittweite von 2 angegeben haben, fügt Application Auto Scaling 2 hinzu.

Ruhephase

In Ihrer Richtlinie für die schrittweise Skalierung können Sie optional eine Ruhephase definieren.

Eine Ruhephase ist die Zeitspanne, die die Skalierungsrichtlinie warten muss, bis eine vorherige Skalierungsaktivität wirksam wird.

Es gibt zwei Möglichkeiten, die Verwendung von Ruhephasen für eine Konfiguration mit schrittweiser Skalierung zu planen:

- Mit den Richtlinien zur Ruhephase der Aufskalierung wird beabsichtigt, kontinuierlich (aber nicht übermäßig) aufzuskalieren. Nachdem Application Auto Scaling unter Verwendung einer Skalierungsrichtlinie erfolgreich aufskaliert wurde, wird die Berechnung der Ruhezeit gestartet. Eine Skalierungsrichtlinie erhöht die gewünschte Kapazität nicht erneut, es sei denn, es wird eine größere Aufskalierung ausgelöst oder die Ruhephase endet. Während die Scale-Out-Ruhephase wirksam ist, wird die durch die initiierende horizontale Skalierung nach oben (Scale-Out) hinzugefügte Kapazität als Teil der gewünschten Kapazität für die nächste horizontale Skalierung nach oben berechnet.
- Mit den Richtlinien der Ruhephase für die Abskalierung ist beabsichtigt, die Abskalierung konservativ durchzuführen, um die Verfügbarkeit Ihrer Anwendung zu schützen, sodass Abskalierungsaktivitäten blockiert werden, bis die Ruhephase für die Abskalierung abgelaufen ist. Wenn jedoch ein anderer Alarm während der Abkühlphase nach einer Abskalier-Aktivität eine Aufskalier-Aktivität auslöst, wird das Ziel durch Application Auto Scaling sofort abskaliert. In diesem Fall wird die Ruhephase für die Abskalierung angehalten und nicht abgeschlossen.

Wenn beispielsweise eine Spitze im Datenverkehr auftritt, wird ein Alarm ausgelöst und Application Auto Scaling fügt automatisch Kapazität hinzu, um die erhöhte Last zu bewältigen. Wenn Sie eine Ruhephase für Ihre Richtlinie für Aufskalierung festlegen und der Alarm die Richtlinie auslöst, um die Kapazität um 2 zu erhöhen, wird die Skalierung erfolgreich abgeschlossen und die Ruhephase für die Aufskalierung beginnt. Wenn ein Alarm die gleiche Richtlinie, aber mit einer aggressiveren Stufenanpassung, z. B. um 3, während der Ruhephase erneut auslöst, wird die vorherige Erhöhung um 2 als Teil der aktuellen Kapazität betrachtet. Daher wird der Kapazität nur 1 hinzugefügt. Dies ermöglicht eine schnellere Skalierung als das Warten auf den Ablauf der Ruhephase, ohne dass Sie mehr Kapazität hinzufügen, als Sie benötigen.

Die Ruhephase wird in Sekunden gemessen und gilt nur für Skalierungsrichtlinien-bezogene Skalierungen. Wenn eine geplante Aktion während einer Ruhephase zum geplanten Zeitpunkt

beginnt, kann sie umgehend eine Skalierung auslösen, ohne das Ablaufen der Ruhephase abzuwarten.

Der Standardwert ist 300, wenn kein Wert angegeben wird.

Häufig verwendete Befehle zur Erstellung, Verwaltung und Löschung von Skalierungsrichtlinien

Zu den häufig verwendeten Befehlen für die Arbeit mit Skalierungsrichtlinien gehören:

- register-scalable-targetum Ressourcen als skalierbare Ziele zu registrieren AWS oder anzupassen (eine Ressource, die Application Auto Scaling skalieren kann) und die Skalierung auszusetzen und wieder aufzunehmen.
- put-scaling-policyum Skalierungsrichtlinien für ein vorhandenes skalierbares Ziel hinzuzufügen oder zu ändern.
- describe-scaling-activitiesum Informationen über Skalierungsaktivitäten in einer AWS Region zurückzugeben.
- describe-scaling-policiesum Informationen über Skalierungsrichtlinien in einer AWS Region zurückzugeben.
- delete-scaling-policyum eine Skalierungsrichtlinie zu löschen.

Überlegungen

Bei der Arbeit mit Richtlinien zur schrittweisen Skalierung ist Folgendes zu beachten:

- Überlegen Sie, ob Sie die Schrittanpassungen in der Anwendung genau genug vorhersagen können, um die schrittweise Skalierung zu verwenden. Wenn Ihre Skalierungsmetrik die Kapazität des skalierbaren Ziels proportional vergrößert oder verkleinert, raten wir stattdessen zur Verwendung einer Skalierungsrichtlinie für die Ziel-Nachverfolgung. Sie haben weiterhin die Möglichkeit, die Schrittskalierung als zusätzliche Richtlinie für eine erweiterte Konfiguration zu verwenden. Beispiel: Sie können eine striktere Antwort konfigurieren, sobald die Auslastung ein bestimmtes Niveau erreicht.
- Achten Sie darauf, einen angemessenen Abstand zwischen den Schwellenwerten für Scale-Out und Scale-In zu wählen, um ein Flattern zu verhindern. Flattern beschreibt eine Endlosschleife aus Auf- und Abwärtsskalieren. Das heißt, wenn eine Skalierungsaktion durchgeführt wird, würde sich der Metrikwert ändern und eine weitere Skalierungsaktion in der umgekehrten Richtung starten.

Zugehörige Ressourcen

Informationen zum Erstellen von Richtlinien zur schrittweisen Skalierung für Auto Scaling-Gruppen finden Sie unter Schrittweise und einfache Skalierungsrichtlinien für Amazon EC2 Auto Scaling im Benutzerhandbuch zu Amazon EC2 Auto Scaling.

Einschränkungen

- Der Konsolenzugriff zum Anzeigen, Hinzufügen, Aktualisieren oder Entfernen von Richtlinien zur schrittweisen Skalierung für die Ziel-Nachverfolgung auf skalierbaren Ressourcen hängt von der verwendeten Ressource ab. Weitere Informationen finden Sie unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

Eine Stufenskalierungsrichtlinie mit der AWS CLI erstellen

Sie können eine schrittweise Skalierungsrichtlinie für Application Auto Scaling erstellen, indem Sie die AWS CLI für die folgenden Konfigurationsaufgaben verwenden.

1. Registrieren eines skalierbaren Ziels
2. Fügen Sie eine Richtlinie zur schrittweisen Skalierung für das skalierbare Ziel hinzu.
3. Erstellen Sie einen CloudWatch Alarm für die Richtlinie.

Der Kürze halber zeigen die Beispiele in diesem Thema CLI-Befehle für einen Amazon ECS-Service. Um ein anderes skalierbares Ziel anzugeben, geben Sie seinen Namespace in `--service-namespace`, seine skalierbare Dimension in `--scalable-dimension` und seine Ressourcen-ID in `--resource-id` an. Weitere Informationen und Beispiele für die einzelnen Services finden Sie in den Themen unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

Denken Sie bei der Verwendung von daran AWS CLI, dass Ihre Befehle in der für Ihr Profil AWS-Region konfigurierten Version ausgeführt werden. Wenn Sie die Befehle in einer anderen Region ausführen möchten, ändern Sie entweder die Standardregion für Ihr Profil, oder verwenden Sie den `--region`-Parameter mit dem Befehl.

Inhalt

- Registrieren eines skalierbaren Ziels
- Erstellen Sie eine Skalierungsrichtlinie

- Erstellen eines Alarms, der die Skalierungsrichtlinie auslöst
- Beschreiben Sie Richtlinien für die Stufenskalierung
- Löschen einer Stufenskalierungsrichtlinie

Registrieren eines skalierbaren Ziels

Wenn Sie dies noch nicht getan haben, registrieren Sie das skalierbare Ziel. Verwenden Sie den register-scalable-targetBefehl, um eine bestimmte Ressource im Zieldienst als skalierbares Ziel zu registrieren. Im folgenden Beispiel wird ein Amazon ECS-Service mit Application Auto Scaling registriert. Application Auto Scaling kann die Anzahl der Aufgaben auf ein Minimum von 2 und ein Maximum von 10 skalieren. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Linux, macOS oder Unix

```
aws application-autoscaling register-scalable-target --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service \
  --min-capacity 2 --max-capacity 10
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace ecs --
scalable-dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-service
 --min-capacity 2 --max-capacity 10
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Erstellen Sie eine Skalierungsrichtlinie

Um eine schrittweise Skalierungsrichtlinie für Ihr skalierbares Ziel zu erstellen, können Sie die folgenden Beispiele verwenden, um Ihnen den Einstieg zu erleichtern.

Scale out

So erstellen Sie eine Richtlinie zur schrittweisen Skalierung für Scale-Out (Kapazitätserhöhung)

1. Verwenden Sie den folgenden `cat` Befehl, um eine Konfiguration einer schrittweisen Skalierungsrichtlinie in einer JSON-Datei mit dem Namen `config.json` in Ihrem Home-Verzeichnis zu speichern. Im Folgenden finden Sie eine Beispielkonfiguration mit einem Anpassungstyp von`PercentChangeInCapacity`, die die Kapazität des skalierbaren Ziels auf der Grundlage der folgenden schrittweisen Anpassungen erhöht (unter der Annahme eines CloudWatch Alarmschwellenwerts von 70):

   - Erhöhen Sie die Kapazität um 10 Prozent, wenn der Wert der Metrik größer oder gleich 70, aber kleiner als 85 ist
   - Erhöhen Sie die Kapazität um 20 Prozent, wenn der Wert der Metrik größer oder gleich 85, aber kleiner als 95 ist
   - Erhöhen Sie die Kapazität um 30 Prozent, wenn der Wert der Metrik größer oder gleich 95 ist

```
$ cat ~/config.json
{
  "AdjustmentType": "PercentChangeInCapacity",
  "MetricAggregationType": "Average",
  "Cooldown": 60,
  "MinAdjustmentMagnitude": 1,
  "StepAdjustments": [
    {
      "MetricIntervalLowerBound": 0.0,
      "MetricIntervalUpperBound": 15.0,
      "ScalingAdjustment": 10
    },
    {
      "MetricIntervalLowerBound": 15.0,
      "MetricIntervalUpperBound": 25.0,
      "ScalingAdjustment": 20
    },
    {
      "MetricIntervalLowerBound": 25.0,
      "ScalingAdjustment": 30
    }
  ]
```

```
}
```

Weitere Informationen finden Sie StepScalingPolicyConfigurationin der API-Referenz für Application Auto Scaling.

2. Verwenden Sie den folgenden put-scaling-policyBefehl zusammen mit der `config.json` Datei, die Sie erstellt haben, um eine Skalierungsrichtlinie mit dem Namen zu erstellen`my-step-scaling-policy`.

   Linux, macOS oder Unix

   ```
   aws application-autoscaling put-scaling-policy --service-namespace ecs \
     --scalable-dimension ecs:service:DesiredCount \
     --resource-id service/my-cluster/my-service \
     --policy-name my-step-scaling-policy --policy-type StepScaling \
     --step-scaling-policy-configuration file://config.json
   ```

   Windows

   ```
   aws application-autoscaling put-scaling-policy --service-namespace ecs --
   scalable-dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-
   service --policy-name my-step-scaling-policy --policy-type StepScaling --step-
   scaling-policy-configuration file://config.json
   ```

   Die Ausgabe enthält den ARN, der als eindeutiger Name für die Richtlinie dient. Sie benötigen ihn, um einen CloudWatch Alarm für Ihre Richtlinie zu erstellen.

   ```
   {
       "PolicyARN":
    "arn:aws:autoscaling:region:123456789012:scalingPolicy:ac542982-cbeb-4294-891c-
   a5a941dfa787:resource/ecs/service/my-cluster/my-service:policyName/my-step-
   scaling-policy"
   }
   ```

Scale in

Um eine schrittweise Skalierungsrichtlinie für die Skalierung (Kapazitätsreduzierung) zu erstellen

1. Verwenden Sie den folgenden `cat` Befehl, um eine Konfiguration einer schrittweisen Skalierungsrichtlinie in einer JSON-Datei mit dem Namen `config.json` in Ihrem Home-

Verzeichnis zu speichern. Im Folgenden finden Sie eine Beispielkonfiguration mit einem Anpassungstyp von`ChangeInCapacity`, der die Kapazität des skalierbaren Ziels auf der Grundlage der folgenden schrittweisen Anpassungen verringert (unter der Annahme eines CloudWatch Alarmschwellenwerts von 50):

- Verringern Sie die Kapazität um 1, wenn der Wert der Metrik kleiner oder gleich 50, aber größer als 40 ist
- Verringern Sie die Kapazität um 2, wenn der Wert der Metrik kleiner oder gleich 40, aber größer als 30 ist
- Verringern Sie die Kapazität um 3, wenn der Wert der Metrik kleiner oder gleich 30 ist

```
$ cat ~/config.json
{
  "AdjustmentType": "ChangeInCapacity",
  "MetricAggregationType": "Average",
  "Cooldown": 60,
  "StepAdjustments": [
    {
      "MetricIntervalUpperBound": 0.0,
      "MetricIntervalLowerBound": -10.0,
      "ScalingAdjustment": -1
    },
    {
      "MetricIntervalUpperBound": -10.0,
      "MetricIntervalLowerBound": -20.0,
      "ScalingAdjustment": -2
    },
    {
      "MetricIntervalUpperBound": -20.0,
      "ScalingAdjustment": -3
    }
  ]
}
```

Weitere Informationen finden Sie StepScalingPolicyConfigurationin der API-Referenz für Application Auto Scaling.

2. Verwenden Sie den folgenden put-scaling-policyBefehl zusammen mit der `config.json` Datei, die Sie erstellt haben, um eine Skalierungsrichtlinie mit dem Namen zu erstellen`my-step-scaling-policy`.

Linux, macOS oder Unix

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service \
  --policy-name my-step-scaling-policy --policy-type StepScaling \
  --step-scaling-policy-configuration file://config.json
```

Windows

```
aws application-autoscaling put-scaling-policy --service-namespace ecs --
scalable-dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-
service --policy-name my-step-scaling-policy --policy-type StepScaling --step-
scaling-policy-configuration file://config.json
```

Die Ausgabe enthält den ARN, der als eindeutiger Name für die Richtlinie dient. Sie benötigen ihn, um einen CloudWatch Alarm für Ihre Richtlinie zu erstellen.

```
{
    "PolicyARN":
 "arn:aws:autoscaling:region:123456789012:scalingPolicy:ac542982-cbeb-4294-891c-
a5a941dfa787:resource/ecs/service/my-cluster/my-service:policyName/my-step-
scaling-policy"
}
```

Erstellen eines Alarms, der die Skalierungsrichtlinie auslöst

Verwenden Sie abschließend den folgenden CloudWatch put-metric-alarmBefehl, um einen Alarm zu erstellen, der mit Ihrer Richtlinie zur schrittweisen Skalierung verwendet werden kann. In diesem Beispiel haben Sie einen Alarm, der auf der durchschnittlichen CPU-Auslastung basiert. Der Alarm ist so konfiguriert, dass er sich in einem ALARM-Zustand befindet, wenn er für mindestens zwei aufeinanderfolgende Auswerteperioden von 60 Sekunden einen Schwellenwert von 70 Prozent erreicht. Um eine andere CloudWatch Metrik anzugeben oder Ihre eigene benutzerdefinierte Metrik zu verwenden, geben Sie ihren Namen `--metric-name` und ihren Namespace in `--namespace` an.

Linux, macOS oder Unix

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmHigh-ECS:service/my-
cluster/my-service \
  --metric-name CPUUtilization --namespace AWS/ECS --statistic Average \
  --period 60 --evaluation-periods 2 --threshold 70 \
  --comparison-operator GreaterThanOrEqualToThreshold \
  --dimensions Name=ClusterName,Value=default Name=ServiceName,Value=sample-app-service
 \
  --alarm-actions PolicyARN
```

Windows

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmHigh-ECS:service/
my-cluster/my-service --metric-name CPUUtilization --namespace AWS/ECS --statistic
 Average --period 60 --evaluation-periods 2 --threshold 70 --comparison-operator
 GreaterThanOrEqualToThreshold --dimensions Name=ClusterName,Value=default
 Name=ServiceName,Value=sample-app-service --alarm-actions PolicyARN
```

Beschreiben Sie Richtlinien für die Stufenskalierung

Mit dem folgenden describe-scaling-policiesBefehl können Sie alle Skalierungsrichtlinien für den angegebenen Dienst-Namespace beschreiben.

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

Sie können die Ergebnisse mithilfe des Parameters `--query` filtern, um nur die Richtlinien zur schrittweisen Skalierung zu erhalten. Weitere Informationen über die Syntax von `query` finden Sie unter Steuerung der Befehlsausgabe vom AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs \
  --query 'ScalingPolicies[?PolicyType==`StepScaling`]'
```

Windows

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs --query
 "ScalingPolicies[?PolicyType==`StepScaling`]"
```

Es folgt eine Beispielausgabe.

```
[
    {
        "PolicyARN": "PolicyARN",
        "StepScalingPolicyConfiguration": {
            "MetricAggregationType": "Average",
            "Cooldown": 60,
            "StepAdjustments": [
                {
                    "MetricIntervalLowerBound": 0.0,
                    "MetricIntervalUpperBound": 15.0,
                    "ScalingAdjustment": 1
                },
                {
                    "MetricIntervalLowerBound": 15.0,
                    "MetricIntervalUpperBound": 25.0,
                    "ScalingAdjustment": 2
                },
                {
                    "MetricIntervalLowerBound": 25.0,
                    "ScalingAdjustment": 3
                }
            ],
            "AdjustmentType": "ChangeInCapacity"
        },
        "PolicyType": "StepScaling",
        "ResourceId": "service/my-cluster/my-service",
        "ServiceNamespace": "ecs",
        "Alarms": [
            {
                "AlarmName": "Step-Scaling-AlarmHigh-ECS:service/my-cluster/my-
service",
                "AlarmARN": "arn:aws:cloudwatch:region:012345678910:alarm:Step-Scaling-
AlarmHigh-ECS:service/my-cluster/my-service"
            }
        ],
        "PolicyName": "my-step-scaling-policy",
        "ScalableDimension": "ecs:service:DesiredCount",
        "CreationTime": 1515024099.901
    }
]
```

Löschen einer Stufenskalierungsrichtlinie

Wenn Sie keine Richtlinie für eine schrittweise Skalierung mehr benötigen, können Sie diese löschen. Führen Sie die folgenden Aufgaben aus, um sowohl die Skalierungsrichtlinie als auch den CloudWatch Alarm zu löschen.

So löschen Sie Ihre Skalierungsrichtlinie

Verwenden Sie den folgenden delete-scaling-policy-Befehl.

Linux, macOS oder Unix

```
aws application-autoscaling delete-scaling-policy --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service \
  --policy-name my-step-scaling-policy
```

Windows

```
aws application-autoscaling delete-scaling-policy --service-namespace ecs --scalable-
dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-service --
policy-name my-step-scaling-policy
```

Um den CloudWatch Alarm zu löschen

Verwenden Sie den delete-alarms-Befehl. Sie können einen oder mehrere Alarme gleichzeitig löschen. Sie können beispielsweise den folgenden Befehl verwenden, um die Alarme `Step-Scaling-AlarmHigh-ECS:service/my-cluster/my-service` und `Step-Scaling-AlarmLow-ECS:service/my-cluster/my-service` zu löschen.

```
aws cloudwatch delete-alarms --alarm-name Step-Scaling-AlarmHigh-ECS:service/my-
cluster/my-service Step-Scaling-AlarmLow-ECS:service/my-cluster/my-service
```

Tutorial: Auto Scaling zur Bewältigung eines hohen Workloads konfigurieren

> **Important**
>
> Bevor Sie sich mit diesem Tutorial beschäftigen, empfehlen wir, dass Sie zunächst die folgenden einführenden Themen durcharbeiten: Tutorial: Erste Schritte mit der geplanten Skalierung mit AWS CLI.

In diesem Tutorial lernen Sie, wie Sie in Zeitfenstern, in denen Ihre Anwendung ein höheres als das normale Workload aufweist, horizontal skalieren. Dies ist hilfreich, wenn Sie eine Anwendung haben, die in regelmäßigen Abständen oder saisonal plötzlich eine große Anzahl von Besuchern haben kann.

Sie können eine Zielverfolgungs-Skalierungsrichtlinie zusammen mit einer geplanten Skalierung verwenden, um die zusätzliche Last zu bewältigen. Bei der geplanten Skalierung werden automatisch Änderungen an Ihren `MinCapacity` und `MaxCapacity` in Ihrem Namen nach einem von Ihnen festgelegten Zeitplan initiiert. Wenn eine Zielverfolgungs-Skalierungsrichtlinie für die Ressource aktiv ist, kann sie dynamisch auf der Grundlage der aktuellen Ressourcenauslastung innerhalb des neuen minimalen und maximalen Kapazitätsbereichs skaliert werden.

Nach Abschluss dieses Tutorials wissen Sie, wie Sie:

- Mit der geplanten Skalierung können Sie zusätzliche Kapazität hinzufügen, um eine hohe Last zu bewältigen, bevor sie eintrifft, und die zusätzliche Kapazität entfernen, wenn sie nicht mehr benötigt wird.
- Verwenden Sie eine Zielverfolgungs-Skalierungsrichtlinie, um Ihre Anwendung auf der Grundlage der aktuellen Ressourcenauslastung zu skalieren.

Inhalt

- Voraussetzungen
- Schritt 1: Registrieren Sie Ihr skalierbares Ziel
- Schritt 2: Richten Sie geplante Aktionen entsprechend Ihren Anforderungen ein
- Schritt 3: Hinzufügen einer Skalierungsrichtlinie für die Zielverfolgung

- Schritt 4: Nächste Schritte
- Schritt 5: Bereinigen

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie Folgendes bereits gemacht haben:

- Sie haben ein AWS-Konto erstellt. Weitere Informationen finden Sie unter Einrichten, um Application Auto Scaling zu verwenden.
- Sie installierten und konfigurierten die AWS CLI. Weitere Informationen finden Sie unter Richten Sie das ein AWS CLI.
- Ihr Konto verfügt über alle erforderlichen Berechtigungen zum Registrieren und Deregistrieren von Ressourcen als skalierbare Ziele mit Application Auto Scaling. Außerdem verfügt sie über alle erforderlichen Berechtigungen zur Erstellung von Skalierungsrichtlinien und geplanten Aktionen. Weitere Informationen finden Sie unter Identity and Access Management für Application Auto Scaling.
- Sie verfügen über eine unterstützte Ressource in einer nicht produktiven Umgebung, die Sie für dieses Lernprogramm verwenden können. Wenn Sie noch über keines verfügen, erstellen Sie jetzt eines. Weitere Informationen zu den AWS-Services und Ressourcen, die mit Application Auto Scaling zusammenarbeiten, finden Sie im AWS -Services, die Sie mit Application Auto Scaling verwenden können-Abschnitt.

> **Note**
>
> Bei der Durchführung dieses Tutorials gibt es zwei Schritte, in denen Sie die minimalen und maximalen Kapazitätswerte Ihrer Ressource auf 0 festlegen, um die aktuelle Kapazität auf 0 zurückzusetzen. Je nachdem, welche Ressource Sie mit Application Auto Scaling verwenden, können Sie die aktuelle Kapazität während dieser Schritte möglicherweise nicht auf 0 zurücksetzen. Um Ihnen bei der Lösung des Problems zu helfen, wird eine Meldung in der Ausgabe darauf hinweisen, dass die Mindestkapazität nicht kleiner als der angegebene Wert sein kann, und den Mindestkapazitätswert angeben, den die AWS-Ressource akzeptieren kann.

Schritt 1: Registrieren Sie Ihr skalierbares Ziel

Beginnen Sie damit, Ihre Ressource als skalierbares Ziel bei Application Auto Scaling zu registrieren. Ein skalierbares Ziel ist eine Ressource, die dank Application Auto Scaling auf- und abskaliert werden kann.

So registrieren Sie Ihr skalierbares Ziel mit Application Auto Scaling

- Verwenden Sie den folgenden Befehl register-scalable-target, um ein neues skalierbares Ziel zu registrieren. Setzen Sie die Werte `--min-capacity` und `--max-capacity` auf 0, um die aktuelle Kapazität auf 0 zurückzusetzen.

  Ersetzen Sie den Beispieltext für `--service-namespace` mit dem Namespace des AWS-Services, den Sie mit Application Auto Scaling verwenden, `--scalable-dimension` mit der skalierbaren Dimension im Zusammenhang mit der Ressource, die Sie registrieren, `--resource-id` mit einem Bezeichner für die Ressource. Diese Werte variieren je nachdem, welche Ressource verwendet wird und wie die Ressourcen-ID erstellt wird. Sehen Sie sich die Themen im AWS -Services, die Sie mit Application Auto Scaling verwenden können-Abschnitt für weitere Informationen an. Zu diesen Themen gehören Beispielbefehle, die Ihnen zeigen, wie Sie skalierbare Ziele bei Application Auto Scaling registrieren.

  Linux, macOS oder Unix

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace namespace \
    --scalable-dimension dimension \
    --resource-id identifier \
    --min-capacity 0 --max-capacity 0
  ```

  Windows

  ```
  aws application-autoscaling register-scalable-target --service-namespace namespace
   --scalable-dimension dimension --resource-id identifier --min-capacity 0 --max-
  capacity 0
  ```

  Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

  ```
  {
  ```

```
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-
id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Schritt 2: Richten Sie geplante Aktionen entsprechend Ihren Anforderungen ein

Mit dem Befehl put-scheduled-action können Sie zeitgesteuerte Aktionen erstellen, die so konfiguriert sind, dass sie Ihren geschäftlichen Anforderungen entsprechen. In diesem Tutorial konzentrieren wir uns auf eine Konfiguration, die den Verbrauch von Ressourcen außerhalb der Arbeitszeiten stoppt, indem die Kapazität auf 0 reduziert wird.

So erstellen Sie eine geplante Aktion, die am Morgen ausläuft

1. Um das skalierbare Ziel zu skalieren, verwenden Sie den folgenden Befehl put-scheduled-action. Fügen Sie den Parameter `--schedule` mit einem wiederkehrenden Zeitplan in UTC ein, indem Sie einen cron-Ausdruck verwenden.

   Nach dem festgelegten Zeitplan (jeden Tag um 9:00 Uhr UTC) aktualisiert Application Auto Scaling die Werte `MinCapacity` und `MaxCapacity` auf den gewünschten Bereich von 1-5 Kapazitätseinheiten.

   Linux, macOS oder Unix

   ```
   aws application-autoscaling put-scheduled-action \
     --service-namespace namespace \
     --scalable-dimension dimension \
     --resource-id identifier \
     --scheduled-action-name my-first-scheduled-action \
     --schedule "cron(0 9 * * ? *)" \
     --scalable-target-action MinCapacity=1,MaxCapacity=5
   ```

   Windows

   ```
   aws application-autoscaling put-scheduled-action --service-namespace namespace --
   scalable-dimension dimension --resource-id identifier --scheduled-action-name my-
   first-scheduled-action --schedule "cron(0 9 * * ? *)" --scalable-target-action
    MinCapacity=1,MaxCapacity=5
   ```

Dieser Befehl gibt keine Ausgabe zurück, wenn er nicht erfolgreich ist.

2. Um zu bestätigen, dass Ihre geplante Aktion existiert, verwenden Sie den folgenden Befehl describe-scheduled-actions.

   Linux, macOS oder Unix

```
aws application-autoscaling describe-scheduled-actions \
  --service-namespace namespace \
  --query 'ScheduledActions[?ResourceId==`identifier`]'
```

   Windows

```
aws application-autoscaling describe-scheduled-actions --service-
namespace namespace --query "ScheduledActions[?ResourceId==`identifier`]"
```

   Es folgt eine Beispielausgabe.

```
[
    {
        "ScheduledActionName": "my-first-scheduled-action",
        "ScheduledActionARN": "arn",
        "Schedule": "cron(0 9 * * ? *)",
        "ScalableTargetAction": {
            "MinCapacity": 1,
            "MaxCapacity": 5
        },
        ...
    }
]
```

So erstellen Sie eine geplante Aktion, die nachts aktiviert wird

1. Wiederholen Sie das vorangegangene Verfahren, um eine weitere geplante Aktion zu erstellen, mit der Application Auto Scaling am Ende des Tages aktiviert wird.

   Am angegebenen Zeitplan (jeden Tag um 20:00 Uhr UTC) aktualisiert Application Auto Scaling die Werte `MinCapacity` und `MaxCapacity` des Ziels auf 0, wie durch den folgenden Befehl put-scheduled-action angewiesen.

Linux, macOS oder Unix

```
aws application-autoscaling put-scheduled-action \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier \
  --scheduled-action-name my-second-scheduled-action \
  --schedule "cron(0 20 * * ? *)" \
  --scalable-target-action MinCapacity=0,MaxCapacity=0
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace namespace --
scalable-dimension dimension --resource-id identifier --scheduled-action-name my-
second-scheduled-action --schedule "cron(0 20 * * ? *)" --scalable-target-action
 MinCapacity=0,MaxCapacity=0
```

2. Um zu bestätigen, dass Ihre geplante Aktion existiert, verwenden Sie den folgenden Befehl describe-scheduled-actions.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scheduled-actions \
  --service-namespace namespace \
  --query 'ScheduledActions[?ResourceId==`identifier`]'
```

Windows

```
aws application-autoscaling describe-scheduled-actions --service-
namespace namespace --query "ScheduledActions[?ResourceId==`identifier`]"
```

Es folgt eine Beispielausgabe.

```
[
    {
        "ScheduledActionName": "my-first-scheduled-action",
        "ScheduledActionARN": "arn",
        "Schedule": "cron(0 9 * * ? *)",
        "ScalableTargetAction": {
            "MinCapacity": 1,
```

```
            "MaxCapacity": 5
        },
        ...
    },
    {
        "ScheduledActionName": "my-second-scheduled-action",
        "ScheduledActionARN": "arn",
        "Schedule": "cron(0 20 * * ? *)",
        "ScalableTargetAction": {
            "MinCapacity": 0,
            "MaxCapacity": 0
        },
        ...
    }
]
```

Schritt 3: Hinzufügen einer Skalierungsrichtlinie für die Zielverfolgung

Bei der Zielverfolgung vergleicht Application Auto Scaling den Zielwert in der Richtlinie mit dem aktuellen Wert der angegebenen Metrik.

Bei der Zielverfolgung vergleicht Application Auto Scaling den Zielwert in der Richtlinie mit dem aktuellen Wert der angegebenen Metrik. Wenn diese Werte über einen bestimmten Zeitraum hinweg nicht übereinstimmen, fügt Application Auto Scaling Kapazitäten hinzu oder entfernt sie, um eine gleichmäßige Leistung zu gewährleisten. Wenn die Belastung Ihrer Anwendung und der Metrikwert steigen, fügt Application Auto Scaling so schnell wie möglich Kapazität hinzu, ohne den Wert `MaxCapacity` zu überschreiten. Wenn Application Auto Scaling Kapazität entfernt, weil die Last minimal ist, geschieht dies, ohne dass der Wert unter `MinCapacity` fällt. Durch die Anpassung der Kapazität an die Nutzung zahlen Sie nur für das, was Ihre Anwendung benötigt.

Wenn die Metrik unzureichende Daten aufweist, weil Ihre Anwendung nicht ausgelastet ist, wird durch Application Auto Scaling keine Kapazität hinzugefügt oder entfernt. Mit anderen Worten: Application Auto Scaling priorisiert die Verfügbarkeit in Situationen, in denen nicht genügend Informationen verfügbar sind.

Sie können mehrere Skalierungsrichtlinien hinzufügen, aber stellen Sie sicher, dass Sie keine widersprüchlichen Stufenskalierungsrichtlinien hinzufügen, da dies zu unerwünschtem Verhalten führen könnte. Wenn beispielsweise die Schrittskalierungsrichtlinie eine

Abwärtsskalierungsaktivität initiiert, bevor die Zielverfolgungsrichtlinie abwärts skaliert werden kann, wird die Abwärtsskalierungsaktivität nicht blockiert. Nach Abschluss der Skalierungsaktivität kann die Zielverfolgungsrichtlinie Application Auto Scaling anweisen, die Skalierung wieder zu beenden.

So erstellen Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung

1. Verwenden Sie den folgenden Befehl put-scaling-policy, um die Richtlinie zu erstellen.

   Die Metriken, die am häufigsten für die Zielverfolgung verwendet werden, sind vordefiniert, und Sie können sie verwenden, ohne die vollständige Metrikspezifikation von CloudWatch bereitzustellen. Weitere Informationen zu den verfügbaren vordefinierten Metriken finden Sie unter Skalierungsrichtlinien für die Ziel-Nachverfolgung.

   Bevor Sie diesen Befehl ausführen, stellen Sie sicher, dass Ihre vordefinierte Metrik den Zielwert erwartet. Wenn Sie beispielsweise eine Skalierung vornehmen möchten, wenn die CPU-Auslastung 50 % erreicht, geben Sie einen Zielwert von 50,0 an. Oder geben Sie einen Zielwert von 0,7 an, um die von Lambda bereitgestellte Gleichzeitigkeit zu reduzieren, wenn die Auslastung 70 % erreicht. Informationen zu den Zielwerten für eine bestimmte Ressource finden Sie in der vom Dienst bereitgestellten Dokumentation zur Konfiguration der Zielverfolgung. Weitere Informationen finden Sie unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

   Linux, macOS oder Unix

   ```
   aws application-autoscaling put-scaling-policy \
     --service-namespace namespace \
     --scalable-dimension dimension \
     --resource-id identifier \
     --policy-name my-scaling-policy --policy-type TargetTrackingScaling \
     --target-tracking-scaling-policy-configuration '{ "TargetValue": 50.0,
    "PredefinedMetricSpecification": { "PredefinedMetricType": "predefinedmetric" }}'
   ```

   Windows

   ```
   aws application-autoscaling put-scaling-policy --service-namespace namespace --
   scalable-dimension dimension --resource-id identifier --policy-name my-scaling-
   policy --policy-type TargetTrackingScaling --target-tracking-scaling-policy-
   configuration "{ \"TargetValue\": 50.0, \"PredefinedMetricSpecification\":
    { \"PredefinedMetricType\": \"predefinedmetric\" }}"
   ```

Bei Erfolg gibt dieser Befehl die ARNs und Namen der beiden CloudWatch-Alarme zurück, die in Ihrem Namen erstellt wurden.

2. Um zu bestätigen, dass Ihre geplante Aktion existiert, verwenden Sie den folgenden Befehl describe-scaling-policies.

   Linux, macOS oder Unix

   ```
   aws application-autoscaling describe-scaling-policies --service-namespace namespace
    \
     --query 'ScalingPolicies[?ResourceId==`identifier`]'
   ```

   Windows

   ```
   aws application-autoscaling describe-scaling-policies --service-namespace namespace
    --query "ScalingPolicies[?ResourceId==`identifier`]"
   ```

   Es folgt eine Beispielausgabe.

   ```
   [
       {
           "PolicyARN": "arn",
           "TargetTrackingScalingPolicyConfiguration": {
               "PredefinedMetricSpecification": {
                   "PredefinedMetricType": "predefinedmetric"
               },
               "TargetValue": 50.0
           },
           "PolicyName": "my-scaling-policy",
           "PolicyType": "TargetTrackingScaling",
           "Alarms": [],
           ...
       }
   ]
   ```

Schritt 4: Nächste Schritte

Wenn eine Skalierung auftritt, wird in der Ausgabe der Skalierung für das skalierbare Ziel eine Aufzeichnung angezeigt, zum Beispiel:

```
Successfully set desired count to 1. Change successfully fulfilled by ecs.
```

Um Ihre Skalierungsaktivitäten mit Application Auto Scaling zu überwachen, können Sie den folgenden Befehl describe-scaling-activities verwenden.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scaling-activities
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier
```

Windows

```
aws application-autoscaling describe-scaling-activities --service-namespace namespace
 --scalable-dimension dimension --resource-id identifier
```

Schritt 5: Bereinigen

Um zu verhindern, dass für Ihr Konto Gebühren für Ressourcen anfallen, die während der aktiven Skalierung erstellt wurden, können Sie die zugehörige Skalierungskonfiguration wie folgt bereinigen.

Durch das Löschen der Skalierungskonfiguration wird Ihre zugrunde liegende AWS-Ressource nicht gelöscht. Sie wird auch nicht auf ihre ursprüngliche Kapazität zurückgesetzt. Sie können die Konsole des Dienstes, in dem Sie die Ressource erstellt haben, verwenden, um sie zu löschen oder ihre Kapazität anzupassen.

So löschen Sie die geplanten Aktionen

Der folgende Befehl delete-scheduled-action löscht eine angegebene geplante Aktion. Sie können diesen Schritt überspringen, wenn Sie die von Ihnen erstellten geplanten Aktionen beibehalten möchten.

Linux, macOS oder Unix

```
aws application-autoscaling delete-scheduled-action \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier \
```

```
  --scheduled-action-name my-second-scheduled-action
```

Windows

```
aws application-autoscaling delete-scheduled-action --service-namespace namespace
 --scalable-dimension dimension --resource-id identifier --scheduled-action-name my-
second-scheduled-action
```

So löschen Sie die Skalierungsrichtlinie

Der folgende Befehl delete-scaling-policy löscht eine angegebene Zielverfolgungs-Skalierungsrichtlinie. Sie können diesen Schritt überspringen, wenn Sie die von Ihnen erstellte Skalierungsrichtlinie beibehalten möchten.

Linux, macOS oder Unix

```
aws application-autoscaling delete-scaling-policy \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier \
  --policy-name my-scaling-policy
```

Windows

```
aws application-autoscaling delete-scaling-policy --service-namespace namespace --
scalable-dimension dimension --resource-id identifier --policy-name my-scaling-policy
```

So melden Sie das skalierbare Ziel ab

Verwenden Sie den folgenden Befehl deregister-scalable-target, um das skalierbare Ziel zu deregistrieren. Mit diesem Befehl werden alle Skalierungsrichtlinien, die Sie erstellt haben, und alle geplanten Aktionen, die Sie noch nicht gelöscht haben, gelöscht. Sie können diesen Schritt überspringen, wenn das skalierbare Ziel für eine zukünftige Verwendung registriert bleiben soll.

Linux, macOS oder Unix

```
aws application-autoscaling deregister-scalable-target \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier
```

Windows

```
aws application-autoscaling deregister-scalable-target --service-namespace namespace --
scalable-dimension dimension --resource-id identifier
```

Die Skalierung von Application Auto Scaling unterbrechen und wiederaufnehmen

In diesem Thema wird erläutert, wie Sie mindestens eine der Skalierungsaktivitäten für die skalierbare Ziele in Ihrer Anwendung anhalten und anschließend wieder fortsetzen. Die Funktion zum Aus- und Fortsetzen wird zum vorübergehenden Anhalten von Skalierungsaktivitäten verwendet, die von Ihren Skalierungsrichtlinien und geplanten Aktionen ausgelöst wurden. Dies kann beispielsweise nützlich sein, wenn die automatische Skalierung Sie nicht beim Ausführen von Änderungen oder Untersuchen von Konfigurationsproblemen unterbrechen soll. Ihre Skalierungsrichtlinien und geplanten Aktionen werden beibehalten und die Skalierungsaktivitäten werden fortgesetzt werden, wenn Sie bereit sind.

In den folgenden CLI-Beispielbefehlen übergeben Sie die JSON-formatierten Parameter in einer config.json-Datei. Sie können diese Parameter auch in der Befehlszeile übergeben, indem Sie die JSON-Datenstruktur in Anführungszeichen einschließen. Weitere Informationen finden Sie unter Verwendung von Anführungszeichen bei Zeichenketten im AWS CLI in the AWS Command Line Interface User Guide.

Inhalt

- Skalierung von Aktivitäten
- Unterbrechen und Fortsetzen von Skalierungsaktivitäten

> **Note**
>
> Anweisungen zum Aussetzen von Aufskalierungsprozessen während Amazon-ECS-Bereitstellungen finden Sie in der folgenden Dokumentation:
> Service Auto Scaling und Bereitstellungen im Amazon Elastic Container Service-Entwicklerhandbuch

Skalierung von Aktivitäten

Application Auto Scaling unterstützt die Aussetzung der folgenden Skalierungsaktivitäten:

- Alle Skalierungsaktivitäten nach unten, die von einer Skalierungsrichtlinie ausgelöst werden.

- Alle Skalierungsaktivitäten nach oben, die von einer Skalierungsrichtlinie ausgelöst werden.
- Alle Skalierungen, die geplante Aktionen umfassen.

In den folgenden Beschreibungen wird erläutert, was passiert, wenn einzelne Skalierungen ausgesetzt werden. Jede davon kann unabhängig voneinander ausgesetzt und fortgesetzt werden. Je nach Grund für das Aussetzen einer Skalierungsaktivität müssen Sie möglicherweise mehrere Skalierungsaktivitäten zusammen aussetzen.

`DynamicScalingInSuspended`

- Application Auto Scaling entfernt keine Kapazität, wenn eine Zielverfolgungs-Skalierungsrichtlinie oder eine Stufenskalierungsrichtlinie ausgelöst wird. Auf diese Weise können Sie Skalierungsaktivitäten nach unten im Zusammenhang mit Skalierungsrichtlinien vorübergehend deaktivieren, ohne die Skalierungsrichtlinien oder die zugehörigen CloudWatch -Alarme löschen zu müssen. Wenn Sie die Skalierung wieder aufnehmen, bewertet Application Auto Scaling Richtlinien mit Alarmschwellenwerten, die derzeit verletzt werden.

`DynamicScalingOutSuspended`

- Application Auto Scaling fügt keine Kapazität hinzu, wenn eine Zielverfolgungs-Skalierungsrichtlinie oder eine Stufenskalierungsrichtlinie ausgelöst wird. Auf diese Weise können Sie horizontale Skalierungsaktivitäten nach oben im Zusammenhang mit Skalierungsrichtlinien vorübergehend deaktivieren, ohne die Skalierungsrichtlinien oder die zugehörigen CloudWatch -Alarme löschen zu müssen. Wenn Sie die Skalierung wieder aufnehmen, bewertet Application Auto Scaling Richtlinien mit Alarmschwellenwerten, die derzeit verletzt werden.

`ScheduledScalingSuspended`

- Application Auto Scaling initiiert nicht die Skalierungsaktionen, die für den Zeitraum der Aussetzung geplant sind. Wenn Sie die geplante Skalierung wieder aufnehmen, wertet Application Auto Scaling nur die geplanten Aktionen aus, deren Ausführungszeit noch nicht abgelaufen ist.

Unterbrechen und Fortsetzen von Skalierungsaktivitäten

Sie können einzelne Skalierungsaktivitäten oder alle Skalierungsaktivitäten für Ihr skalierbares Ziel von Application Auto Scaling aussetzen und wieder aufnehmen.

> **Note**
>
> Der Kürze halber wird in diesen Beispielen gezeigt, wie die Skalierung für eine DynamoDB-Tabelle ausgesetzt und wieder aufgenommen wird. Um ein anderes skalierbares Ziel anzugeben, geben Sie seinen Namespace in `--service-namespace`, seine skalierbare Dimension in `--scalable-dimension` und seine Ressourcen-ID in `--resource-id` an. Weitere Informationen und Beispiele für die einzelnen Services finden Sie in den Themen unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

So setzen Sie eine Skalierung aus

Öffnen Sie ein Befehlszeilenfenster und verwenden Sie den register-scalable-target-Befehl mit der `--suspended-state`-Option wie folgt.

Linux, macOS oder Unix

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table \
  --suspended-state file://config.json
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb --
scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table --
suspended-state file://config.json
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Wenn Sie nur horizontale Skalierungsaktivitäten nach unten aussetzen möchten, die von einer Skalierungsrichtlinie, ausgelöst werden, geben Sie Folgendes in der config.json-Datei an.

```
{
    "DynamicScalingInSuspended":true
```

```
}
```

Wenn Sie nur horizontale Skalierungsaktivitäten nach oben aussetzen möchten, die von einer Skalierungsrichtlinie, ausgelöst werden, geben Sie in der config.json-Datei Folgendes an.

```
{
    "DynamicScalingOutSuspended":true
}
```

Wenn Sie nur Skalierungen aussetzen möchten, die geplante Aktionen umfassen, geben Sie in der config.json-Datei Folgendes an.

```
{
    "ScheduledScalingSuspended":true
}
```

So setzen Sie alle Skalierungen aus

Verwenden Sie den register-scalable-target Befehl mit der `--suspended-state` Option wie folgt.

Linux, macOS oder Unix

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table \
  --suspended-state file://config.json
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb --
scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table --
suspended-state file://config.json
```

In diesem Beispiel wird davon ausgegangen, dass die config.json-Datei die folgenden JSON-formatierten Parameter enthält.

```
{
    "DynamicScalingInSuspended":true,
    "DynamicScalingOutSuspended":true,
    "ScheduledScalingSuspended":true
}
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Ausgesetzte Skalierungsaktivitäten anzeigen

Mit dem describe-scalable-targets-Befehl können Sie für ein skalierbares Ziel bestimmen, welche Skalierungen sich in einem ausgesetzten Zustand befinden.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scalable-targets --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table
```

Windows

```
aws application-autoscaling describe-scalable-targets --service-namespace dynamodb --
scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table
```

Es folgt eine Beispielausgabe.

```
{
    "ScalableTargets": [
        {
            "ServiceNamespace": "dynamodb",
            "ScalableDimension": "dynamodb:table:ReadCapacityUnits",
            "ResourceId": "table/my-table",
            "MinCapacity": 1,
            "MaxCapacity": 20,
            "SuspendedState": {
                "DynamicScalingOutSuspended": true,
                "DynamicScalingInSuspended": true,
                "ScheduledScalingSuspended": true
            },
            "CreationTime": 1558125758.957,
            "RoleARN": "arn:aws:iam::123456789012:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable"
```

```
        }
    ]
}
```

Wiederaufnahme der Skalierungsaktivitäten

Wenn die Skalierungsaktivität fortgesetzt werden kann, ist dies mit dem register-scalable-target-Befehl möglich.

Mit dem folgenden Beispielbefehl werden alle Skalierungen für das angegebene skalierbare Ziel fortgesetzt.

Linux, macOS oder Unix

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table \
  --suspended-state file://config.json
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb --
scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table --
suspended-state file://config.json
```

In diesem Beispiel wird davon ausgegangen, dass die config.json-Datei die folgenden JSON-formatierten Parameter enthält.

```
{
    "DynamicScalingInSuspended":false,
    "DynamicScalingOutSuspended":false,
    "ScheduledScalingSuspended":false
}
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Skalierungsaktivitäten für Application Auto Scaling

Application Auto Scaling überwacht die CloudWatch Metriken Ihrer Skalierungsrichtlinie und initiiert eine Skalierung, wenn Schwellenwerte überschritten werden. Es initiiert auch Skalierungsaktivitäten, wenn Sie die maximale oder minimale Größe des skalierbaren Ziels ändern, entweder manuell oder nach einem Zeitplan.

Wenn eine Skalierungsaktivität auftritt, führt Application Auto Scaling eine der folgenden Aktionen aus:

- Erhöht die Kapazität des skalierbaren Ziels (als Hochskalieren bezeichnet)
- Verringert die Kapazität des skalierbaren Ziels (als Herunterskalieren bezeichnet)

Sie können die Skalierungsaktivitäten der letzten sechs Wochen abrufen.

Abrufen von Skalierungsaktivitäten nach skalierbarem Ziel

Verwenden Sie den folgenden describe-scaling-activities Befehl, um die Skalierungsaktivitäten für ein bestimmtes skalierbares Ziel anzuzeigen.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scaling-activities --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-service
```

Windows

```
aws application-autoscaling describe-scaling-activities --service-namespace ecs --
scalable-dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-service
```

Im Folgenden finden Sie eine Beispielantwort, bei der `StatusCode` den aktuellen Status der Aktivität und `StatusMessage` Informationen über den Status der Skalierungsaktivität enthält.

```
{
    "ScalingActivities": [
        {
```

```
            "ScalableDimension": "ecs:service:DesiredCount",
            "Description": "Setting desired count to 1.",
            "ResourceId": "service/my-cluster/my-service",
            "ActivityId": "e6c5f7d1-dbbb-4a3f-89b2-51f33e766399",
            "StartTime": 1462575838.171,
            "ServiceNamespace": "ecs",
            "EndTime": 1462575872.111,
            "Cause": "monitor alarm web-app-cpu-lt-25 in state ALARM triggered policy
 web-app-cpu-lt-25",
            "StatusMessage": "Successfully set desired count to 1. Change successfully
 fulfilled by ecs.",
            "StatusCode": "Successful"
        }
    ]
}
```

Eine Beschreibung der Felder in der Antwort finden Sie unter ScalingActivity in der API-Referenz für Application Auto Scaling.

Die folgenden Statuscodes zeigen an, wann das Skalierungsereignis, das zur Skalierungsaktivität führt, einen abgeschlossenen Status erreicht:

- `Successful` – Die Skalierung wurde erfolgreich abgeschlossen
- `Overridden` – Die gewünschte Kapazität wurde durch ein neueres Skalierungsereignis aktualisiert
- `Unfulfilled` – Das Zeitlimit für die Skalierung wurde überschritten oder der Ziel-Service kann die Anfrage nicht erfüllen
- `Failed` – Die Skalierung ist mit einer Ausnahme fehlgeschlagen

> **Note**
>
> Die Skalierungsaktivität kann auch den Status `Pending` oder `InProgress` haben. Alle Skalierungsaktivitäten haben einen `Pending`-Status, bevor der Zielservice reagiert. Nachdem das Ziel reagiert hat, ändert sich der Status der Skalierungsaktivität zu `InProgress`.

Einbeziehung nicht skalierter Aktivitäten

Standardmäßig spiegeln die Skalierungsaktivitäten nicht die Zeiten wider, in denen Application Auto Scaling eine Entscheidung darüber trifft, ob keine Skalierung durchgeführt werden soll.

Nehmen wir beispielsweise an, dass ein Amazon-ECS-Service den maximalen Schwellenwert einer bestimmten Metrik überschreitet, aber die Anzahl der Aufgaben bereits die maximal zulässige Anzahl von Aufgaben erreicht. In diesem Fall wird Application Auto Scaling nicht die gewünschte Anzahl von Aufgaben aufskalieren.

Um Aktivitäten, die nicht skaliert sind (keine skalierten Aktivitäten), in die Antwort aufzunehmen, fügen Sie dem describe-scaling-activities Befehl die `--include-not-scaled-activities` Option hinzu.

Linux, macOS oder Unix

```
aws application-autoscaling describe-scaling-activities --include-not-scaled-activities
 \
  --service-namespace ecs --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service
```

Windows

```
aws application-autoscaling describe-scaling-activities --include-not-scaled-activities
 --service-namespace ecs --scalable-dimension ecs:service:DesiredCount --resource-
id service/my-cluster/my-service
```

> **Note**
>
> Wenn dieser Befehl einen Fehler auslöst, stellen Sie sicher, dass Sie lokal AWS CLI auf die neueste Version aktualisiert haben.

Um zu bestätigen, dass die Antwort die nicht skalierten Aktivitäten enthält, wird das `NotScaledReasons`-Element in der Ausgabe für einige, wenn nicht alle fehlgeschlagenen Skalierungsaktivitäten angezeigt.

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "ecs:service:DesiredCount",
            "Description": "Attempting to scale due to alarm triggered",
            "ResourceId": "service/my-cluster/my-service",
            "ActivityId": "4d759079-a31f-4d0c-8468-504c56e2eecf",
            "StartTime": 1664928867.915,
```

```
            "ServiceNamespace": "ecs",
            "Cause": "monitor alarm web-app-cpu-gt-75 in state ALARM triggered policy
 web-app-cpu-gt-75",
            "StatusCode": "Failed",
            "NotScaledReasons": [
                {
                    "Code": "AlreadyAtMaxCapacity",
                    "MaxCapacity": 4
                }
            ]
        }
    ]
}
```

Eine Beschreibung der Felder in der Antwort finden Sie unter ScalingActivity in der API-Referenz für Application Auto Scaling.

Wenn eine nicht skalierte Aktivität zurückgegeben wird, können abhängig vom in `Code` aufgeführten Ursachencode Attribute wie `CurrentCapacity`, `MaxCapacity` und `MinCapacity` in der Antwort vorhanden sein.

Um große Mengen doppelter Einträge zu verhindern, wird nur die erste nicht skalierte Aktivität im Skalierungsaktivitätsverlauf aufgezeichnet. Alle nachfolgenden nicht skalierten Aktivitäten generieren keine neuen Einträge, es sei denn, der Grund für die Nichtskalierung ändert sich.

Verstehen von nicht skalierten Ursachencodes

Im Folgenden sind die Ursachencodes für eine nicht skalierte Aktivität aufgeführt.

Ursachencode	Definition
AutoScalingAnticipatedFlapping	Der automatische Skalierungsalgorithmus hat beschlossen, keine Skalierungsaktion durchzuführen, da dies zu einem Flattern führen

Ursachencode	Definition			
	würde. Flattern beschreibt eine Endlosschleife aus Auf- und Abwärtsskalieren. Das heißt, wenn eine Skalierungsaktion durchgeführt wird, würde sich der Metrikwert ändern, um eine weitere Skalierungsaktion in der umgekehrten Richtung zu starten.			
TargetServicePutResourceAsUnscalable	Der Ziel-Service hat die Ressource vorübergehend in einen nicht skalierbaren Status versetzt. Application Auto Scaling wird es erneut versuchen, wenn die in der Skalierungsrichtlinie konfigurierten Bedingungen für die automatische Skalierung erfüllt sind.			

Ursachencode	Definition			
AlreadyAtMaxCapacity	Die Skalierung wird durch die von Ihnen angegebene maximale Kapazität blockiert. Wenn Application Auto Scaling aufskalieren soll, müssen Sie die maximale Kapazität erhöhen.			
AlreadyAtMinCapacity	Die Skalierung wird durch die von Ihnen angegebene minimale Kapazität blockiert. Wenn Application Auto Scaling abskalieren soll, müssen Sie die Mindestkapazität verringern.			
AlreadyAtDesiredCapacity	Der automatische Skalierungsalgorithmus berechnete die geänderte Kapazität so, dass sie der aktuellen Kapazität entspricht.			

Überwachung von Application Auto Scaling

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Application Auto Scaling und Ihren anderen AWS Lösungen. Sie sollten Überwachungsdaten von allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen Mehrpunktfehler leichter beheben können, wenn er auftritt. AWS bietet Überwachungstools, um Application Auto Scaling zu überwachen, Fehler zu melden und gegebenenfalls automatische Maßnahmen zu ergreifen.

Sie können die folgenden Funktionen nutzen, um Ihre AWS Ressourcen zu verwalten:

AWS CloudTrail

Mit AWS CloudTrail können Sie die Aufrufe an die Application Auto Scaling API durch oder im Namen Ihres AWS-Konto verfolgen. CloudTrail speichert die Informationen in Protokolldateien im Amazon S3-Bucket, den Sie angegeben haben. Sie können feststellen, welche Benutzer und Konten Application Auto Scaling aufgerufen haben, von welcher Quell-IP-Adresse die Anrufe ausgingen und wann die Anrufe erfolgten. Weitere Informationen finden Sie unter API-Aufrufe von Application Auto Scaling mit AWS CloudTrail protokollieren.

> **Note**
>
> Informationen zu anderen AWS-Services, die Sie beim Protokollieren und Sammeln von Daten über Ihre Workloads unterstützen können, finden Sie im Leitfaden zur Protokollierung und Überwachung für Anwendungseigentümer in AWS Prescriptive Guidance.

Amazon CloudWatch

Amazon CloudWatch hilft Ihnen bei der Analyse von Protokollen und bei der Überwachung der Metriken Ihrer AWS-Ressourcen und gehosteten Anwendungen in Echtzeit. Sie können Metriken erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können CloudWatch zum Beispiel die Ressourcenauslastung verfolgen lassen und Sie benachrichtigen, wenn die Auslastung sehr hoch ist oder wenn der Alarm der Metrik den Status `INSUFFICIENT_DATA` erreicht hat. Weitere Informationen finden Sie unter Ihre Ressourcen mit CloudWatch überwachen.

CloudWatch verfolgt auch AWS-API-Nutzungsmetriken für Application Auto Scaling. Sie können diese Metriken verwenden, um Alarme zu konfigurieren, die Sie benachrichtigen, wenn Ihr API-Aufrufvolumen einen von Ihnen definierten Schwellenwert überschreitet. Weitere Informationen finden Sie unter AWS-Nutzungsmetriken im Benutzerhandbuch zu Amazon CloudWatch.

Amazon EventBridge

Amazon EventBridge ist ein Serverless-Event-Bus-Service, über den Sie Ihre Anwendungen einfach mit Daten aus einer Vielzahl von Quellen verbinden können. EventBridge stellt einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, Software-as-a-Service-(SaaS)-Anwendungen und AWS-Services und leitet diese Daten dann an Ziele wie Lambda weiter. Auf diese Weise können Sie Ereignisse überwachen, die in Services auftreten, und ereignisgesteuerte Architekturen erstellen. Weitere Informationen finden Sie unter Ereignisse von Application Auto Scaling mit Amazon EventBridge überwachen.

AWS Health Dashboard

Das AWS Health Dashboard (PHD) zeigt Informationen an und stellt auch Benachrichtigungen bereit, die durch Änderungen des Zustands von AWS-Ressourcen aufgerufen werden. Diese Informationen werden auf zweierlei Weise dargestellt: in einem Dashboard, das kürzliche und kommende Ereignisse nach Kategorie sortiert anzeigt, und in einem vollständigen Ereignisprotokoll, das alle Ereignisse der letzten 90 Tage enthält. Weitere Informationen finden Sie unter AWS Health Dashboard Benachrichtigungen für Application Auto Scaling.

API-Aufrufe von Application Auto Scaling mit AWS CloudTrail protokollieren

Application Auto Scaling ist in AWS CloudTrail integriert, einem Service, der Aktionen eines Benutzers, einer Rolle oder eines AWS-Services mit Application Auto Scaling API aufzeichnet. CloudTrail erfasst alle API-Aufrufe für Application Auto Scaling als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe aus dem AWS Management Console und Code-Aufrufe an die Application Auto Scaling API. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail-Ereignissen an einen Amazon-S3-Bucket, einschließlich Ereignissen für Application Auto Scaling, aktivieren. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole trotzdem in Event history (Ereignisverlauf) anzeigen. Anhand der von CloudTrail gesammelten Informationen können Sie die Anfrage, die an Application Auto Scaling gestellt wurde, die IP-Adresse, von der die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde und weitere Details ermitteln.

Weitere Informationen zu CloudTrail finden Sie im AWS CloudTrail-Benutzerhandbuch.

Informationen von Application Auto Scaling in CloudTrail

CloudTrail wird beim Erstellen Ihres AWS-Konto für Sie aktiviert. Wenn eine Aktivität des Application Auto Scaling auftritt, wird diese Aktivität in einem CloudTrail-Ereignis zusammen mit anderen AWS-Serviceereignissen im Ereignisverlauf aufgezeichnet. Sie können die neusten Ereignisse in Ihr(em) AWS-Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Anzeigen von Ereignissen mit dem CloudTrail-API-Ereignisverlauf.

Für eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS-Konto, einschließlich Ereignissen für Application Auto Scaling, erstellen Sie einen Trail. Ein Trail ermöglicht es CloudTrail, Protokolldateien in einem Amazon-S3-Bucket bereitzustellen. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-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 Amazon Web Services konfigurieren, um die in den CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- In CloudTrail unterstützte Services und Integrationen
- Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail-Protokolldateien von mehreren Konten.

Alle Application-Auto-Scaling-Aktionen werden von CloudTrail protokolliert und in der Referenz zu Application Auto Scaling API dokumentiert. Zum Beispiel generieren Aufrufe der Aktionen `PutScalingPolicy`, `DeleteScalingPolicy` und `DescribeScalingPolicies` Einträge in den CloudTrail-Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Anhand der Identitätsinformationen zur Benutzeridentität können Sie Folgendes bestimmen:

- Ob die Anfrage mit Stammbenutzer- oder AWS Identity and Access Management (IAM)-Anmeldeinformationen ausgeführt wurde.
- Ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer ausgeführt wurde.

- Gibt an, ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter CloudTrail-Element userIdentity.

Grundlegendes zu Application-Auto-Scaling-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon-S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail-Protokolleinträge sind kein geordnetes Stack-Trace der öffentlichen API-Aufrufe und erscheinen daher in keiner bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail-Protokolleintrag, der die Aktion `DescribeScalableTargets` demonstriert.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-08-21T17:05:42Z"
            }
        }
    },
    "eventTime": "2018-08-16T23:20:32Z",
    "eventSource": "autoscaling.amazonaws.com",
    "eventName": "DescribeScalableTargets",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "72.21.196.68",
    "userAgent": "EC2 Spot Console",
    "requestParameters": {
        "serviceNamespace": "ec2",
        "scalableDimension": "ec2:spot-fleet-request:TargetCapacity",
        "resourceIds": [
            "spot-fleet-request/sfr-05ceaf79-3ba2-405d-e87b-612857f1357a"
```

```
        ]
    },
    "responseElements": null,
    "additionalEventData": {
        "service": "application-autoscaling"
    },
    "requestID": "0737e2ea-fb2d-11e3-bfd8-99133058e7bb",
    "eventID": "3fcfb182-98f8-4744-bd45-b38835ab61cb",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Zugehörige Ressourcen

CloudWatch Logs ermöglicht Ihnen die Überwachung und den Erhalt Benachrichtigungen für bestimmte Ereignisse, die von CloudTrail erfasst werden. Die an CloudWatch Logs gesendeten Ereignisse sind so konfiguriert, dass sie von Ihrem Trail protokolliert werden. Stellen Sie daher sicher, dass Sie Ihre(n) Trail(s) so konfiguriert haben, dass die Ereignisarten protokolliert werden, die Sie überwachen möchten. CloudWatch Logs kann Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Leitfaden für Amazon CloudWatch Logs und im Thema Überwachen von CloudTrail-Protokolldateien mit Amazon CloudWatch Logs im Benutzerhandbuch zu AWS CloudTrail.

Ihre Ressourcen mit CloudWatch überwachen

Dieser Abschnitt enthält Informationen zur Überwachung von Metriken für Ihre skalierbaren Ressourcen mithilfe von CloudWatch.

Themen

- Dashboards mit CloudWatch erstellen
- CloudWatch-Alarmen überwachen
- Ressourcennutzung mit CloudWatch überwachen

Dashboards mit CloudWatch erstellen

Sie können die Ressourcennutzung Ihrer Anwendung mithilfe von Amazon CloudWatch überwachen, das Metriken über Ihre Nutzung und Leistung generiert. CloudWatch sammelt Rohdaten von Ihren

AWS Ressourcen und den Anwendungen, die Sie darauf AWS ausführen, und verarbeitet sie zu lesbaren Metriken, die nahezu in Echtzeit vorliegen. Die Metriken werden für 15 Monate aufbewahrt, damit Sie auf historische Daten zugreifen können und einen besseren Überblick zur Leistung der Anwendung erhalten. Weitere Informationen finden Sie im Amazon CloudWatch-Benutzerhandbuch.

CloudWatch-Dashboards sind anpassbare Startseiten in der CloudWatch-Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können, auch solche, die über verschiedene Regionen verteilt sind. Sie können CloudWatch-Dashboards verwenden, um benutzerdefinierte Ansichten der Metriken und Alarme für Ihre AWS Ressourcen zu erstellen. Sie können die für jede Metrik in den verschiedenen Diagrammen verwendete Farbe auswählen, damit Sie eine Metrik einfach über mehrere Diagramme hinweg verfolgen können.

So erstellen Sie ein CloudWatch-Dashboard

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboard und dann Neues Dashboard erstellen aus.
3. Geben Sie einen Namen für das Dashboard ein, z. B. den Namen des Dienstes, für den Sie CloudWatch-Daten anzeigen möchten.
4. Klicken Sie auf Dashboard erstellen.
5. Wählen Sie einen Typ für das Widget aus, das dem Dashboard hinzugefügt werden soll (z. B. ein Liniendiagramm). Wählen Sie dann Konfigurieren und anschließend die Metrik aus, die Sie dem Dashboard hinzufügen möchten. Weitere Informationen finden Sie unter Hinzufügen oder Entfernen eines Diagramms aus einem CloudWatch-Dashboard im Amazon CloudWatch Benutzerhandbuch

Standardmäßig sind die Metriken, die Sie in den CloudWatch-Dashboards erstellen, Durchschnittswerte. Mit CloudWatch können Sie zwar jede Statistik für jede Metrik auswählen, aber nicht alle Kombinationen sind sinnvoll. Für die CPU-Auslastung sind beispielsweise die Statistiken „Mittelwert“, „Minimum“ und „Maximum“ sinnvoll, „Summe“ dagegen nicht.

Ein häufig verwendetes Maß für die Anwendungsleistung ist die durchschnittliche CPU-Auslastung. Wenn die CPU-Auslastung zunimmt und die Kapazität dafür nicht ausreicht, reagiert die Anwendung möglicherweise nicht mehr. Wenn auf der anderen Seite zu viel Kapazität verfügbar ist und Ressourcen bei geringer Auslastung ausgeführt werden, erhöht dies die Kosten für die Nutzung des betreffenden Services.

Je nach Service liegen außerdem Metriken vor, die den verfügbaren bereitgestellten Durchsatz verfolgen. Für die Anzahl der Aufrufe, die für einen Funktionsalias oder eine Version mit bereitgestellter Gleichzeitigkeit verarbeitet werden, gibt Lambda beispielsweise die Metrik `ProvisionedConcurrencyUtilization` aus. Wenn Sie einen großen Auftrag starten und dieselbe Funktion mehrmals gleichzeitig aufrufen, kann für den Auftrag Latenz auftreten, wenn er die Provisioned Concurrency überschreitet. Wenn Sie dagegen mehr Provisioned Concurrency haben, als Sie benötigen, sind die Kosten höher als erforderlich.

Metriken werden nicht angezeigt, bevor die Ressource vollständig eingerichtet wurde. Wenn eine Metrik in den letzten 14 Tagen keine Daten veröffentlicht hat, können Sie sie auch nicht finden, wenn Sie nach Metriken suchen, die Sie zu einem Diagramm in einem CloudWatch-Dashboard hinzufügen möchten. Informationen über das manuelle Hinzufügen von Metriken finden Sie unter Graph metrics manually on a CloudWatch dashboard im Amazon CloudWatch User Guide.

Weitere Informationen finden Sie in der Servicedokumentation in der Tabelle unter Ressourcennutzung mit CloudWatch überwachen.

CloudWatch-Alarmen überwachen

Sie können Alarme erstellen, um Sie zu benachrichtigen, wenn Amazon CloudWatch Probleme entdeckt hat, die Ihre Aufmerksamkeit erfordern könnten.

Ein CloudWatch-Alarm überwacht eine einzelne Metrik. Er ruft nur dann eine oder mehrere Aktionen auf, wenn sich der Alarmzustand ändert und für den von Ihnen angegebenen Zeitraum anhält. Sie können beispielsweise einen Alarm einstellen, der Sie benachrichtigt, wenn ein Metrikwert auf einen bestimmten Wert fällt oder diesen überschreitet, um sicherzustellen, dass Sie benachrichtigt werden, bevor ein potenzielles Problem auftritt.

Mit CloudWatch können Sie auch einen Alarm einstellen, der Sie benachrichtigt, wenn sich die Metrik im Zustand `INSUFFICIENT_DATA` befindet. Jede Metrik für einen beliebigen AWS Dienst kann einen Alarm bei `INSUFFICIENT_DATA` auslösen. Dies ist der anfängliche Zustand eines neuen Alarms, aber der Alarmzustand ändert sich auch auf `INSUFFICIENT_DATA`, wenn CloudWatch-Metriken nicht mehr verfügbar sind oder nicht genügend Daten für die Metrik verfügbar sind, um den Alarmzustand zu bestimmen. Zum Beispiel sendet AWS Lambda die Metrik `ProvisionedConcurrencyUtilization` nur dann jede Minute an CloudWatch, wenn die Lambda-Funktion aktiv ist. Wenn die Funktion nicht aktiv ist, geht der Alarm in den Zustand `INSUFFICIENT_DATA` über, während er auf die Metriken wartet. Das ist normal und muss nicht unbedingt bedeuten, dass ein Problem vorliegt, aber es könnte ein Hinweis auf ein Problem sein,

wenn Sie innerhalb einer bestimmten Zeitspanne Aktivität erwartet haben, die aber nicht eingetreten ist.

In diesem Thema wird erklärt, wie Sie einen Alarm erstellen, der eine Benachrichtigung sendet, wenn die Metrik innerhalb oder außerhalb eines von Ihnen definierten Schwellenwerts liegt oder wenn nicht genügend Daten vorhanden sind. Ausführlichere Informationen zu Alarmen finden Sie unter Verwendung von Amazon CloudWatch-Alarmen im Amazon CloudWatch Benutzerhandbuch.

So erstellen Sie einen Alarm, der eine E-Mail sendet

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms und Alarm erstellen aus.
3. Wählen Sie Metrik auswählen.

   Sie werden auf eine Seite weitergeleitet, auf der Sie alle Ihre Metriken finden können. Welche Arten von Metriken Ihnen zur Verfügung stehen, hängt von den Diensten und Funktionen ab, die Sie verwenden. Die Metriken sind zunächst nach dem Namespace des Dienstes und dann nach den verschiedenen Dimensionskombinationen innerhalb jedes Namespaces gruppiert.
4. Wählen Sie einen metrischen Namespace (z. B. Lambda) und dann eine metrische Dimension (z. B. Nach Funktionsname).

   Auf der Registerkarte Alle Metriken werden alle Metriken für die ausgewählte Dimension und den ausgewählten Namespace angezeigt.
5. Aktivieren Sie das Kontrollkästchen neben der Metrik, für die Sie einen Alarm erstellen möchten, und wählen Sie dann Metrik auswählen.
6. Konfigurieren Sie den Alarm wie folgt, und wählen Sie dann Weiter:

   - Wählen Sie unter Metrik einen Aggregationszeitraum von `1 minute` oder `5 minutes`. Wenn Sie eine Minute als Aggregationszeitraum für eine Metrik verwenden, gibt es jede Minute einen Datenpunkt. Je kürzer der Zeitraum ist, desto empfindlicher ist der Alarm.
   - Unter Bedingungen konfigurieren Sie Ihren Schwellenwert, z. B. den Wert, den die Metrik überschreiten muss, bevor eine Benachrichtigung erzeugt wird.
   - Geben Sie unter Zusätzliche Konfiguration für Datenpunkte bis Alarm die Anzahl der Datenpunkte (Auswertungszeiträume) ein, in denen der Metrikwert die Schwellenwertbedingungen erfüllen muss, um den Alarm auszulösen. So würde es bei zwei aufeinanderfolgenden Zeiträume von je 5 Minuten z. B. 10 Minuten dauern, den Alarm auszulösen.

- Behalten Sie bei Behandlung fehlender Daten die Standardeinstellung bei und behandeln Sie fehlende Datenpunkte als fehlend.

  Einige Metriken werden nur gemeldet, wenn eine Aktivität stattfindet. Dies kann zu einer spärlich gemeldeten Metrik führen. Wenn bei einer Metrik planmäßig häufig Datenpunkte fehlen, ist der Status des Alarms während dieser Zeiträume `INSUFFICIENT_DATA`. Um den Alarm zu zwingen, den vorherigen Status `ALARM` oder `OK` beizubehalten, damit die Alarme nicht ausschlagen, können Sie stattdessen die fehlenden Daten ignorieren.

7. Wählen oder erstellen Sie unter Benachrichtigung ein SNS-Thema, das Sie benachrichtigen soll, wenn der Alarm den Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` hat. Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.
8. Wenn Sie fertig sind, wählen Sie Weiter.
9. Geben Sie einen Namen und optional eine Beschreibung für den Alarm ein und wählen Sie dann Weiter.
10. Wählen Sie Alarm erstellen aus.

So prüfen Sie den Status Ihrer Alarme

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Alarme, um eine Liste der Alarme anzuzeigen.
3. Um Alarme zu filtern, verwenden Sie die Dropdown-Filter neben dem Suchfeld und wählen Sie die gewünschte Filteroption.
4. Um einen Alarm zu bearbeiten oder zu löschen, wählen Sie den Alarm aus und wählen Sie dann Aktionen, Bearbeiten oder Aktionen, Löschen.

Ressourcennutzung mit CloudWatch überwachen

Mit Amazon CloudWatch erhalten Sie einen nahezu kontinuierlichen Überblick über Ihre Anwendungen in skalierbaren Ressourcen. CloudWatch ist ein Überwachungsservice für AWS-Ressourcen. Sie können mit CloudWatch Metriken erfassen und nachverfolgen, Alarme festlegen und auf Änderungen in Ihren AWS-Ressourcen automatisch reagieren. Sie können auch Dashboards erstellen, um die spezifischen Metriken oder Gruppen von Metriken zu überwachen, die Sie benötigen.

Wenn Sie mit den Diensten interagieren, die mit Application Auto Scaling integriert sind, senden diese die in der folgenden Tabelle aufgeführten Metriken an CloudWatch. In CloudWatch werden die Metriken zunächst nach dem Dienst-Namespace und dann nach den verschiedenen Dimensionskombinationen innerhalb jedes Namespaces gruppiert. Diese Metriken können Ihnen helfen, die Ressourcennutzung zu überwachen und die Kapazität Ihrer Anwendungen zu planen. Wenn der Workload Ihrer Anwendung nicht konstant ist, sollten Sie die Verwendung von Auto Scaling in Betracht ziehen. Ausführliche Beschreibungen dieser Metriken finden Sie in der Dokumentation zur jeweiligen Metrik.

Inhalt

- CloudWatch-Metriken zur Überwachung der Ressourcennutzung
- Vordefinierte Metriken für Skalierungsrichtlinien für die Zielverfolgung

CloudWatch-Metriken zur Überwachung der Ressourcennutzung

In der folgenden Tabelle sind die CloudWatch-Metriken, Namespaces und Dimensionen aufgeführt, die zur Überwachung der Ressourcennutzung verfügbar sind. Die Liste ist nicht vollständig, bietet Ihnen aber einen guten Ausgangspunkt. Wenn Sie diese Metriken nicht in der CloudWatch-Konsole sehen, stellen Sie sicher, dass Sie die Einrichtung der Ressource abgeschlossen haben. Weitere Informationen finden Sie im Amazon CloudWatch-Benutzerhandbuch.

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
AppStream 2.0			
Flotten	AWS/AppStream	Name: AvailableCapacity Dimension: Flotte	AppStream-2.0-Metriken
Flotten	AWS/AppStream	Name: CapacityUtilization	AppStream-2.0-Metriken

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
		Dimension: Flotte	
Aurora			
Replikas	AWS/RDS	Name: CPUUtilization Dimensionen: DBClusterIdentifier, Rolle (LESER)	Aurora-Metriken auf Clusterebene
Replikas	AWS/RDS	Name: DatabaseConnections Dimensionen: DBClusterIdentifier, Rolle (LESER)	Aurora-Metriken auf Clusterebene
Amazon Comprehend			

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Dokumentklassifizierungsendpunkte	AWS/Comprehend	Name: InferenceUtilization Dimension: EndpointArn	Endpoint-Metriken für Amazon Comprehend
Endpunkte der Entitätserkennung	AWS/Comprehend	Name: InferenceUtilization Dimension: EndpointArn	Endpoint-Metriken für Amazon Comprehend
DynamoDB			
Tabellen und globale sekundäre Indizes	AWS/DynamoDB	Name: ProvisionedReadCapacityUnits Dimensionen: TableName, GlobalSecondaryIndexName	DynamoDB-Metriken

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Tabellen und globale sekundäre Indizes	AWS/DynamoDB	Name: ProvisionedWriteCapacityUnits Dimensionen: TableName, GlobalSecondaryIndexName	DynamoDB-Metriken
Tabellen und globale sekundäre Indizes	AWS/DynamoDB	Name: ConsumedReadCapacityUnits Dimensionen: TableName, GlobalSecondaryIndexName	DynamoDB-Metriken

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Tabellen und globale sekundäre Indizes	AWS/DynamoDB	Name: ConsumedWriteCapacityUnits Dimensionen: TableName, GlobalSecondaryIndexName	DynamoDB-Metriken
Amazon ECS			
Services	AWS/ECS	Name: CPUUtilization Dimensionen: ClusterName, ServiceName	Amazon-ECS-Metriken

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Services	AWS/ECS	Name: MemoryUtilization Dimensionen: ClusterName, ServiceName	Amazon-ECS-Metriken
Services	AWS/ApplicationELB	Name: RequestCountPerTarget Dimension: TargetGroup	Application-Load-Balancer-Metriken
ElastiCache			
Cluster (Replikationsgruppen)	AWS/ElastiCache	Name: DatabaseMemoryUsageCountedForEvictPercentage Dimension: ReplicationGroupId	ElastiCache-für-Redis-Metriken

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Cluster (Replikationsgruppen)	AWS/ElastiCache	Name: DatabaseCapacityUsageCountedForEvictPercentage Dimension: ReplicationGroupId	ElastiCache-für-Redis-Metriken
Cluster (Replikationsgruppen)	AWS/ElastiCache	Name: EngineCPUUtilization Dimensionen: ReplicationGroupId, Rolle (Primär)	ElastiCache-für-Redis-Metriken
Cluster (Replikationsgruppen)	AWS/ElastiCache	Name: EngineCPUUtilization Dimensionen: ReplicationGroupId, Rolle (Replikat)	ElastiCache-für-Redis-Metriken

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Amazon EMR			
Cluster	AWS/ElasticMapReduce	Name: YARNMemoryAvailablePercentage Dimension: ClusterId	Amazon-EMR-Metriken
Amazon Keyspaces			
Tabellen	AWS/Cassandra	Name: ProvisionedReadCapacityUnits Dimensionen: Keyspace, TableName	Amazon-Keyspaces-Metriken
Tabellen	AWS/Cassandra	Name: ProvisionedWriteCapacityUnits Dimensionen: Keyspace, TableName	Amazon-Keyspaces-Metriken

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Tabellen	AWS/Cassandra	Name: ConsumedReadCapacityUnits Dimensionen: Keyspace, TableName	Amazon-Keyspaces-Metriken
Tabellen	AWS/Cassandra	Name: ConsumedWriteCapacityUnits Dimensionen: Keyspace, TableName	Amazon-Keyspaces-Metriken
Lambda			
Bereitgestellte Gleichzeitigkeit	AWS/Lambda	Name: ProvisionedConcurrencyUtilization Dimensionen: FunctionName, Ressource	Lambda-Funktionsmetriken
Amazon MSK			

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Broker-Speicher	AWS/Kafka	Name: KafkaDataLogsDiskUsed Dimensionen: Clustername	Amazon-MSK-Metriken
Broker-Speicher	AWS/Kafka	Name: KafkaDataLogsDiskUsed Dimensionen: Clustername, Broker-ID	Amazon-MSK-Metriken
Neptune			
Cluster	AWS/Neptune	Name: CPUUtilization Dimensionen: DBClusterIdentifier, Rolle (LESER)	Neptune-Metriken
SageMaker			

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Endpunktvarianten	AWS/SageMaker	Name: InvocationsPerInstance Dimensionen: EndpointName, VariantName	Aufrufmetriken
Inferenzkomponenten	AWS/SageMaker	Name: InvocationsPerCopy Dimensionen: InferenceComponentName	Aufrufmetriken
Bereitgestellte Gleichzeitigkeit für einen Serverless-Endpunkt	AWS/SageMaker	Name: ServerlessProvisionedConcurrencyUtilization Dimensionen: EndpointName, VariantName	Metriken für Serverless-Endgeräte

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Amazon EC2-Spot-Flotte			
Spot Flotten	AWS/EC2Spot	Name: CPUUtilization Dimension: FleetRequestId	Metriken für Spot-Flotten
Spot Flotten	AWS/EC2Spot	Name: NetworkIn Dimension: FleetRequestId	Metriken für Spot-Flotten
Spot Flotten	AWS/EC2Spot	Name: NetWorkout Dimension: FleetRequestId	Metriken für Spot-Flotten

Skalierbare Ressource	Namespace	CloudWatch-Metrik	Link zur Dokumentation
Spot Flotten	AWS/ApplicationELB	Name: RequestCountPerTarget Dimension: TargetGroup	Application-Load-Balancer-Metriken

Vordefinierte Metriken für Skalierungsrichtlinien für die Zielverfolgung

In der folgenden Tabelle sind die vordefinierten Metriktypen aus der API-Referenz für Application Auto Scaling mit ihrem entsprechenden CloudWatch-Metriknamen aufgeführt. Jede vordefinierte Metrik stellt eine Aggregation der Werte der zugrunde liegenden CloudWatch-Metrik dar. Das Ergebnis ist die durchschnittliche Ressourcennutzung über einen Zeitraum von einer Minute, basierend auf einem Prozentsatz, sofern nicht anders angegeben. Die vordefinierten Metriken werden nur im Rahmen der Einrichtung von Skalierungsrichtlinien für die Zielverfolgung verwendet.

Weitere Informationen zu diesen Metriken finden Sie in der Dokumentation des von Ihnen verwendeten Service, die Sie in der Tabelle unter CloudWatch-Metriken zur Überwachung der Ressourcennutzung finden.

Vordefinierter Metriktyp	CloudWatch-Metrikname
AppStream 2.0	
`AppStreamAverageCapacityUtilization`	CapacityUtilization
Aurora	
`RDSReaderAverageCPUUtilization`	CPUUtilization

Vordefinierter Metriktyp	CloudWatch-Metrikname
`RDSReaderAverageDatabaseConnections`	DatabaseConnections[1]
Amazon Comprehend	
`ComprehendInferenceUtilization`	InferenceUtilization
DynamoDB	
`DynamoDBReadCapacityUtilization`	ProvisionedReadCapacityUnits, ConsumedReadCapacityUnits[2]
`DynamoDBWriteCapacityUtilization`	ProvisionedWriteCapacityUnits, ConsumedWriteCapacityUnits[2]
Amazon ECS	
`ECSServiceAverageCPUUtilization`	CPUUtilization
`ECSServiceAverageMemoryUtilization`	MemoryUtilization
`ALBRequestCountPerTarget`	RequestCountPerTarget[1]
ElastiCache	
`ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`	DatabaseMemoryUsageCountedForEvictPercentage
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`	DatabaseCapacityUsageCountedForEvictPercentage
`ElastiCachePrimaryEngineCPUUtilization`	EngineCPUUtilization
`ElastiCacheReplicaEngineCPUUtilization`	EngineCPUUtilization
Amazon Keyspaces	

Vordefinierter Metriktyp	CloudWatch-Metrikname
`CassandraReadCapacityUtilization`	ProvisionedReadCapacityUnits, ConsumedReadCapacityUnits[2]
`CassandraWriteCapacityUtilization`	ProvisionedWriteCapacityUnits, ConsumedWriteCapacityUnits[2]
Lambda	
`LambdaProvisionedConcurrencyUtilization`	ProvisionedConcurrencyUtilization
Amazon MSK	
`KafkaBrokerStorageUtilization`	KafkadatalogsDiskUsed
Neptune	
`NeptuneReaderAverageCPUUtilization`	CPUUtilization
SageMaker	
`SageMakerVariantInvocationsPerInstance`	InvocationsPerInstance[1]
`SageMakerInferenceComponentInvocationsPerCopy`	InvocationsPerCopy[1]
`SageMakerVariantProvisionedConcurrencyUtilization`	ServerlessProvisionedConcurrencyUtilization
Spot-Flotte	
`EC2SpotFleetRequestAverageCPUUtilization`	CPUUtilization[3]
`EC2SpotFleetRequestAverageNetworkIn[3]`	NetworkIn[1] [3]

Vordefinierter Metriktyp	CloudWatch-Metrikname
`EC2SpotFleetRequestAverageNetworkOut`[3]	NetworkOut[1] [3]
`ALBRequestCountPerTarget`	RequestCountPerTarget[1]

[1]Metrik basiert auf einer Anzahl statt auf einem Prozentsatz.

[2]Für DynamoDB und Amazon Keyspaces sind die vordefinierten Metriken eine Aggregation von zwei CloudWatch-Metriken, um die Skalierung auf der Grundlage des bereitgestellten Durchsatzes zu unterstützen.

[3]Für eine optimale Skalierungsleistung sollte die detaillierte Überwachung von Amazon EC2 verwendet werden.

Ereignisse von Application Auto Scaling mit Amazon EventBridge überwachen

Mit Amazon EventBridge, vormals CloudWatch Events, können Sie Ereignisse überwachen, die spezifisch für Application Auto Scaling sind, und Zielaktionen initiieren, die andere AWS-Services nutzen. Ereignisse von AWS-Services werden in EventBridge nahezu in Echtzeit bereitgestellt.

Mit EventBridge können Sie Regeln erstellen, die eingehenden Ereignissen entsprechen und sie zur Verarbeitung an Ziele weiterleiten.

Weitere Informationen finden Sie unter Erste Schritte mit Amazon EventBridge im Amazon EventBridge Benutzerhandbuch.

Application Auto Scaling-Ereignisse

Die folgenden Beispiele zeigen Ereignisse für Application Auto Scaling. Ereignisse werden auf die bestmögliche Weise ausgegeben.

Nur Ereignisse, die spezifisch für „Scaled to max“ und API-Aufrufe über CloudTrail sind derzeit für „Application Auto Scaling“ verfügbar.

Ereignistypen

- Ereignis für Statusänderung: skaliert auf Maximum

- Ereignisse für API-Aufrufe über CloudTrail

Ereignis für Statusänderung: skaliert auf Maximum

Das folgende Beispielereignis zeigt, dass Application Auto Scaling die Kapazität des skalierbaren Ziels auf seine maximale Größe erhöhte (aufskalierte). Wenn die Anforderungen erneut zunehmen, wird verhindert, dass Application Auto Scaling das Ziel noch weiter skaliert, da es bereits auf seine maximale Größe skaliert ist.

Im Objekt `detail` identifizieren die Werte für die Attribute `resourceId`, `serviceNamespace` und `scalableDimension` das skalierbare Ziel. Die Werte für die Attribute `newDesiredCapacity` und `oldDesiredCapacity` beziehen sich auf die neue Kapazität nach dem Aufskalierungsereignis und die ursprüngliche Kapazität vor dem Aufskalierungsereignis. Bei `maxCapacity` handelt es sich um die maximale Größe des skalierbaren Ziels.

```
{
  "version": "0",
  "id": "11112222-3333-4444-5555-666677778888",
  "detail-type": "Application Auto Scaling Scaling Activity State Change",
  "source": "aws.application-autoscaling",
  "account": "123456789012",
  "time": "2019-06-12T10:23:40Z",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "startTime": "2022-06-12T10:20:43Z",
    "endTime": "2022-06-12T10:23:40Z",
    "newDesiredCapacity": 8,
    "oldDesiredCapacity": 5,
    "minCapacity": 2,
    "maxCapacity": 8,
    "resourceId": "table/my-table",
    "scalableDimension": "dynamodb:table:WriteCapacityUnits",
    "serviceNamespace": "dynamodb",
    "statusCode": "Successful",
    "scaledToMax": true,
    "direction": "scale-out"
}
```

Um eine Regel zu erstellen, die alle `scaledToMax`-Statusänderungsereignisse für alle skalierbaren Ziele erfasst, verwenden Sie das folgende beispielhafte Ereignismuster.

```
{
  "source": [
    "aws.application-autoscaling"
  ],
  "detail-type": [
    "Application Auto Scaling Scaling Activity State Change"
  ],
  "detail": {
    "scaledToMax": [
      true
    ]
  }
}
```

Ereignisse für API-Aufrufe über CloudTrail

Ein Trail ist eine Konfiguration, mit deren Hilfe AWS CloudTrail Ereignisse als Protokolldateien an einen Amazon S3-Bucket übermittelt. CloudTrail-Protokolldateien können Protokolleinträge enthalten. Ein Ereignis stellt einen Protokolleintrag dar und enthält Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie Anforderungsparameter. Informationen zu den ersten Schritten mit CloudTrail finden Sie unter Erstellen eines Trails im Benutzerhandbuch von AWS CloudTrail.

Über CloudTrail bereitgestellte Ereignisse weisen `AWS API Call via CloudTrail` als Wert für `detail-type` auf.

Das folgende Beispielereignis stellt einen CloudTrail-Protokolldateieintrag dar, der zeigt, dass ein Konsolenbenutzer die Aktion RegisterScalableTarget von Application Auto Scaling aufgerufen hat.

```
{
  "version": "0",
  "id": "99998888-7777-6666-5555-444433332222",
  "detail-type": "AWS API Call via CloudTrail",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2022-07-13T16:50:15Z",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "eventVersion": "1.08",
    "userIdentity": {
      "type": "IAMUser",
```

```
      "principalId": "123456789012",
      "arn": "arn:aws:iam::123456789012:user/Bob",
      "accountId": "123456789012",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "sessionContext": {
        "sessionIssuer": {
          "type": "Role",
          "principalId": "123456789012",
          "arn": "arn:aws:iam::123456789012:role/Admin",
          "accountId": "123456789012",
          "userName": "Admin"
        },
        "webIdFederationData": {},
        "attributes": {
          "creationDate": "2022-07-13T15:17:08Z",
          "mfaAuthenticated": "false"
        }
      }
    },
    "eventTime": "2022-07-13T16:50:15Z",
    "eventSource": "autoscaling.amazonaws.com",
    "eventName": "RegisterScalableTarget",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "EC2 Spot Console",
    "requestParameters": {
      "resourceId": "spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "serviceNamespace": "ec2",
      "scalableDimension": "ec2:spot-fleet-request:TargetCapacity",
      "minCapacity": 2,
      "maxCapacity": 10
    },
    "responseElements": null,
    "additionalEventData": {
      "service": "application-autoscaling"
    },
    "requestID": "e9caf887-8d88-11e5-a331-3332aa445952",
    "eventID": "49d14f36-6450-44a5-a501-b0fdcdfaeb98",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
```

```
  }
}
```

Um eine Regel auf der Basis von allen DeleteScalingPolicy- und DeregisterScalableTarget-API-Aufrufen für alle skalierbaren Ziele zu erstellen, verwenden Sie das folgende beispielhafte Ereignismuster:

```
{
  "source": [
    "aws.autoscaling"
  ],
  "detail-type": [
    "AWS API Call via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "autoscaling.amazonaws.com"
    ],
    "eventName": [
       "DeleteScalingPolicy",
       "DeregisterScalableTarget"
    ],
    "additionalEventData": {
      "service": [
        "application-autoscaling"
      ]
    }
  }
}
```

Weitere Informationen zur Verwendung von CloudTrail finden Sie unter API-Aufrufe von Application Auto Scaling mit AWS CloudTrail protokollieren.

AWS Health Dashboard Benachrichtigungen für Application Auto Scaling

Um Sie bei der Verwaltung fehlgeschlagener Skalierungsereignisse zu unterstützen, bietet Ihr AWS Health Dashboard Unterstützung für Benachrichtigungen, die von Application Auto Scaling ausgegeben werden. Derzeit sind nur Skalierungsereignisse verfügbar, die für Ihre DynamoDB-Ressourcen spezifisch sind.

Das AWS Health Dashboard ist Bestandteil des AWS Health-Service. Sie benötigt keine Einrichtung und kann von jedem Benutzer angezeigt werden, der in Ihrem Konto authentifiziert ist. Weitere Informationen finden Sie unter Erste Schritte mit dem AWS Health Dashboard.

Wenn Ihre DynamoDB-Ressourcen aufgrund der Kontingentgrenzen des DynamoDB-Dienstes nicht skaliert werden können, erhalten Sie eine Meldung ähnlich der folgenden. Wenn Sie diese Meldung erhalten, muss sie als Alarm behandelt werden, der geeignete Maßnahmen erforderlich macht.

```
  Hello,

  A scaling action has attempted to scale out your DynamoDB resources in the
  eu-west-1 region. This operation has been prevented because it would have exceeded a
  table-level write throughput limit (Provisioned mode). This limit restricts the
  provisioned write capacity of the table and all of its associated global secondary
  indexes. To address the issue, refer to the Amazon DynamoDB Developer Guide for
  current limits and how to request higher limits [1].

  To identify your DynamoDB resources that are impacted, use the
  describe-scaling-activities command or the DescribeScalingActivities operation [2]
[3].
  Look for a scaling activity with StatusCode "Failed" and a StatusMessage similar to
  "Failed to set write capacity units to 45000. Reason: The requested
 WriteCapacityUnits,
  45000, is above the per table maximum for the account in eu-west-1. Per table
 maximum:
  40000." You can also view these scaling activities from the Capacity tab of your
 tables
  in the AWS Management Console for DynamoDB.

  We strongly recommend that you address this issue to ensure that your tables
  are prepared to handle increases in traffic. This notification is sent only once in
  each 12 hour period, even if another failed scaling action occurs.

  [1] https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/
Limits.html#default-limits-throughput-capacity-modes
  [2] https://docs.aws.amazon.com/cli/latest/reference/application-autoscaling/
describe-scaling-activities.html
  [3] https://docs.aws.amazon.com/autoscaling/application/APIReference/
API_DescribeScalingActivities.html

  Sincerely,
  Amazon Web Services
```

Tagging-Unterstützung für Auto Scaling von Anwendungen

Sie können das AWS CLI oder ein SDK verwenden, um skalierbare Ziele von Application Auto Scaling zu markieren. Skalierbare Ziele sind die Entitäten, die die AWS oder benutzerdefinierten Ressourcen darstellen, die Application Auto Scaling skalieren kann.

Jedes Tag ist ein Label, das aus einem benutzerdefinierten Schlüssel und einem Wert besteht, der über die API von Application Auto Scaling definiert wird. Mithilfe von Tags können Sie den Zugriff auf bestimmte skalierbare Ziele entsprechend den Anforderungen Ihres Unternehmens granular konfigurieren. Weitere Informationen finden Sie unter ABAC mit Application Auto Scaling.

Sie können Tags zu neuen skalierbaren Zielen hinzufügen, wenn Sie diese registrieren, oder Sie können sie zu vorhandenen skalierbaren Zielen hinzufügen.

Zu den häufig verwendeten Befehlen für die Verwaltung von Tags gehören:

- register-scalable-target, um neue skalierbare Ziele zu markieren, wenn Sie sie registrieren.
- tag-resource zum Hinzufügen von Tags zu einem vorhandenen skalierbaren Ziel.
- list-tags-for-resource, um die Tags für ein skalierbares Ziel zurückzugeben.
- untag-resource um ein Tag zu löschen.

Beispiel für eine Markierung

Verwenden Sie den Befehl register-scalable-target mit der Option `--tags` wie folgt. In diesem Beispiel wird ein skalierbares Ziel mit zwei Tags markiert: einem Tag-Schlüssel mit dem Namen **`environment`** mit dem Tag-Wert von **`production`** und einem Tag-Schlüssel mit dem Namen **`iscontainerbased`** mit dem Tag-Wert von **`true`**.

Ersetzen Sie die Beispielwerte für `--min-capacity` und `--max-capacity` und den Beispieltext für `--service-namespace` mit dem Namespace des AWS-Services, den Sie mit Application Auto Scaling verwenden, `--scalable-dimension` mit der skalierbaren Dimension im Zusammenhang mit der Ressource, die Sie registrieren, `--resource-id` mit einem Bezeichner für die Ressource. Weitere Informationen und Beispiele für die einzelnen Services finden Sie in den Themen unter AWS -Services, die Sie mit Application Auto Scaling verwenden können.

```
aws application-autoscaling register-scalable-target \
  --service-namespace namespace \
```

```
  --scalable-dimension dimension \
  --resource-id identifier \
  --min-capacity 1 --max-capacity 10 \
  --tags environment=production,iscontainerbased=true
```

Bei Erfolg gibt dieser Befehl den ARN des skalierbaren Ziels zurück.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

> **Note**
>
> Wenn dieser Befehl einen Fehler auslöst, vergewissern Sie sich, dass Sie die AWS CLI lokal auf die neueste Version aktualisiert haben.

Tags für Sicherheit

Verwenden Sie Tags, um zu überprüfen, ob der Anforderer (z. B. ein IAM-Benutzer oder eine Rolle) die Berechtigung hat, bestimmte Aktionen durchzuführen. Geben Sie Tag-Informationen im Bedingungselement einer IAM-Richtlinie mithilfe eines oder mehrerer der folgenden Bedingungsschlüssel an:

- Verwenden Sie `aws:ResourceTag/tag-key: tag-value`, um Benutzeraktionen für skalierbare Ziele mit bestimmten Tags zuzulassen (oder zu verweigern).
- Schreiben Sie mit `aws:RequestTag/tag-key: tag-value` vor, dass in einer Anforderung ein bestimmtes Tag vorhanden (oder nicht vorhanden) sein muss.
- Schreiben Sie mit `aws:TagKeys [tag-key, ...]` vor, dass in einer Anforderung bestimmte Tag-Schlüssel vorhanden (oder nicht vorhanden) sein müssen.

Die folgende IAM-Richtlinie erteilt beispielsweise dem Benutzer Berechtigungen für die folgenden Aktionen: `DeregisterScalableTarget`, `DeleteScalingPolicy` und `DeleteScheduledAction`. Es lehnt die Aktionen jedoch auch dann ab, wenn das skalierbare Ziel, auf die die Aktion abzielt, über das Tag **`environment=production`** verfügt.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeleteScheduledAction"
            ],
            "Resource": "*"
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeleteScheduledAction"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/environment": "production"}
            }
        }
    ]
}
```

Steuern des Zugriffs auf Tags

Verwenden Sie Tags, um zu überprüfen, ob der Anforderer (z. B. ein IAM-Benutzer oder eine IAM-Rolle) über Berechtigungen zum Hinzufügen, Ändern oder Löschen von Tags für skalierbare Ziele verfügt.

Sie könnten beispielsweise eine IAM-Richtlinie erstellen, die nur das Entfernen des Tags mithilfe des **`temporary`**-Schlüssels aus skalierbaren Zielen erlaubt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "application-autoscaling:UntagResource",
```

```
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": { "aws:TagKeys": ["temporary"] }
            }
        }
    ]
}
```

Sicherheit bei Application Auto Scaling

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud 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 und . Weitere Informationen zu den Compliance-Programmen, die für Application Auto Scaling gelten, finden Sie unter AWS Services im Umfang nach Compliance-Programm AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, einschließlich der 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 geteilten Verantwortung bei der Verwendung von Application Auto Scaling anwenden. In den folgenden Themen erfahren Sie, wie Sie Application Auto Scaling so konfigurieren, dass Ihre Sicherheits- und Compliance-Ziele erreicht werden. Sie lernen auch, wie Sie andere AWS Dienste verwenden können, mit denen Sie Ihre Application Auto Scaling Scaling-Ressourcen überwachen und sichern können.

Inhalt

- Application Auto Scaling und Datensicherung
- Identity and Access Management für Application Auto Scaling
- Application Auto Scaling und VPC-Endpunkte als Schnittstelle
- Ausfallsicherheit bei Application Auto Scaling
- Sicherheit der Infrastruktur bei Application Auto Scaling
- Compliance-Validierung für Application Auto Scaling

Application Auto Scaling und Datensicherung

Das AWS Modell der gilt für den Datenschutz in Application Auto Scaling. Wie in diesem Modell beschrieben, AWS ist es für den Schutz der globalen Infrastruktur verantwortlich, 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 gilt auch, wenn Sie mit Application Auto Scaling oder anderen Anwendungen arbeiten und die Konsole, die API oder AWS SDKs AWS-Services verwenden. AWS CLI 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.

Identity and Access Management für Application Auto Scaling

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) ist, Application Auto Scaling-Ressourcen zu nutzen. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Um Application Auto Scaling verwenden zu können, benötigen Sie eine AWS-Konto und Ihre Sicherheitsanmeldedaten, um sich bei Ihrem Konto anzumelden. Weitere Informationen finden Sie unter Einrichten, um Application Auto Scaling zu verwenden.

Eine umfassende IAM-Dokumentation finden Sie im IAM User Guide.

Zugriffskontrolle

Sie können über gültige Anmeldedaten verfügen, um Ihre Anfragen zu authentifizieren, aber ohne die entsprechenden Berechtigungen können Sie keine Ressourcen für Application Auto Scaling erstellen oder darauf zugreifen. Beispielsweise müssen Sie über Berechtigungen zum Erstellen von Skalierungsrichtlinien, zum Konfigurieren der geplanten Skalierung usw. verfügen.

In den folgenden Abschnitten erfahren Sie, wie ein IAM-Administrator IAM verwenden kann, um Ihre AWS Ressourcen zu schützen, indem er steuert, wer Application Auto Scaling Scaling-API-Aktionen ausführen kann.

Inhalt

- Wie Application Auto Scaling mit IAM funktioniert
- AWS verwaltete Richtlinien für Application Auto Scaling
- Servicegebundene Rollen für Application Auto Scaling
- Beispiele für identitätsbasierte Richtlinien für Application Auto Scaling
- Fehlerbehebung beim Zugriff auf Application Auto Scaling
- Validierung von Berechtigungen für API-Aufrufe auf Zielressourcen

Wie Application Auto Scaling mit IAM funktioniert

> **Note**
>
> Im Dezember 2017 gab es ein Update für Application Auto Scaling, das mehrere dienstverknüpfte Rollen für integrierte Dienste von Application Auto Scaling ermöglichte. Damit Benutzer die Skalierung konfigurieren können, sind spezielle IAM-Berechtigungen und eine mit dem Service Application Auto Scaling verknüpfte Rolle (oder eine Service-Rolle für Amazon EMR Auto Scaling) erforderlich.

Bevor Sie IAM verwenden, um den Zugriff auf Application Auto Scaling zu verwalten, lernen Sie, welche IAM-Funktionen für die Verwendung mit Application Auto Scaling verfügbar sind.

IAM-Features, die mit Application Auto Scaling verwendet werden können

IAM-Feature	Unterstützung für Auto Scaling von Anwendungen
Identitätsbasierte Richtlinien	Ja
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
Ressourcenbasierte Richtlinien	Nein
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Ja

Einen allgemeinen Überblick darüber, wie Application Auto Scaling und andere Funktionen mit den meisten IAM-Funktionen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Funktionen mit IAM.

Application Auto Scaling identitätsbasierte Richtlinien

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 Richtlinien für Application Auto Scaling

Beispiele für identitätsbasierte Richtlinien von Application Auto Scaling finden Sie unter Beispiele für identitätsbasierte Richtlinien für Application Auto Scaling.

Aktionen

Unterstützt Richtlinienaktionen	Ja

In einer IAM-Richtlinienanweisung können Sie jede API-Aktion von jedem Service, der IAM unterstützt, angeben. Bei Application Auto Scaling setzen Sie folgendes Präfix vor den Namen der API-Aktion: `application-autoscaling:`. Beispiel: `application-autoscaling:RegisterScalableTarget`, `application-autoscaling:PutScalingPolicy` und `application-autoscaling:DeregisterScalableTarget`.

Um mehrere Aktionen in einer einzelnen Anweisung anzugeben, trennen Sie sie durch Beistriche, wie im folgenden Beispiel gezeigt.

```
"Action": [
      "application-autoscaling:DescribeScalingPolicies",
      "application-autoscaling:DescribeScalingActivities"
```

Sie können auch Platzhalter (*) verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "application-autoscaling:Describe*"
```

Eine Liste der Application Auto Scaling-Aktionen finden Sie unter Von AWS Application Auto Scaling definierte Aktionen in der Service Authorization Reference.

Ressourcen

Unterstützt Richtlinienressourcen	Ja

In einer IAM-Richtlinienanweisung gibt das `Resource`-Element das Objekt oder die Objekte an, für die die Anweisung gilt. Für Application Auto Scaling gilt jede IAM-Richtlinienanweisung für die skalierbaren Ziele, die Sie über ihre Amazon-Ressourcennamen (ARN) (ARNs) angeben.

Das ARN-Ressourcenformat für skalierbare Ziele:

```
arn:aws:application-autoscaling:region:account-id:scalable-target/unique-identifier
```

Sie können zum Beispiel ein bestimmtes skalierbares Ziel in Ihrer Anweisung mit seinem ARN wie folgt angeben. Die eindeutige ID (1234abcd56ab78cd901ef1234567890ab123) ist ein Wert, der dem skalierbaren Ziel von Application Auto Scaling zugewiesen wird.

```
"Resource": "arn:aws:application-autoscaling:us-east-1:123456789012:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
```

Sie können alle Instances angeben, die zu einem bestimmten Konto gehören, indem Sie den eindeutigen Bezeichner wie folgt durch einen Platzhalter (*) ersetzen.

```
"Resource": "arn:aws:application-autoscaling:us-east-1:123456789012:scalable-target/*"
```

Wenn Sie alle Ressourcen angeben möchten oder wenn eine bestimmte API-Aktion keine ARNs unterstützt, verwenden Sie ein Platzhalterzeichen (*) wie folgt im `Resource`-Element.

```
"Resource": "*"
```

Weitere Informationen finden Sie unter Von AWS Application Auto Scaling definierte Ressourcentypen in der Service Authorization Reference.

Bedingungsschlüssel

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja

Sie können in den IAM-Richtlinien Bedingungen festlegen, die den Zugriff auf Application Auto Scaling-Ressourcen steuern. Die Richtlinienanweisung ist nur wirksam, wenn diese Bedingungen erfüllt sind.

Application Auto Scaling unterstützt die folgenden servicedefinierten Bedingungsschlüssel, die Sie in identitätsbasierten Richtlinien verwenden können, um zu bestimmen, wer Application Auto Scaling-API-Aktionen durchführen darf.

- `application-autoscaling:scalable-dimension`
- `application-autoscaling:service-namespace`

Informationen zu den API-Aktionen von Application Auto Scaling, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von AWS Application Auto Scaling definierte Aktionen in der Service Authorization Reference. Weitere Informationen zur Verwendung von Application Auto Scaling-Bedingungsschlüsseln finden Sie unter Bedingungsschlüssel für AWS Application Auto Scaling.

Informationen zu den globalen Bedingungsschlüsseln, die für alle Dienste verfügbar sind, finden Sie unter globalen Bedingungskontextschlüsseln für AWS im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Unterstützt ressourcenbasierte Richtlinien	Nein

Andere AWS Dienste, wie Amazon Simple Storage Service, unterstützen ressourcenbasierte Berechtigungsrichtlinien. Beispielsweise können Sie einem S3-Bucket eine Berechtigungsrichtlinie zuweisen, um die Zugriffsberechtigungen für diesen Bucket zu verwalten.

Application Auto Scaling unterstützt keine ressourcenbasierten Richtlinien.

Zugriffssteuerungslisten (ACLs)

Unterstützt ACLs	Nein

Application Auto Scaling unterstützt keine Access Control Lists (ACLs).

ABAC mit Application Auto Scaling

Unterstützt ABAC (Tags in Richtlinien)	Teilweise

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.

ABAC ist für Ressourcen möglich, die Tags unterstützen. Tags werden jedoch nicht von allen Ressourcen unterstützt. Geplante Aktionen und Skalierungsrichtlinien unterstützen keine Tags, aber

skalierbare Ziele unterstützen Tags. Weitere Informationen finden Sie unter Tagging-Unterstützung für Auto Scaling von Anwendungen.

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.

Verwendung temporärer Anmeldeinformationen mit Application Auto Scaling

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.

Servicerollen

Unterstützt Servicerollen	Ja

Wenn Ihr Amazon EMR-Cluster automatische Skalierung verwendet, ermöglicht dieses Feature Application Auto Scaling, eine Service-Rolle in Ihrem Namen zu übernehmen. Ähnlich wie bei einer serviceverknüpften Rolle ermöglicht eine Servicerolle dem Service den Zugriff auf Ressourcen in

anderen Services, um eine Aktion in Ihrem Namen durchzuführen. Servicerollen werden in Ihrem IAM-Konto angezeigt und gehören zum Konto. Dies bedeutet, dass ein IAM-Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktionalität des Dienstes beeinträchtigen.

Application Auto Scaling unterstützt nur Service-Rollen für Amazon EMR. Die Dokumentation für die EMR-Service-Rolle finden Sie unter Verwendung der automatischen Skalierung mit einer benutzerdefinierten Richtlinie für Instance-Gruppe im Amazon EMR Management Leitfaden.

> **Note**
>
> Mit der Einführung von serviceverknüpften Rollen sind mehrere Legacy-Servicerollen nicht mehr erforderlich, beispielsweise für Amazon ECS und Spot-Flotte.

Service-verknüpfte Rollen

Unterstützt serviceverknüpfte Rollen	Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer AWS-Service verknüpft ist. 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.

Weitere Informationen zu serviceverknüpften Rollen für Application Auto Scaling finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

AWS verwaltete Richtlinien für Application Auto Scaling

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie 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: AppStream 2.0 und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingAppStreamFleetPolicy

Diese Richtlinie ist der dienstbezogenen Rolle zugeordnet, die benannt wurde AWSServiceRoleForApplicationAutoScaling_AppStreamFleet, damit Application Auto Scaling Amazon anrufen AppStream CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für alle zugehörigen Ressourcen durchzuführen („Ressource“: „*“):

- Aktion: `appstream:DescribeFleets`
- Aktion: `appstream:UpdateFleet`
- Aktion: `cloudwatch:DescribeAlarms`
- Aktion: `cloudwatch:PutMetricAlarm`
- Aktion: `cloudwatch:DeleteAlarms`

AWS verwaltete Richtlinie: Aurora und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingRDSClusterPolicy

Diese Richtlinie ist der serviceverknüpften Rolle zugeordnet, die benannt wurde AWSServiceRoleForApplicationAutoScaling_RDSCluster, damit Application Auto Scaling Aurora aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für alle zugehörigen Ressourcen durchzuführen („Ressource“: „*“):

- Aktion: `rds:AddTagsToResource`
- Aktion: `rds:CreateDBInstance`
- Aktion: `rds:DeleteDBInstance`
- Aktion: `rds:DescribeDBClusters`
- Aktion: `rds:DescribeDBInstance`
- Aktion: `cloudwatch:DescribeAlarms`
- Aktion: `cloudwatch:PutMetricAlarm`
- Aktion: `cloudwatch:DeleteAlarms`

AWS verwaltete Richtlinie: Amazon Comprehend und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingComprehendEndpointPolicy

Diese Richtlinie ist der dienstbezogenen Rolle zugeordnet, die benannt wurde AWSServiceRoleForApplicationAutoScaling_ComprehendEndpoint, damit Application Auto Scaling Amazon Comprehend aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für alle zugehörigen Ressourcen durchzuführen („Ressource“: „*“):

- Aktion: `comprehend:UpdateEndpoint`
- Aktion: `comprehend:DescribeEndpoint`
- Aktion: `cloudwatch:DescribeAlarms`
- Aktion: `cloudwatch:PutMetricAlarm`
- Aktion: `cloudwatch:DeleteAlarms`

AWS verwaltete Richtlinie: DynamoDB und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingDynamoDBTablePolicy

Diese Richtlinie ist der dienstbezogenen Rolle mit dem Namen zugeordnet, AWSServiceRoleForApplicationAutoScaling_DynamoDBTabledamit Application Auto Scaling DynamoDBand aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für alle zugehörigen Ressourcen durchzuführen („Ressource“: „*“):

- Aktion: `dynamodb:DescribeTable`
- Aktion: `dynamodb:UpdateTable`
- Aktion: `cloudwatch:DescribeAlarms`
- Aktion: `cloudwatch:PutMetricAlarm`
- Aktion: `cloudwatch:DeleteAlarms`

AWS verwaltete Richtlinie: Amazon ECS und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingECSServicePolicy

Diese Richtlinie ist der dienstbezogenen Rolle zugeordnet, die benannt wurde AWSServiceRoleForApplicationAutoScaling_ECSService, damit Application Auto Scaling Amazon ECS aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für alle zugehörigen Ressourcen durchzuführen („Ressource“: „*“):

- Aktion: `ecs:DescribeServices`
- Aktion: `ecs:UpdateService`
- Aktion: `cloudwatch:DescribeAlarms`
- Aktion: `cloudwatch:PutMetricAlarm`
- Aktion: `cloudwatch:DeleteAlarms`

AWS verwaltete Richtlinie: ElastiCache und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingElastiCacheRGPolicy

Diese Richtlinie ist der dienstbezogenen Rolle mit dem Namen zugeordnet AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG, damit Application Auto Scaling die Skalierung in Ihrem Namen aufrufen ElastiCache CloudWatch und die Skalierung durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `elasticache:DescribeReplicationGroups` auf alle -Ressourcen.
- Aktion: `elasticache:ModifyReplicationGroupShardConfiguration` auf alle -Ressourcen.
- Aktion: `elasticache:IncreaseReplicaCount` auf alle -Ressourcen.
- Aktion: `elasticache:DecreaseReplicaCount` auf alle -Ressourcen.
- Aktion: `elasticache:DescribeCacheClusters` auf alle -Ressourcen.
- Aktion: `elasticache:DescribeCacheParameters` auf alle -Ressourcen.
- Aktion: `cloudwatch:DescribeAlarms` auf alle -Ressourcen.
- Aktion:`cloudwatch:PutMetricAlarm` auf die Ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`
- Aktion:`cloudwatch:DeleteAlarms` auf die Ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`
- Aktion: `cloudwatch:DeleteAlarms`

AWS verwaltete Richtlinie: Amazon Keyspaces und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingCassandraTablePolicy

Diese Richtlinie ist der serviceverknüpften Rolle zugeordnet, die benannt wurde AWSServiceRoleForApplicationAutoScaling_CassandraTable, damit Application Auto Scaling Amazon Keyspaces aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `cassandra:Select` für die folgenden Ressourcen:
  - `arn:*:cassandra:*:*:/keyspace/system/table/*`
  - `arn:*:cassandra:*:*:/keyspace/system_schema/table/*`
  - `arn:*:cassandra:*:*:/keyspace/system_schema_mcs/table/*`

- Aktion: `cassandra:Alter` auf alle -Ressourcen.
- Aktion: `cloudwatch:DescribeAlarms` auf alle -Ressourcen.
- Aktion: `cloudwatch:PutMetricAlarm` auf alle -Ressourcen.
- Aktion: `cloudwatch:DeleteAlarms` auf alle -Ressourcen.

AWS verwaltete Richtlinie: Lambda und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingLambdaConcurrencyPolicy

Diese Richtlinie ist der dienstbezogenen Rolle zugeordnet, die benannt ist AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency, damit Application Auto Scaling Lambda aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für alle zugehörigen Ressourcen durchzuführen („Ressource“: „*“):

- Aktion: `lambda:PutProvisionedConcurrencyConfig`
- Aktion: `lambda:GetProvisionedConcurrencyConfig`
- Aktion: `lambda:DeleteProvisionedConcurrencyConfig`
- Aktion: `cloudwatch:DescribeAlarms`
- Aktion: `cloudwatch:PutMetricAlarm`
- Aktion: `cloudwatch:DeleteAlarms`

AWS verwaltete Richtlinie: Amazon MSK und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingKafkaClusterPolicy

Diese Richtlinie ist der serviceverknüpften Rolle zugeordnet, die benannt wurde AWSServiceRoleForApplicationAutoScaling_KafkaCluster, damit Application Auto Scaling Amazon MSK aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für alle zugehörigen Ressourcen durchzuführen („Ressource“: „*“):

- Aktion: `kafka:DescribeCluster`
- Aktion: `kafka:DescribeClusterOperation`
- Aktion: `kafka:UpdateBrokerStorage`
- Aktion: `cloudwatch:DescribeAlarms`
- Aktion: `cloudwatch:PutMetricAlarm`
- Aktion: `cloudwatch:DeleteAlarms`

AWS verwaltete Richtlinie: Neptune und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingNeptuneClusterPolicy

Diese Richtlinie ist der dienstbezogenen Rolle zugeordnet, die benannt wurde AWSServiceRoleForApplicationAutoScaling_NeptuneCluster, damit Application Auto Scaling Neptune aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `rds:ListTagsForResource` auf alle -Ressourcen.
- Aktion: `rds:DescribeDBInstances` auf alle -Ressourcen.
- Aktion: `rds:DescribeDBClusters` auf alle -Ressourcen.
- Aktion: `rds:DescribeDBClusterParameters` auf alle -Ressourcen.
- Aktion: `cloudwatch:DescribeAlarms` auf alle -Ressourcen.
- Aktion: `rds:AddTagsToResource` auf Ressourcen mit dem Präfix autoscaled-reader in der Amazon Neptune Datenbank-Engine (`"Condition":{"StringEquals": {"rds:DatabaseEngine":"neptune"}}`)
- Aktion: `rds:CreateDBInstance` auf Ressourcen mit dem Präfix autoscaled-reader in allen DB-Clustern (`"Resource":"arn:*:rds:*:*:db:autoscaled-reader*", "arn:aws:rds:*:*:cluster:*"`) in der Amazon Neptune-Datenbank-Engine (`"Condition": {"StringEquals":{"rds:DatabaseEngine":"neptune"}}`)
- Aktion:`rds:DeleteDBInstance` auf die Ressource `arn:aws:rds:*:*:db:autoscaled-reader*`

- Aktion:`cloudwatch:PutMetricAlarm` auf die Ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`
- Aktion:`cloudwatch:DeleteAlarms` auf Ressource`arn:aws:cloudwatch:*:*:alarm:TargetTracking*`

AWS verwaltete Richtlinie: SageMaker und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingSageMakerEndpointPolicy

Diese Richtlinie ist der dienstbezogenen Rolle mit dem Namen zugeordnet AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint, damit Application Auto Scaling die Skalierung in Ihrem Namen aufrufen SageMaker CloudWatch und die Skalierung durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `sagemaker:DescribeEndpoint` auf alle -Ressourcen.
- Aktion: `sagemaker:DescribeEndpointConfig` auf alle -Ressourcen.
- Aktion: `sagemaker:DescribeInferenceComponent` auf alle -Ressourcen.
- Aktion: `sagemaker:UpdateEndpointWeightsAndCapacities` auf alle -Ressourcen.
- Aktion: `sagemaker:UpdateInferenceComponentRuntimeConfig` auf alle -Ressourcen.
- Aktion: `cloudwatch:DescribeAlarms` auf alle -Ressourcen.
- Aktion: `cloudwatch:GetMetricData` auf alle -Ressourcen.
- Aktion:`cloudwatch:PutMetricAlarm` auf die Ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`
- Aktion:`cloudwatch:DeleteAlarms` auf Ressource`arn:aws:cloudwatch:*:*:alarm:TargetTracking*`

AWS verwaltete Richtlinie: EC2 Spot Fleet und CloudWatch

Name der Richtlinie: AWSApplicationAutoscalingEC2SpotFleetRequestPolicy

Diese Richtlinie ist der serviceverknüpften Rolle mit dem Namen AWSServiceRoleForApplicationAutoScaling_EC2 zugeordnetSpotFleetRequest, damit Application

Auto Scaling Amazon EC2 aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für alle zugehörigen Ressourcen durchzuführen („Ressource“: „*“):

- Aktion: `ec2:DescribeSpotFleetRequests`
- Aktion: `ec2:ModifySpotFleetRequest`
- Aktion: `cloudwatch:DescribeAlarms`
- Aktion: `cloudwatch:PutMetricAlarm`
- Aktion: `cloudwatch:DeleteAlarms`

AWS verwaltete Richtlinie: benutzerdefinierte Ressourcen und CloudWatch

Name der Richtlinie: AWSApplicationAutoScalingCustomResourcePolicy

Diese Richtlinie ist der dienstbezogenen Rolle mit dem Namen AWSServiceRoleForApplicationAutoScaling_CustomResourcezugeordnet, sodass Application Auto Scaling Ihre benutzerdefinierten Ressourcen, die über API Gateway verfügbar sind, aufrufen CloudWatch und die Skalierung in Ihrem Namen durchführen kann.

Berechtigungsdetails

Die Berechtigungsrichtlinie ermöglicht es Application Auto Scaling, die folgenden Aktionen für alle zugehörigen Ressourcen durchzuführen („Ressource“: „*“):

- Aktion: `execute-api:Invoke`
- Aktion: `cloudwatch:DescribeAlarms`
- Aktion: `cloudwatch:PutMetricAlarm`
- Aktion: `cloudwatch:DeleteAlarms`

Updates von Application Auto Scaling für AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Application Auto Scaling an, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen. Um automatisch über

Änderungen auf dieser Seite benachrichtigt zu werden, abonnieren Sie den RSS-Feed auf der Dokumentverlaufsseite Application Auto Scaling.

Änderung	Beschreibung	Datum
Application Auto Scaling fügt seiner SageMaker serviceverknüpften Rolle Berechtigungen hinzu	Diese Richtlinie gewährt dem Dienst nun die Erlaubnis, die SageMaker `DescribeInferenceComponent` und `UpdateInferenceComponentRuntimeConfig` API-Aktionen aufzurufen, um die Kompatibilität für die auto Skalierung von SageMaker Ressourcen für eine bevorstehende Integration zu unterstützen. Die Richtlinie beschränkt nun auch die Aktionen CloudWatch `PutMetricAlarm` und die `DeleteAlarms` API auf CloudWatch Alarme, die zusammen mit Skalierungsrichtlinien für die Zielverfolgung verwendet werden.	13. November 2023
Application Auto Scaling fügt Neptune Policy hinzu	Application Auto Scaling hat eine neue verwaltete Richtlinie für Neptune hinzugefügt. Diese Richtlinie ist einer dienstbezogenen Rolle zugeordnet, die es Application Auto Scaling ermöglicht, Neptune aufzurufen CloudWatch und die Skalierung in Ihrem Namen durchzuführen.	6. Oktober 2021

Änderung	Beschreibung	Datum
Application Auto Scaling fügt ElastiCache die Redis-Richtlinie hinzu	Application Auto Scaling hat eine neue verwaltete Richtlinie für hinzugefügt ElastiCache. Diese Richtlinie ist einer dienstbezogenen Rolle zugeordnet, die es Application Auto Scaling ermöglicht, in Ihrem Namen die Skalierung aufzurufen ElastiCache CloudWatch und durchzuführen.	19. August 2021
Application Auto Scaling hat mit der Verfolgung von Änderungen begonnen	Application Auto Scaling begann, Änderungen für seine AWS verwalteten Richtlinien zu verfolgen.	19. August 2021

Servicegebundene Rollen für Application Auto Scaling

Application Auto Scaling verwendet dienstverknüpfte Rollen für die Berechtigungen, die erforderlich sind, um andere AWS Dienste in Ihrem Namen aufzurufen. Eine dienstverknüpfte Rolle ist ein einzigartiger Rollentyp AWS Identity and Access Management (IAM), der direkt mit einem Dienst verknüpft ist. AWS Mit Diensten verknüpfte Rollen bieten eine sichere Möglichkeit, Berechtigungen an AWS Dienste zu delegieren, da nur der verknüpfte Dienst eine dienstbezogene Rolle übernehmen kann.

Für Dienste, die in Application Auto Scaling integriert sind, erstellt Application Auto Scaling für Sie dienstverknüpfte Rollen. Für jeden Dienst gibt es eine serviceverknüpfte Rolle. Jede serviceverknüpfte Rolle vertraut dem angegebenen Service-Prinzipal, sodass er sie übernehmen kann. Weitere Informationen finden Sie unter ARN-Referenz für serviceverknüpfte Rollen.

Application Auto Scaling umfasst alle erforderlichen Berechtigungen für jede dienstbezogene Rolle. Diese verwalteten Berechtigungen werden von Application Auto Scaling erstellt und verwaltet, und sie definieren die zulässigen Aktionen für jeden Ressourcentyp. Einzelheiten zu den Berechtigungen, die jede Rolle gewährt, finden Sie unter AWS verwaltete Richtlinien für Application Auto Scaling.

Inhalt

- Erforderliche Berechtigungen zum Erstellen einer dienstgebundenen Rolle
- Erstellen von dienstverknüpften Rollen (automatisch)
- Service-verknüpfte Rollen erstellen (manuell)
- Bearbeiten Sie die mit dem Dienst verknüpften Rollen
- Löschen Sie die mit dem Dienst verknüpften Rollen
- Unterstützte Regionen für Application Auto Scaling dienstgebundene Rollen
- ARN-Referenz für serviceverknüpfte Rollen

Erforderliche Berechtigungen zum Erstellen einer dienstgebundenen Rolle

Application Auto Scaling benötigt Berechtigungen, um eine dienstverknüpfte Rolle zu erstellen, wenn ein Benutzer in Ihrem Unternehmen `RegisterScalableTarget` zum ersten Mal einen bestimmten Dienst AWS-Konto aufruft. Application Auto Scaling erstellt eine dienstverknüpfte Rolle für den Zieldienst in Ihrem Konto, wenn die Rolle noch nicht vorhanden ist. Die serviceverknüpfte Rolle gewährt Application Auto Scaling Berechtigungen, damit es den Zieldienst in Ihrem Namen aufrufen kann.

Damit die automatische Rollenerstellung erfolgreich ist, müssen die Benutzer über die Berechtigung für die Aktion `iam:CreateServiceLinkedRole` verfügen.

```
"Action": "iam:CreateServiceLinkedRole"
```

Im Folgenden finden Sie eine identitätsbasierte Richtlinie, die die Berechtigung zum Erstellen einer serviceverknüpften Rolle für die Spot-Flotte gewährt. Sie können die dienstverknüpfte Rolle im Feld `Resource` der Richtlinie als ARN und den Dienstprinzipal für Ihre dienstverknüpfte Rolle als Bedingung angeben, wie gezeigt. Den ARN für jeden Dienst finden Sie unter ARN-Referenz für serviceverknüpfte Rollen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/ec2.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest",
```

```
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName":"ec2.application-autoscaling.amazonaws.com"
                }
            }
        }
    ]
}
```

> **Note**
>
> Der IAM-Bedingungsschlüssel `iam:AWSServiceName` gibt den Service-Principal an, dem die Rolle zugeordnet ist, was in dieser Beispielrichtlinie mit *`ec2.application-autoscaling`*`.amazonaws.com` angegeben ist. Versuchen Sie nicht, den Dienstprinzipal zu erraten. Um den Dienstprinzipal für einen Dienst anzuzeigen, siehe AWS -Services, die Sie mit Application Auto Scaling verwenden können.

Erstellen von dienstverknüpften Rollen (automatisch)

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Application Auto Scaling erstellt die entsprechende dienstgebundene Rolle für Sie, wenn Sie die `RegisterScalableTarget`. Wenn Sie zum Beispiel eine automatische Skalierung für einen Amazon ECS-Service einrichten, erstellt Application Auto Scaling die Rolle `AWSServiceRoleForApplicationAutoScaling_ECSService`.

Service-verknüpfte Rollen erstellen (manuell)

Um die dienstverknüpfte Rolle zu erstellen, können Sie die IAM-Konsole oder die IAM-API AWS CLI verwenden. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

So erstellen Sie eine serviceverknüpfte Rolle (AWS CLI)

Verwenden Sie den folgenden create-service-linked-roleCLI-Befehl, um die serviceverknüpfte Rolle für Application Auto Scaling zu erstellen. Geben Sie in der Anforderung den Dienstnamen "prefix" an.

Den Präfix des Servicenamens finden Sie in den Informationen über den Service-Principal für die serviceverknüpfte Rolle für jeden Service im Abschnitt AWS -Services, die Sie mit Application

Auto Scaling verwenden können. Der Dienstname und der Dienstprinzipal haben das gleiche Präfix. Um beispielsweise die AWS Lambda serviceverknüpfte Rolle zu erstellen, verwenden Sie. `lambda.application-autoscaling.amazonaws.com`

```
aws iam create-service-linked-role --aws-service-name prefix.application-
autoscaling.amazonaws.com
```

Bearbeiten Sie die mit dem Dienst verknüpften Rollen

Bei den dienstverknüpften Rollen, die von Application Auto Scaling erstellt wurden, können Sie nur die Beschreibungen bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen Sie die mit dem Dienst verknüpften Rollen

Wenn Sie Application Auto Scaling nicht mehr mit einem unterstützten Service verwenden, empfehlen wir Ihnen, die entsprechende serviceverknüpfte Rolle zu löschen.

Sie können eine serviceverknüpfte Rolle nur löschen, nachdem Sie zuvor die zugehörigen AWS Ressourcen gelöscht haben. Dies schützt Sie vor dem versehentlichen Entzug von Application Auto Scaling-Berechtigungen für Ihre Ressourcen. Weitere Informationen finden Sie in der Dokumentation zu der skalierbaren Ressource. Um beispielsweise einen Amazon ECS-Service zu löschen, siehe Deleting a service im Amazon Elastic Container Service Developer Guide.

Sie können IAM verwenden, um eine dienstverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Nachdem Sie eine serviceverknüpfte Rolle gelöscht haben, erstellt Application Auto Scaling die Rolle erneut, wenn Sie `RegisterScalableTarget`.

Unterstützte Regionen für Application Auto Scaling dienstgebundene Rollen

Application Auto Scaling unterstützt die Verwendung von dienstbezogenen Rollen in allen AWS Regionen, in denen der Service verfügbar ist.

ARN-Referenz für serviceverknüpfte Rollen

In der folgenden Tabelle ist der Amazon-Ressourcenname (ARN) der serviceverknüpften Rolle für jede Rolle aufgeführt AWS-Service , die mit Application Auto Scaling funktioniert.

Service	ARN
AppStream 2.0	arn:aws:iam:: *012345678910* :role/aws-service-role/appstream.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_AppStreamFleet
Aurora	arn:aws:iam:: *012345678910* :role/aws-service-role/rds.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_RDSCluster
Comprehend	arn:aws:iam:: *012345678910* :role/aws-service-role/comprehend.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ComprehendEndpoint
DynamoDB	arn:aws:iam:: *012345678910* :role/aws-service-role/dynamodb.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_DynamoDBTable
ECS	arn:aws:iam:: *012345678910* :role/aws-service-role/ecs.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService
ElastiCache	arn:aws:iam:: *012345678910* :role/aws-service-role/elasticache.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG
Keyspaces	arn:aws:iam:: *012345678910* :role/aws-service-role/cassandra.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_CassandraTable
Lambda	arn:aws:iam:: *012345678910* :role/aws-service-role/lambda.application-autoscaling.amazonaws.com/AWSS

Service	ARN
	`erviceRoleForApplicationAutoScaling_LambdaCon currency`
MSK	`arn:aws:iam:: 012345678910 :role/aws-service-role/ kafka.application-autoscaling.amazonaws.com/AWSSe rviceRoleForApplicationAutoScaling_KafkaCluster`
Neptune	`arn:aws:iam:: 012345678910 :role/aws-service-role/ neptune.application-autoscaling.amazonaws.com/ AWSServiceRoleForApplicationAutoScaling_NeptuneC luster`
SageMaker	`arn:aws:iam:: 012345678910 :role/aws-service-role/ sagemaker.application-autoscaling.amazonaws.com/ AWSServiceRoleForApplicationAutoScaling_SageMa kerEndpoint`
Spot Flotten	`arn:aws:iam:: 012345678910 :role/aws-service-role/ ec2.application-autoscaling.amazonaws.com/AWSServ iceRoleForApplicationAutoScaling_EC2SpotFleet Request`
Benutzerdefinierte Ressourcen	`arn:aws:iam:: 012345678910 :role/aws-service-role/cust om-resource.application-autoscaling.amazonaws.com/ AWSServiceRoleForApplicationAutoScaling_CustomRes ource`

> **Note**
>
> Sie können den ARN einer dienstverknüpften Rolle für die `RoleARN` Eigenschaft einer AWS::ApplicationAutoScaling::ScalableTargetRessource in Ihren AWS CloudFormation Stack-Vorlagen angeben, auch wenn die angegebene dienstverknüpfte Rolle noch nicht existiert. Application Auto Scaling erstellt die Rolle automatisch für Sie.

Beispiele für identitätsbasierte Richtlinien für Application Auto Scaling

Standardmäßig AWS-Konto hat ein brandneuer Benutzer in Ihrem Bereich keine Rechte, etwas zu tun. Ein IAM-Administrator muss IAM-Richtlinien erstellen und zuweisen, die einer IAM-Identität (etwa einem Benutzer oder einer Rolle) die Berechtigung zum Ausführen von API-Aktionen von Application Auto Scaling erteilen.

Wie Sie eine IAM-Richtlinie anhand der folgenden JSON-Beispielrichtliniendokumente erstellen, erfahren Sie unter Erstellen von Richtlinien auf der Registerkarte JSON im IAM-Benutzerhandbuch.

Inhalt

- Erforderliche Berechtigungen für Application Auto Scaling API-Aktionen
- Erforderliche Berechtigungen für API-Aktionen auf Zieldiensten und CloudWatch
- Berechtigungen für die Arbeit in der AWS Management Console

Erforderliche Berechtigungen für Application Auto Scaling API-Aktionen

Die folgenden Richtlinien gewähren Berechtigungen für häufige Anwendungsfälle beim Aufruf der Application Auto Scaling API. Lesen Sie diesen Abschnitt beim Schreiben von identitätsbasierten Richtlinien. Jede Richtlinie gewährt Berechtigungen für alle oder einige der API-Aktionen von Application Auto Scaling. Sie müssen außerdem sicherstellen, dass Endbenutzer über Berechtigungen für den Zieldienst und verfügen CloudWatch (Einzelheiten finden Sie im nächsten Abschnitt).

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle API-Aktionen von Application Auto Scaling.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "application-autoscaling:*"
            ],
            "Resource": "*"
        }
    ]
}
```

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle API-Aktionen von Application Auto Scaling, die zum Konfigurieren von Skalierungsrichtlinien erforderlich sind, und nicht für geplante Aktionen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "application-autoscaling:RegisterScalableTarget",
              "application-autoscaling:DescribeScalableTargets",
              "application-autoscaling:DeregisterScalableTarget",
              "application-autoscaling:PutScalingPolicy",
              "application-autoscaling:DescribeScalingPolicies",
              "application-autoscaling:DescribeScalingActivities",
              "application-autoscaling:DeleteScalingPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle API-Aktionen von Application Auto Scaling, die zum Konfigurieren von geplanten Aktionen und nicht von Skalierungsrichtlinien erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "application-autoscaling:RegisterScalableTarget",
              "application-autoscaling:DescribeScalableTargets",
              "application-autoscaling:DeregisterScalableTarget",
              "application-autoscaling:PutScheduledAction",
              "application-autoscaling:DescribeScheduledActions",
              "application-autoscaling:DescribeScalingActivities",
              "application-autoscaling:DeleteScheduledAction"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

Erforderliche Berechtigungen für API-Aktionen auf Zieldiensten und CloudWatch

Um Application Auto Scaling erfolgreich mit dem Zielservice zu konfigurieren und zu verwenden, müssen Endbenutzern Berechtigungen für Amazon CloudWatch und für jeden Zielservice, für den sie die Skalierung konfigurieren, erteilt werden. Verwenden Sie die folgenden Richtlinien, um die Mindestberechtigungen zu gewähren, die für die Arbeit mit den Zieldiensten und erforderlich sind CloudWatch.

Inhalt

- AppStream 2.0-Flotten
- Aurora-Replikate
- Amazon Comprehend-Dokumentklassifizierungs- und Entitätserkennungs-Endpunkte
- DynamoDB-Tabellen und globale sekundäre Indizes
- ECS-Services
- ElastiCache Replikationsgruppen
- Amazon EMR-Cluster
- Amazon Keyspace-Tabellen
- Lambda-Funktionen
- Amazon Managed Streaming for Apache Kafka (MSK) Broker-Speicher
- Neptune-Cluster
- SageMaker Endpunkte
- Amazon EC2-Spot-Flotte
- Benutzerdefinierte Ressourcen

AppStream 2.0-Flotten

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle AppStream 2.0- und CloudWatch API-Aktionen, die erforderlich sind.

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

```
        {
            "Effect": "Allow",
            "Action": [
              "appstream:DescribeFleets",
              "appstream:UpdateFleet",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Aurora-Replikate

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle Aurora- und CloudWatch API-Aktionen, die erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "rds:AddTagsToResource",
              "rds:CreateDBInstance",
              "rds:DeleteDBInstance",
              "rds:DescribeDBClusters",
              "rds:DescribeDBInstances",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Comprehend-Dokumentklassifizierungs- und Entitätserkennungs-Endpunkte

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle Amazon Comprehend- und CloudWatch API-Aktionen, die erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "comprehend:UpdateEndpoint",
              "comprehend:DescribeEndpoint",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

DynamoDB-Tabellen und globale sekundäre Indizes

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle erforderlichen DynamoDB- und CloudWatch API-Aktionen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "dynamodb:DescribeTable",
              "dynamodb:UpdateTable",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

ECS-Services

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle erforderlichen ECS- und CloudWatch API-Aktionen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "ecs:DescribeServices",
              "ecs:UpdateService",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

ElastiCache Replikationsgruppen

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle ElastiCache und CloudWatch API-Aktionen, die erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "elasticache:ModifyReplicationGroupShardConfiguration",
              "elasticache:IncreaseReplicaCount",
              "elasticache:DecreaseReplicaCount",
              "elasticache:DescribeReplicationGroups",
              "elasticache:DescribeCacheClusters",
              "elasticache:DescribeCacheParameters",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

Amazon EMR-Cluster

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle erforderlichen Amazon EMR- und CloudWatch API-Aktionen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "elasticmapreduce:ModifyInstanceGroups",
              "elasticmapreduce:ListInstanceGroups",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Keyspace-Tabellen

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle Amazon Keyspaces und CloudWatch API-Aktionen, die erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "cassandra:Select",
              "cassandra:Alter",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
```

```
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Lambda-Funktionen

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle erforderlichen Lambda- und CloudWatch API-Aktionen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "lambda:PutProvisionedConcurrencyConfig",
              "lambda:GetProvisionedConcurrencyConfig",
              "lambda:DeleteProvisionedConcurrencyConfig",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Managed Streaming for Apache Kafka (MSK) Broker-Speicher

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle Amazon MSK- und CloudWatch API-Aktionen, die erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "kafka:DescribeCluster",
              "kafka:DescribeClusterOperation",
```

```
              "kafka:UpdateBrokerStorage",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Neptune-Cluster

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle erforderlichen Neptune- und CloudWatch API-Aktionen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "rds:AddTagsToResource",
              "rds:CreateDBInstance",
              "rds:DescribeDBInstances",
              "rds:DescribeDBClusters",
              "rds:DescribeDBClusterParameters",
              "rds:DeleteDBInstance",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

SageMaker Endpunkte

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle SageMaker erforderlichen CloudWatch API-Aktionen.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "sagemaker:DescribeEndpoint",
              "sagemaker:DescribeEndpointConfig",
              "sagemaker:DescribeInferenceComponent",
              "sagemaker:UpdateEndpointWeightsAndCapacities",
              "sagemaker:UpdateInferenceComponentRuntimeConfig",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon EC2-Spot-Flotte

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen für alle erforderlichen Spot-Flotten- und CloudWatch API-Aktionen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "ec2:DescribeSpotFleetRequests",
              "ec2:ModifySpotFleetRequest",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Benutzerdefinierte Ressourcen

Die folgende identitätsbasierte Richtlinie gewährt die Berechtigung für die API-Ausführungsaktion des API-Gateways. Diese Richtlinie gewährt auch Berechtigungen für alle erforderlichen CloudWatch Aktionen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "execute-api:Invoke",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Berechtigungen für die Arbeit in der AWS Management Console

Es gibt keine eigenständige Konsole für Application Auto Scaling. Die meisten Dienste, die in Application Auto Scaling integriert sind, verfügen über Funktionen, die Sie bei der Konfiguration der Skalierung über ihre Konsole unterstützen.

In den meisten Fällen stellt jeder Dienst AWS verwaltete (vordefinierte) IAM-Richtlinien bereit, die den Zugriff auf seine Konsole definieren, was auch Berechtigungen für die Application Auto Scaling Scaling-API-Aktionen beinhaltet. Weitere Informationen finden Sie in der Dokumentation des Service, dessen Konsole Sie verwenden möchten.

Sie können auch Ihre eigenen benutzerdefinierten IAM-Richtlinien erstellen, um Benutzern fein abgestufte Berechtigungen zum Anzeigen und Arbeiten mit bestimmten Application Auto Scaling API-Aktionen in der AWS Management Console. Sie können die Beispielrichtlinien in den vorherigen Abschnitten verwenden. Sie sind jedoch für Anfragen konzipiert, die mit dem AWS CLI oder einem SDK gestellt werden. Für die Konsole werden zusätzliche API-Aktionen für bestimmte Features verwendet, was bei diesen Richtlinien zu unerwarteten Ergebnissen führen kann. Um beispielsweise Step Scaling zu konfigurieren, benötigen Benutzer möglicherweise zusätzliche Berechtigungen, um CloudWatch Alarme zu erstellen und zu verwalten.

> **Tip**
>
> Verwenden Sie einen Service wie , um einfacher herauszufinden, welche API-Aktionen zum Ausführen von Aufgaben in der Konsole erforderlich sind AWS CloudTrail. Weitere Informationen finden Sie im AWS CloudTrail -Benutzerhandbuch.

Die folgende identitätsbasierte Richtlinie gewährt Berechtigungen zum Konfigurieren von Skalierungsrichtlinien für die Spot-Flotte. Zusätzlich zu den IAM-Berechtigungen für Spot-Flotte muss der Konsolenbenutzer, der über die Amazon-EC2-Konsole auf Flottenskalierungseinstellungen zugreift, über die entsprechenden Berechtigungen für die Services verfügen, die dynamische Skalierung unterstützen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:*",
                "ec2:DescribeSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Get*",
                "sns:List*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
```

```
            "Resource": "arn:aws:iam::*:role/aws-service-role/ec2.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName":"ec2.application-autoscaling.amazonaws.com"
                }
            }
        }
    ]
}
```

Diese Richtlinie ermöglicht es Konsolenbenutzern, Skalierungsrichtlinien in der Amazon EC2 EC2-Konsole anzuzeigen und zu ändern sowie CloudWatch Alarme in der CloudWatch Konsole zu erstellen und zu verwalten.

Sie können die API-Aktionen anpassen, um den Benutzerzugriff zu beschränken. Beispielsweise bedeutet das Ersetzen von `application-autoscaling:*` durch `application-autoscaling:Describe*`, dass der Benutzer schreibgeschützten Zugriff hat.

Sie können die CloudWatch Berechtigungen auch nach Bedarf anpassen, um den Benutzerzugriff auf CloudWatch Funktionen einzuschränken. Weitere Informationen finden Sie unter Für die Nutzung der CloudWatch Konsole erforderliche Berechtigungen im CloudWatch Amazon-Benutzerhandbuch.

Fehlerbehebung beim Zugriff auf Application Auto Scaling

Wenn Sie bei der Arbeit mit Application Auto Scaling auf `AccessDeniedException` oder ähnliche Schwierigkeiten stoßen, lesen Sie bitte die Informationen in diesem Abschnitt.

Ich bin nicht berechtigt, eine Aktion in Application Auto Scaling durchzuführen

Wenn Sie `AccessDeniedException` beim Aufrufen einer AWS API-Operation eine Meldung erhalten, bedeutet dies, dass die von Ihnen verwendeten AWS Identity and Access Management (IAM-) Anmeldeinformationen nicht über die erforderlichen Berechtigungen für diesen Aufruf verfügen.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson`-Benutzer versucht, Details zu einem skalierbaren Ziel anzuzeigen, aber über keine `application-autoscaling:DescribeScalableTargets`-Berechtigung verfügt.

```
An error occurred (AccessDeniedException) when calling the DescribeScalableTargets
 operation: User: arn:aws:iam::123456789012:user/mateojackson is not authorized to
 perform: application-autoscaling:DescribeScalableTargets
```

Wenn Sie diese oder ähnliche Fehler erhalten, müssen Sie Ihren Administrator um Hilfe bitten.

Ein Administrator für Ihr Konto muss sicherstellen, dass Sie über Berechtigungen für den Zugriff auf alle API-Aktionen verfügen, die Application Auto Scaling für den Zugriff auf Ressourcen im Zieldienst und verwendet CloudWatch. Es sind unterschiedliche Berechtigungen erforderlich, je nachdem, mit welchen Ressourcen Sie arbeiten. Application Auto Scaling benötigt auch die Berechtigung, eine serviceverknüpfte Rolle zu erstellen, wenn ein Benutzer zum ersten Mal die Skalierung für eine bestimmte Ressource konfiguriert.

Ich bin ein Administrator und meine IAM-Richtlinie hat einen Fehler zurückgegeben oder funktioniert nicht wie erwartet

Zusätzlich zu den Application Auto Scaling Scaling-Aktionen müssen Ihre IAM-Richtlinien Berechtigungen zum Aufrufen des Zieldienstes und CloudWatch gewähren. Wenn ein Benutzer oder eine Anwendung nicht über diese zusätzlichen Berechtigungen verfügt, wird der Zugriff möglicherweise unerwartet verweigert. Um IAM-Richtlinien für Benutzer und Anwendungen in Ihren Konten zu erstellen, lesen Sie die Informationen unter Beispiele für identitätsbasierte Richtlinien für Application Auto Scaling.

Informationen darüber, wie die Validierung durchgeführt wird, finden Sie unter Validierung von Berechtigungen für API-Aufrufe auf Zielressourcen.

Beachten Sie, dass einige Berechtigungsprobleme auch auf ein Problem bei der Erstellung der dienstverknüpften Rollen zurückzuführen sein können, die von Application Auto Scaling verwendet werden. Informationen zur Erstellung dieser dienstgebundenen Rollen finden Sie unter Servicegebundene Rollen für Application Auto Scaling.

Validierung von Berechtigungen für API-Aufrufe auf Zielressourcen

Um autorisierte Anfragen für API-Aktionen von Application Auto Scaling zu stellen, muss der API-Aufrufer über Berechtigungen für den Zugriff auf AWS Ressourcen im Zieldienst und in CloudWatch verfügen. Application Auto Scaling validiert die Berechtigungen für Anfragen, die sowohl mit dem Zieldienst verknüpft sind, als auch CloudWatch bevor mit der Anfrage fortgefahren wird. Dazu führen wir eine Reihe von Aufrufen durch, um die IAM-Berechtigungen der Zielressourcen zu

überprüfen. Wenn eine Antwort zurückgegeben wird, wird sie von Application Auto Scaling gelesen. Wenn die IAM-Berechtigungen eine bestimmte Aktion nicht zulassen, schlägt Application Auto Scaling die Anfrage fehl und gibt dem Benutzer eine Fehlermeldung mit Informationen über die fehlende Berechtigung zurück. Dadurch wird sichergestellt, dass die Skalierungskonfiguration, die der Benutzer bereitstellen möchte, wie beabsichtigt funktioniert, und dass ein nützlicher Fehler zurückgegeben wird, wenn die Anfrage fehlschlägt.

Als Beispiel dafür, wie dies funktioniert, finden Sie in den folgenden Informationen Informationen darüber, wie Application Auto Scaling Berechtigungsvalidierungen mit Aurora und durchführt. CloudWatch

Wenn ein Benutzer die `RegisterScalableTarget`-API für einen Aurora-DB-Cluster aufruft, führt Application Auto Scaling alle folgenden Prüfungen durch, um zu überprüfen, ob der Benutzer über die erforderlichen Berechtigungen verfügt (fett gedruckt).

- rds:CreateDBInstance: Um festzustellen, ob der Benutzer diese Berechtigung hat, senden wir eine Anfrage an die API-Operation `CreateDBInstance` und versuchen, eine DB-Instance mit ungültigen Parametern (leere Instance-ID) in dem vom Benutzer angegebenen Aurora-DB-Cluster zu erstellen. Für einen autorisierten Benutzer gibt die API nach der Prüfung der Anfrage eine Antwort mit dem Fehlercode `InvalidParameterValue` zurück. Bei einem nicht autorisierten Benutzer erhalten wir jedoch einen `AccessDenied`-Fehler und die Anforderung für Application Auto Scaling schlägt mit dem `ValidationException`-Fehler für den Benutzer fehl, in der die fehlenden Berechtigungen aufgeführt sind.
- rds:DeleteDBInstance: Wir senden eine leere Instance-ID an den `DeleteDBInstance` API-Vorgang. Für einen autorisierten Benutzer führt diese Anforderung zu einem `InvalidParameterValue`-Fehler. Für einen nicht autorisierten Benutzer führt sie zu einem Fehler `AccessDenied` und sendet eine Validierungsausnahme an den Benutzer (gleiche Behandlung wie im ersten Aufzählungspunkt beschrieben).
- rds:AddTagsToResource: Da der `AddTagsToResource` API-Vorgang einen Amazon-Ressourcennamen (ARN) erfordert, ist es notwendig, eine „Dummy“ -Ressource anzugeben, die eine ungültige Konto-ID (12345) und eine Dummy-Instance-ID () verwendet, um den ARN (non-existing-db) zu erstellen. `arn:aws:rds:us-east-1:12345:db:non-existing-db` Für einen autorisierten Benutzer führt diese Anforderung zu einem `InvalidParameterValue`-Fehler. Für einen nicht autorisierten Benutzer führt sie zu `AccessDenied` und sendet eine Validierungsausnahme an den Benutzer.
- rds:DescribeDBCluster: Wir beschreiben den Clusternamen für die Ressource, die für Auto Scaling registriert wird. Für einen autorisierten Benutzer erhalten wir ein gültiges

Beschreibungsergebnis. Für einen nicht autorisierten Benutzer führt sie zu `AccessDenied` und sendet eine Validierungsausnahme an den Benutzer.

- RDS:DescribeBInstance. Wir rufen die API `DescribeDBInstance` mit einem Filter `db-cluster-id` auf, der nach dem Clusternamen filtert, der vom Benutzer zur Registrierung des skalierbaren Ziels angegeben wurde. Einem autorisierten Benutzer ist es erlaubt, alle DB-Instances im DB-Cluster zu beschreiben. Bei einem nicht autorisierten Benutzer ergibt dieser Aufruf `AccessDenied` und sendet eine Validierungsausnahme an den Benutzer.
- cloudwatch:PutMetricAlarm: Wir rufen die `PutMetricAlarm` API ohne Parameter auf. Da der Name des Alarms fehlt, ergibt die Anfrage für einen autorisierten Benutzer den Wert `ValidationError`. Für einen nicht autorisierten Benutzer führt sie zu `AccessDenied` und sendet eine Validierungsausnahme an den Benutzer.
- cloudwatch:DescribeAlarms: Wir rufen die `DescribeAlarms` API auf, wobei der Wert für die maximale Anzahl von Datensätzen auf 1 gesetzt ist. Für einen autorisierten Benutzer erwarten wir Informationen über einen Alarm in der Antwort. Für einen nicht autorisierten Benutzer ergibt dieser Aufruf `AccessDenied` und sendet eine Validierungsausnahme an den Benutzer.
- cloudwatch:DeleteAlarms: Ähnlich wie `PutMetricAlarm` oben stellen wir keine Parameter zur Anfrage bereit`DeleteAlarms`. Da ein Alarmname in der Anfrage fehlt, schlägt dieser Aufruf mit `ValidationError` für einen autorisierten Benutzer fehl. Für einen nicht autorisierten Benutzer führt sie zu `AccessDenied` und sendet eine Validierungsausnahme an den Benutzer.

Jedes Mal, wenn eine dieser Überprüfungsausnahmen auftritt, wird sie protokolliert. Mithilfe AWS CloudTrail von. Sie können Schritte unternehmen, um manuell zu ermitteln, bei welchen Anrufen die Überprüfung fehlgeschlagen ist. Weitere Informationen finden Sie im AWS CloudTrail - Benutzerhandbuch.

> **Note**
>
> Wenn Sie Benachrichtigungen für Application Auto Scaling Scaling-Ereignisse mit erhalten CloudTrail, enthalten diese Benachrichtigungen standardmäßig die Application Auto Scaling Scaling-Aufrufe zur Überprüfung von Benutzerberechtigungen. Um diese Warnungen herauszufiltern, verwenden Sie das `invokedBy`-Feld, das für diese Validierungsprüfungen `application-autoscaling.amazonaws.com` enthält.

Application Auto Scaling und VPC-Endpunkte als Schnittstelle

Sie können die Sicherheit Ihrer VPC erhöhen, indem Sie Application Auto Scaling so konfigurieren, dass ein Schnittstellen-VPC-Endpunkt verwendet wird. Schnittstellenendpunkte basieren auf einer Technologie AWS PrivateLink, mit der Sie privat auf Application Auto Scaling-APIs zugreifen können, indem der gesamte Netzwerkverkehr zwischen Ihrer VPC und Application Auto Scaling auf das Netzwerk beschränkt wird. AWS Mit Schnittstellenendpunkten benötigen Sie außerdem kein Internet-Gateway, kein NAT-Gerät und kein Virtual Private Gateway.

Eine Konfiguration ist nicht erforderlich AWS PrivateLink, wird aber empfohlen. Weitere Informationen zu AWS PrivateLink VPC-Endpunkten finden Sie unter Was ist? AWS PrivateLink im Leitfaden.AWS PrivateLink

Themen

- Erstellen eines Schnittstellen-VPC-Endpunkts
- Erstellen Sie eine VPC-Endpunktrichtlinie

Erstellen eines Schnittstellen-VPC-Endpunkts

Erstellen Sie einen Endpunkt für Application Auto Scaling mit dem folgenden Dienstnamen:

```
com.amazonaws.region.application-autoscaling
```

Weitere Informationen finden Sie im AWS PrivateLink Handbuch unter Zugreifen auf einen AWS Dienst über einen Schnittstellen-VPC-Endpunkt.

Sie brauchen keine anderen Einstellungen zu ändern. Application Auto Scaling ruft andere AWS Dienste entweder über Dienstendpunkte oder VPC-Endpunkte mit privater Schnittstelle auf, je nachdem, welche verwendet werden.

Erstellen Sie eine VPC-Endpunktrichtlinie

Sie können eine Richtlinie an Ihren VPC-Endpunkt anhängen, um den Zugriff auf die Application Auto Scaling-API zu steuern. Die Richtlinie legt Folgendes fest:

- Prinzipal, der die Aktionen ausführen kann.
- Die Aktionen, die ausgeführt werden können.
- Die Ressource, auf der die Aktionen ausgeführt werden können.

Das folgende Beispiel zeigt eine VPC-Endpunktrichtlinie, die jedem Benutzer die Berechtigung zum Löschen einer Skalierungsrichtlinie über den Endpunkt verweigert. Die Beispielrichtlinie gewährt auch jedem die Berechtigung, alle anderen Aktionen auszuführen.

```
{
   "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "application-autoscaling:DeleteScalingPolicy",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

Weitere Informationen finden Sie unter VPC-Endpunkt-Richtlinien im AWS PrivateLink -Leitfaden.

Ausfallsicherheit bei Application Auto Scaling

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones.

AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind.

Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Sicherheit der Infrastruktur bei Application Auto Scaling

Als verwalteter Service ist Application Auto Scaling 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 Application Auto Scaling 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.

Compliance-Validierung für Application Auto Scaling

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen zu Sicherheit und Compliance — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- Architecting for HIPAA Security and Compliance on Amazon Web Services — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-fähige Anwendungen erstellen AWS können.

> **Note**
>
> AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der Referenz für HIPAA-berechtigte Services.

- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dies AWS-Service bietet einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerungsreferenz.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Kontingente für Application Auto Scaling

Ihr AWS-Konto verfügt über Standardkontingente – früher als Limits bezeichnet – für jeden AWS-Service. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region. Sie können Erhöhungen für einige Kontingente beantragen und andere Kontingente können nicht erhöht werden.

Um die Kontingente für Application Auto Scaling anzuzeigen, öffnen Sie die Service-Quotas-Konsole. Wählen Sie im Navigationsbereich AWS-Services aus und wählen Sie Application Auto Scaling aus.

Informationen zum Beantragen einer Kontingenterhöhung finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch. Wenn das Kontingent unter Service Quotas noch nicht in verfügbar ist, verwenden Sie das Formular zu Limits von Application Auto Scaling. Vergewissern Sie sich, dass Sie bei Ihrer Anfrage nach einer Erhöhung den Ressourcentyp angeben, z. B. Amazon ECS oder DynamoDB.

Ihr AWS-Konto hat die folgenden Kontingente in Bezug auf Application Auto Scaling.

Standardkontingente pro Region und Konto

Item	Standard	Anpassbar
Maximale Anzahl von skalierbaren Zielen pro Ressourcentyp	Standard-Kontingente variieren je nach Ressourcentyp. Bis zu 5 000 skalierbare Amazon-DynamoDB-Ziele, 3 000 skalierbare ECS-Ziele, 1 500 skalierbare Amazon-Keyspaces-Ziele und jeweils 500 skalierbare Ziele für alle anderen Ressourcentypen.	Ja
Maximale Anzahl von Skalierungsrichtlinien pro skalierbarem Ziel	50	Nein

Item	Standard	Anpassbar
	Dazu zählen sowohl Richtlinien zur schrittweisen Skalierung als auch Richtlinien für die Zielnachverfolgung.	
Maximale Anzahl von geplanten Aktionen pro skalierbarem Ziel	200	Nein
Maximale Anzahl von Schrittanpassungen pro Richtlinie für die schrittweise Skalierung.	20	Nein

Denken Sie an die Servicekontingente, wenn Sie Ihre Workloads skalieren. Wenn Sie beispielsweise die maximal zulässige Anzahl von Kapazitätseinheiten eines Services erreichen, wird die Skalierung beendet. Wenn die Nachfrage sinkt und die aktuelle Kapazität abnimmt, kann Application Auto Scaling wieder skalieren. Um die Kapazität nicht erneut auszuschöpfen, können Sie eine Erhöhung anfordern. Jeder Service verfügt über eigene Standardkontingente für die maximale Kapazität der Ressource. Informationen zu den Standardkontingenten für andere AWS-Services finden Sie unter Service-Endpunkte und -Kontingente in Allgemeine Amazon Web Services-Referenz.

Dokumentverlauf

Die folgende Tabelle beschreibt wichtige Ergänzungen der Application Auto Scaling-Dokumentation, die im Januar 2018 beginnen. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie den RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Änderungen im Handbuch	Der Eintrag Maximale Anzahl skalierbarer Ziele pro Ressourcentyp in der Kontingentdokumentation wurde aktualisiert. Siehe Kontingente für Application Auto Scaling.	16. Januar 2024
Unterstützung für SageMaker Inferenzkomponenten	Verwenden Sie Application Auto Scaling, um Kopien einer Inferenzkomponente zu skalieren.	29. November 2023
Aktualisieren auf Berechtigungen für serviceverknüpfte IAM-Rollen	Application Auto Scaling aktualisiert die Richtlinie `AWSApplicationAutoscalingSageMakerEndpointPolicy` . Weitere Informationen finden Sie unter Updates für von AWS verwaltete Richtlinien mit Application Auto Scaling.	13. November 2023
Unterstützung für bereitgestellte Serverless SageMaker - Gleichzeitigkeit	Verwenden Sie Application Auto Scaling, um die bereitgestellte Gleichzeitigkeit eines Serverless-Endpunkts zu skalieren.	9. Mai 2023

Kategorisieren Ihrer skalierbaren Ziele mithilfe von Tags	Sie können Ihren skalierbaren Zielen für Application Auto Scaling jetzt Metadaten in Form von Tags zuweisen. Siehe Tagging-Unterstützung für Application Auto Scaling.	20. März 2023
Unterstützung für CloudWatch Metrikberechnungen	Sie können jetzt Metrikberechnungen verwenden, wenn Sie Skalierungsrichtlinien für die Zielverfolgung erstellen. Mit Metrikberechnungen können Sie mehrere CloudWatch Metriken abfragen und mathematische Ausdrücke verwenden, um neue Zeitreihen basierend auf diesen Metriken zu erstellen. Siehe Erstellen einer Zielverfolgungs-Skalierungsrichtlinie für die automatische Skalierung von Anwendungen mit Hilfe von Metrikberechnungen.	14. März 2023
Änderungen im Handbuch	Ein neues Thema im Application Auto Scaling Scaling-Benutzerhandbuch hilft Ihnen bei den ersten Schritten bei der Verwendung von AWS CloudShell mit Application Auto Scaling. Siehe Verwenden von AWS CloudShell, um von der Befehlszeile aus mit Application Auto Scaling zu arbeiten.	17. Februar 2023

Gründe für die Nichtskalierung	Sie können jetzt mithilfe der Application-Auto-Scaling-API die maschinenlesbaren Gründe für die Nicht-Skalierung Ihrer Ressourcen durch Application Auto Scaling abrufen. Weitere Informationen finden Sie unter Skalierungsaktivitäten für Application Auto Scaling.	4. Januar 2023
Änderungen im Handbuch	Der Eintrag Maximale Anzahl skalierbarer Ziele pro Ressourcentyp in der Kontingentdokumentation wurde aktualisiert. Siehe Kontingente für Application Auto Scaling.	6. Mai 2022
Unterstützung für Amazon Neptune-Cluster hinzufügen	Verwenden Sie Application Auto Scaling, um die Anzahl der Replikate in einem Amazon Neptune DB-Cluster zu skalieren. Weitere Informationen finden Sie unter Amazon Neptune und Application Auto Scaling. Das Thema Application Auto Scaling Updates auf AWS Verwaltete -Richtlinien wurde aktualisiert, um eine neue verwaltete Richtlinie für die Integration mit Neptune aufzulisten.	6. Oktober 2021

Application Auto Scaling meldet jetzt Änderungen an seinen AWS verwalteten Richtlinien	Ab dem 19. August 2021 werden Änderungen an verwalteten Richtlinien im Thema Updates für Application Auto Scaling auf AWS verwaltete Richtlinien gemeldet. Die erste aufgeführte Änderung ist das Hinzufügen von Berechtigungen, die für ElastiCache für Redis erforderlich sind.	19. August 2021
Unterstützung für ElastiCache für Redis-Replikationsgruppen hinzufügen	Verwenden Sie Application Auto Scaling, um die Anzahl der Knotengruppen und die Anzahl der Replikate pro Knotengruppe für eine ElastiCache for Redis-Replikationsgruppe (Cluster) zu skalieren. Weitere Informationen finden Sie unter ElastiCache für Redis und Application Auto Scaling.	19. August 2021

Änderungen im Handbuch	Neue IAM-Themen im Application Auto Scaling-B enutzerhandbuch helfen Ihnen bei der Fehlerbehebung beim Zugriff auf das automatische Skalieren von Anwendungen. Weitere Informationen finden Sie unter Identity and Access Management für Applicati on Auto Scaling. Außerdem wurden neue IAM-Beisp ielberechtigungsrichtlinien für Aktionen auf Zielservices und Amazon hinzugefügt CloudWatch. Weitere Informati onen finden Sie unter Beispielr ichtlinien für die Arbeit mit dem AWS CLI oder einem SDK.	23. Februar 2021
Unterstützung für lokale Zeitzonen hinzufügen	Sie können jetzt geplante Aktionen in der lokalen Zeitzone erstellen. Wenn Ihre Zeitzone die Sommerzei t einhält, wird sie automatis ch an die Sommerzeit (DST) angepasst. Weitere Informationen finden Sie unter Geplante Skalierung.	2. Februar 2021

Änderungen im Handbuch	Ein neues Tutorial im Application Auto Scaling-Benutzerhandbuch zeigt Ihnen, wie Sie mit Hilfe von Skalierungsrichtlinien zur Zielverfolgung und geplanter Skalierung die Verfügbarkeit Ihrer Anwendung erhöhen können, wenn Sie Application Auto Scaling verwenden. Außerdem wird in einem neuen Thema erklärt, wie Sie eine Benachrichtigung auslösen, wenn Probleme entdeckt CloudWatch hat, die Ihre Aufmerksamkeit erfordern könnten.	15. Oktober 2020
Support für Amazon Managed Streaming for Apache Kafka Cluster-Speicher hinzufügen	Verwenden Sie eine Zielverfolgungs-Skalierungsrichtlinie, um die Menge des mit einem Amazon MSK-Cluster verbundenen Brokerspeichers zu skalieren.	30. September 2020
Unterstützung für Amazon Comprehend Entity Recognizer Endpunkte hinzufügen	Verwenden Sie Application Auto Scaling, um die Anzahl der Inferenzeinheiten zu skalieren, die für Ihre Amazon Comprehend Entity Recognizer Endpunkte bereitgestellt werden.	28. September 2020

Hinzufügen von Unterstützung für Amazon Keyspace-Tabellen (für Apache Cassandra)	Verwenden Sie Application Auto Scaling, um den bereitgestellten Durchsatz (Lese- und Schreibkapazität) einer Amazon Keyspaces-Tabelle zu skalieren.	23. April 2020
Neues Sicherheitskapitel	Ein neues Kapitel zum Thema Sicherheit im Application Auto Scaling Benutzerhandbuch hilft Ihnen, die Anwendung des Modells der geteilten Verantwortung bei der Verwendung von Application Auto Scaling zu verstehen. Im Rahmen dieser Aktualisierung wurde das Kapitel "Authentifizierung und Zugriffskontrolle" im Benutzerhandbuch durch einen neuen, nützlicheren Abschnitt Identity and Access Management für Application Auto Scaling ersetzt.	16. Januar 2020
Kleinere Updates	Verschiedene Verbesserungen und Korrekturen.	15. Januar 2020
Hinzufügen einer Benachrichtigungsfunktion	Application Auto Scaling sendet jetzt Ereignisse an Amazon EventBridge und Benachrichtigungen an Ihre , AWS Health Dashboard wenn bestimmte Aktionen stattfinden. Weitere Informationen finden Sie unter Überwachung von Application Auto Scaling.	20. Dezember 2019

Hinzufügen von Support für AWS Lambda-Funktionen	Verwenden Sie Application Auto Scaling, um die bereitgestellte Gleichzeitigkeit einer Lambda-Funktion zu skalieren.	3. Dezember 2019
Hinzufügen von Unterstützung für Amazon Comprehend Dokumentenklassifizierungsendpunkte	Verwenden Sie Application Auto Scaling, um die Durchsatzkapazität eines Amazon Comprehend-Endpunkts für die Dokumentenklassifizierung zu skalieren.	25. November 2019
Unterstützung von AppStream 2.0 für Skalierungsrichtlinien für die Ziel-Nachverfolgung hinzufügen	Verwenden Sie Skalierungsrichtlinien für die Ziel-Nachverfolgung, um die Größe einer AppStream 2.0-Flotte zu skalieren.	25. November 2019
Unterstützung für Amazon VPC-Endpunkte	Sie können nun eine private Verbindung zwischen Ihrer VPC und Application Auto Scaling herstellen. Überlegungen und Anleitungen zur Migration finden Sie unter Application Auto Scaling und Schnittstelle zu VPC-Endpunkten.	22. November 2019
Anhalten und Fortsetzen von Skalierungen	Unterstützung für das Anhalten und Fortsetzen der Skalierung wurde hinzugefügt. Weitere Informationen finden Sie unter Unterbrechen und Wiederaufnehmen der Skalierung für Application Auto Scaling.	29. August 2019

Neuer Abschnitt	Application Auto Scaling wurde um den Abschnitt Einrichten erweitert. Am gesamten Benutzerhandbuch wurden kleinere Verbesserungen vorgenommen.	28. Juni 2019
Änderungen im Handbuch	Die Application Auto Scaling-Dokumentation wurde in den Abschnitten Geplante Skalierung, Stufenskalierungsrichtlinien und Zielverfolgungs-Skalierungsrichtlinien verbessert.	11. März 2019
Hinzufügen von Support für benutzerdefinierte Ressourcen	Verwenden Sie Application Auto Scaling, um benutzerdefinierte Ressourcen zu skalieren, die von Ihren eigenen Anwendungen oder Diensten bereitgestellt werden. Weitere Informationen finden Sie in unserem GitHub Repository .	9. Juli 2018
Unterstützung für SageMaker Endpunktvarianten hinzufügen	Verwenden Sie Application Auto Scaling, um die Anzahl der Endpunktinstanzen zu skalieren, die für eine Variante bereitgestellt werden.	28. Februar 2018

Die folgende Tabelle beschreibt wichtige Änderungen an der Application Auto Scaling-Dokumentation vor Januar 2018.

Änderung	Beschreibung	Datum
Unterstützung für Aurora Replicas hinzugefügt	Verwenden Sie Application Auto Scaling, um die gewünschte Anzahl zu skalieren. Weitere Informationen finden Sie unter Verwendung von Amazon Aurora Auto Scaling mit Aurora-Replikaten im Amazon RDS User Guide.	17. November 2017
Unterstützung für geplante Skalierung hinzugefügt	Verwenden Sie die geplante Skalierung, um Ressourcen zu bestimmten voreingestellten Zeiten oder Intervallen zu skalieren. Weitere Informationen finden Sie unter Geplante Skalierung für Application Auto Scaling.	8. November 2017
Unterstützung für Skalierungsrichtlinien für die Ziel-Nachverfolgung hinzugefügt	Verwenden Sie Skalierungsrichtlinien für die Ziel-Nachverfolgung, um eine dynamische Skalierung für Ihre Anwendung in nur wenigen einfachen Schritten einzurichten. Weitere Informationen finden Sie unter Zielverfolgungs-Skalierungsrichtlinien für Application Auto Scaling.	12. Juli 2017
Hinzufügen von Unterstützung für bereitgestellte Lese- und Schreibkapazität für DynamoDB-Tabellen und globale sekundäre Indizes	Verwenden Sie Application Auto Scaling zur Skalierung des bereitgestellten Durchsatzes (Lese- und Schreibkapazität). Weitere Informationen	14. Juni 2017

Änderung	Beschreibung	Datum
	finden Sie unter Verwalten der Durchsatzkapazität mit DynamoDB Auto Scaling im Amazon DynamoDB Developer Leitfaden.	
Unterstützung für AppStream 2.0-Flotten hinzufügen	Verwenden Sie Application Auto Scaling, um die Größe der Flotte zu skalieren. Weitere Informationen finden Sie unter Fleet Auto Scaling für AppStream 2.0 im Amazon- AppStream 2.0-Administratorhandbuch.	23. März 2017
Unterstützung für Amazon EMR-Cluster hinzufügen	Verwenden Sie Application Auto Scaling zur Skalierung der Kern- und Aufgabenknoten. Weitere Informationen finden Sie unter Verwenden der automatischen Skalierung im Amazon EMR im Amazon EMR Management Guide.	18. November 2016
Unterstützung für Spot-Flotten hinzugefügt	Verwenden Sie Application Auto Scaling, um die Zielkapazität zu skalieren. Weitere Informationen finden Sie unter Automatic scaling for Spot fleet im Amazon EC2 User Guide for Linux Instances.	1. September 2016

Änderung	Beschreibung	Datum
Unterstützung für Amazon ECS-Services hinzufügen	Verwenden Sie Application Auto Scaling, um die gewünschte Anzahl zu skalieren. Weitere Informationen finden Sie unter Service Auto Scaling im Amazon Elastic Container Service Developer Guide.	9. August 2016

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.