aws

Leitfaden

Amazon EC2 Auto Scaling

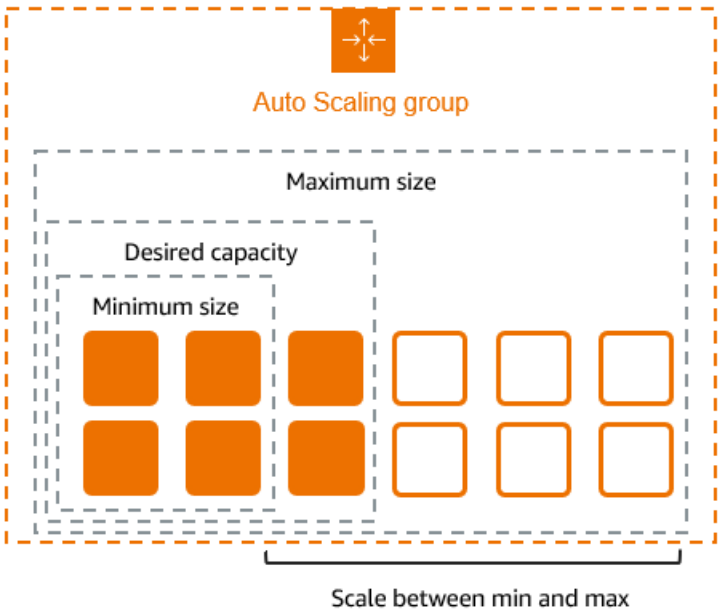
Amazon EC2 Auto Scaling: Leitfaden

Table of Contents

Was ist Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling hilft Ihnen sicherzustellen, dass Sie die richtige Anzahl von Amazon EC2-Instances zur Verfügung haben, um die Auslastung Ihrer Anwendung zu bewältigen. Sie erstellen Sammlungen von EC2 Instances, die als Auto Scaling-Gruppen bezeichnet werden. Sie können die minimale Anzahl von Instances in jeder Auto Scaling-Gruppe angeben und Amazon EC2 Auto Scaling stellt sicher, dass Ihre Gruppe nie kleiner als diese Grösse wird. Sie können die maximale Anzahl von Instances in jeder Auto-Scaling-Gruppe angeben und Amazon EC2 Auto Scaling stellt sicher, dass Ihre Gruppe nie grösser als diese Grösse wird. Wenn Sie die gewünschte Kapazität angeben – entweder beim Erstellen der Gruppe oder später – stellt Amazon EC2 Auto Scaling sicher, dass Ihrer Gruppe die entsprechende Anzahl an Instances zugewiesen wird. Wenn Sie Skalierungsrichtlinien angeben, kann Amazon EC2 Auto Scaling je nach Bedarf Ihrer Anwendung Instances starten oder beenden.

Die folgende Auto Scaling Scaling-Gruppe hat beispielsweise eine Mindestgröße von vier Instances, eine gewünschte Kapazität von sechs Instances und eine Maximalgröße von zwölf Instances. Die Anzahl an Instances wird gemäß den von Ihnen festgelegten Skalierungsrichtlinien angepasst. Sie liegt immer zwischen Ihrer Mindest- und Höchstanzahl an Instances und richtet sich nach den von Ihnen angegebenen Kriterien.



Funktionen von Amazon EC2 Auto Scaling

Mit Amazon EC2 Auto Scaling sind Ihre EC2-Instances in Auto Scaling-Gruppen organisiert, sodass sie für Skalierungs- und Verwaltungszwecke als logische Einheit behandelt werden können. Auto Scaling Scaling-Gruppen verwenden Startvorlagen (oder Startkonfigurationen) als Konfigurationsvorlagen für ihre EC2-Instances.

Im Folgenden sind die wichtigsten Funktionen von Amazon EC2 Auto Scaling aufgeführt:

Überwachung des Zustands laufender Instances

Amazon EC2 Auto Scaling überwacht automatisch den Zustand und die Verfügbarkeit Ihrer Instances mithilfe von EC2-Zustandsprüfungen und ersetzt beendete oder beeinträchtigte Instances, um die gewünschte Kapazität aufrechtzuerhalten.

Benutzerdefinierte Zustandsprüfungen

Zusätzlich zu den integrierten Integritätsprüfungen können Sie benutzerdefinierte Zustandsprüfungen definieren, die speziell auf Ihre Anwendung zugeschnitten sind, um sicherzustellen, dass sie erwartungsgemäß reagiert. Wenn eine Instance Ihre benutzerdefinierte Zustandsprüfung nicht besteht, wird sie automatisch ersetzt, um die gewünschte Kapazität aufrechtzuerhalten.

Kapazitätsausgleich zwischen Availability Zones

Sie können mehrere Availability Zones für Ihre Auto Scaling-Gruppe angeben, und Amazon EC2 Auto Scaling verteilt Ihre Instances gleichmäßig auf die Availability Zones, wenn die Gruppe skaliert. Dies sorgt für hohe Verfügbarkeit und Stabilität, indem Ihre Anwendungen an einem einzigen Standort vor Ausfällen geschützt werden.

Mehrere Instance-Typen und Kaufoptionen

Innerhalb einer einzigen Auto Scaling Scaling-Gruppe können Sie mehrere Instance-Typen und Kaufoptionen (Spot- und On-Demand-Instances) starten, sodass Sie die Kosten durch die Nutzung von Spot-Instances optimieren können. Sie können auch Rabatte für Reserved Instances und Savings Plan nutzen, indem Sie sie in Verbindung mit On-Demand-Instances in der Gruppe verwenden.

Automatischer Ersatz von Spot Instances

Wenn Ihre Gruppe Spot-Instances umfasst, kann Amazon EC2 Auto Scaling automatisch Ersatz-Spot-Kapazität anfordern, falls Ihre Spot-Instances unterbrochen werden. Durch Capacity

Rebalancing kann Amazon EC2 Auto Scaling auch Ihre Spot-Instances, bei denen ein erhöhtes Ausfallrisiko besteht, überwachen und proaktiv ersetzen.

Load Balancing

Mit Elastic Load Balancing Load Balancing und Health Checks können Sie sicherstellen, dass der Anwendungsdatenverkehr gleichmäßig auf Ihre intakten Instances verteilt wird. Immer wenn Instances gestartet oder beendet werden, registriert Amazon EC2 Auto Scaling die Instances automatisch und meldet sie vom Load Balancer ab.

Skalierbarkeit

Amazon EC2 Auto Scaling bietet Ihnen auch mehrere Möglichkeiten, Ihre Auto Scaling-Gruppen zu skalieren. Mithilfe von Auto Scaling können Sie die Anwendungsverfügbarkeit aufrechterhalten und die Kosten senken, indem Sie Kapazität hinzufügen, um Lastspitzen zu bewältigen, und Kapazität entfernen, wenn der Bedarf geringer ist. Sie können die Größe Ihrer Auto Scaling Scaling-Gruppe nach Bedarf auch manuell anpassen.

Instance-Aktualisierung

Die Instance-Aktualisierungsfunktion bietet einen Mechanismus, um Instances fortlaufend zu aktualisieren, wenn Sie Ihr AMI oder Ihre Startvorlage aktualisieren. Sie können auch einen schrittweisen Ansatz verwenden, der als Canary-Deployment bezeichnet wird, um ein neues AMI oder eine neue Startvorlage auf einer kleinen Anzahl von Instances zu testen, bevor Sie es für die gesamte Gruppe bereitstellen.

Lebenszyklus-Hooks

Lifecycle-Hooks sind nützlich, um benutzerdefinierte Aktionen zu definieren, die beim Start neuer Instances oder vor dem Beenden von Instances aufgerufen werden. Diese Funktion ist besonders nützlich für den Aufbau ereignisgesteuerter Architekturen, hilft Ihnen aber auch dabei, Instanzen während ihres gesamten Lebenszyklus zu verwalten.

Support für statusbehaftete Workloads

Lifecycle-Hooks bieten auch einen Mechanismus, um den Status beim Herunterfahren beizubehalten. Um die Kontinuität von statusbehafteten Anwendungen zu gewährleisten, können Sie auch skalierbaren Schutz oder benutzerdefinierte Kündigungsrichtlinien verwenden, um zu verhindern, dass Instanzen mit lang andauernden Prozessen vorzeitig beendet werden.

Weitere Informationen über die Vorteile von Amazon EC2 Auto Scaling finden Sie unter Vorteile von Auto Scaling für die Anwendungsarchitektur.

Preise für Amazon EC2 Auto Scaling

Bei Amazon EC2 Auto Scaling fallen keine zusätzlichen Gebühren an. Sie können es also ganz einfach ausprobieren und herausfinden, wie es Ihrer AWS Architektur zugute kommen kann. Sie zahlen nur für die AWS Ressourcen (z. B. EC2-Instances, EBS-Volumes und CloudWatch Alarme), die Sie tatsächlich nutzen.

Erste Schritte

Schließen Sie zunächst das Tutorial Erstellen Sie Ihre erste Auto Scaling Scaling-Gruppe ab, um eine Auto Scaling Scaling-Gruppe zu erstellen und zu sehen, wie sie reagiert, wenn eine Instance in dieser Gruppe beendet wird.

Arbeiten mit Auto-Scaling-Gruppen

Sie können die folgenden Schnittstellen verwenden, um Ihre Auto-Scaling-Gruppen zu erstellen, auf sie zuzugreifen und sie zu verwalten:

- AWS Management Console – Bietet eine Webschnittstelle für den Zugriff auf Ihre Auto-Scaling-Gruppen. Wenn Sie sich für eine angemeldet haben AWS-Konto, können Sie auf Ihre Auto Scaling Scaling-Gruppen zugreifen, indem Sie sich bei der anmelden AWS Management Console, mit dem Suchfeld in der Navigationsleiste nach Auto Scaling Scaling-Gruppen suchen und dann Auto Scaling Scaling-Gruppen auswählen.
- 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 Bereiten Sie sich auf die Verwendung des vor AWS CLI an. Weitere Informationen finden Sie unter 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.

- Abfrage-API – Bietet API-Aktionen auf niedriger Ebene, die Sie mithilfe von HTTPS-Anforderungen aufrufen. Die Verwendung der Abfrage-API ist die direkteste Möglichkeit für den Zugriff auf AWS-Services. Allerdings müssen dann viele technische Abläufe, wie beispielsweise das Erzeugen des Hashwerts zum Signieren der Anforderung und zur Fehlerbehandlung, in der Anwendung durchgeführt werden. Weitere Informationen finden Sie in der Amazon EC2 Auto Scaling-API-Referenz.
- AWS CloudFormation— Unterstützt das Erstellen von Auto Scaling Scaling-Gruppen mithilfe von CloudFormation Vorlagen. Weitere Informationen finden Sie unter Erstellen von Auto-Scaling-Gruppen mit AWS CloudFormation.

Um programmgesteuert eine Verbindung zu einem herzustellen AWS-Service, verwenden Sie einen Endpunkt.

Vorteile von Auto Scaling für die Anwendungsarchitektur

Das Hinzufügen von Amazon EC2 Auto Scaling zu Ihrer Anwendungsarchitektur ist eine Möglichkeit, die Vorteile der AWS Cloud zu maximieren. Wenn Sie Amazon EC2 Auto Scaling verwenden, profitieren Sie bei Ihren Anwendungen von folgenden Vorteilen:

- Bessere Fehlertoleranz. Amazon EC2 Auto Scaling erkennt, wenn eine Instance fehlerhaft ist, beendet sie und startet eine andere Instance, um die fehlerhafte zu ersetzen. Sie können Amazon EC2 Auto Scaling auch so konfigurieren, dass mehrere Availability Zones genutzt werden. Wenn eine Availability Zone ausfällt, startet Amazon EC2 Auto Scaling die Instances in einer anderen, um den Ausfall zu kompensieren.
- Bessere Verfügbarkeit. Mit Amazon EC2 Auto Scaling können Sie sicherstellen, dass für Ihre Anwendung immer genügend Kapazität zur Verarbeitung des aktuellen Datenverkehrs verfügbar ist.
- Bessere Kostenkontrolle. Amazon EC2 Auto Scaling passt die Kapazität je nach Bedarf dynamisch an. Da Sie für die Nutzung von EC2-Instances zahlen, sparen Sie Kosten, wenn Sie sie nur bei Bedarf starten und wieder beenden, sobald Sie sie nicht mehr brauchen.

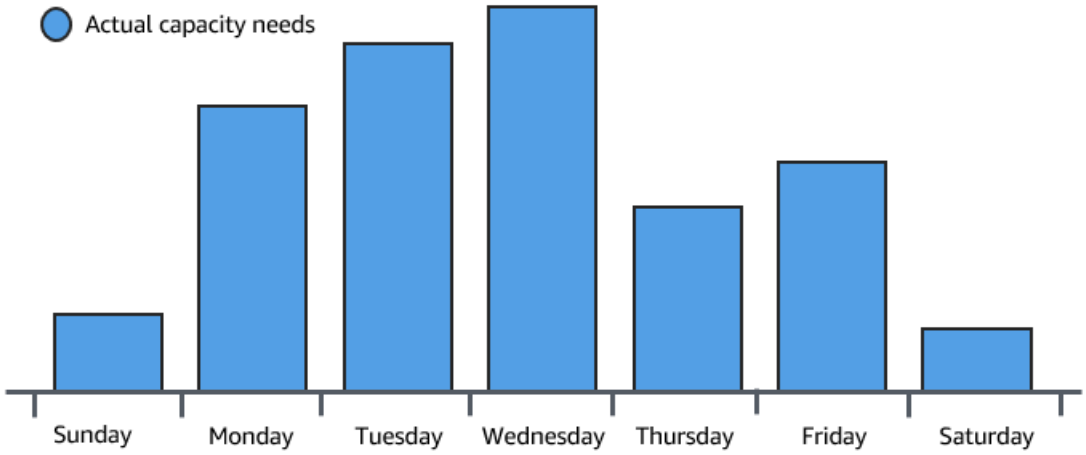
Inhalt

- Instance-Distribution
- Wiederherstellen des Gleichgewichts von Aktivitäten

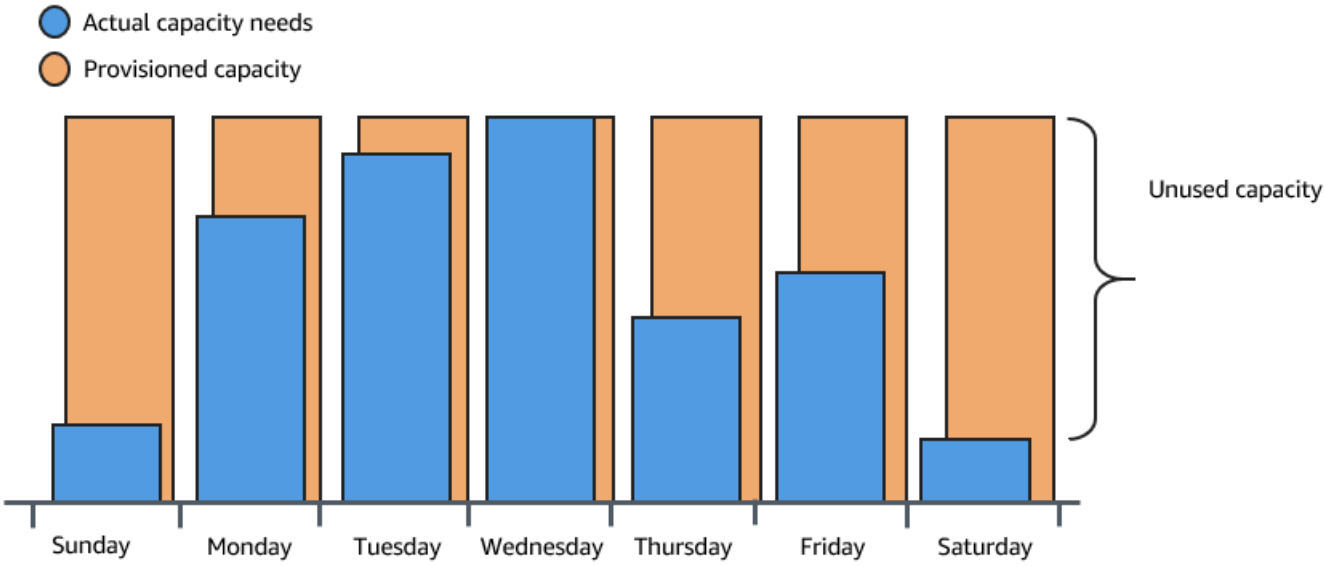
Beispiel: Abdecken des Variablenbedarfs

Wir möchten Ihnen einige der Vorteile von Amazon EC2 Auto Scaling anhand einer einfachen Webanwendung zeigen, die in AWS ausgeführt wird. Mit dieser Anwendung können Mitarbeiter Konferenzräume für Meetings suchen. Am Anfang und am Ende der Woche wird diese Anwendung nur wenig genutzt. Mitte der Woche planen mehr Mitarbeiter ihre Meetings, die Anforderungen an die Anwendung erhöhen sich also deutlich.

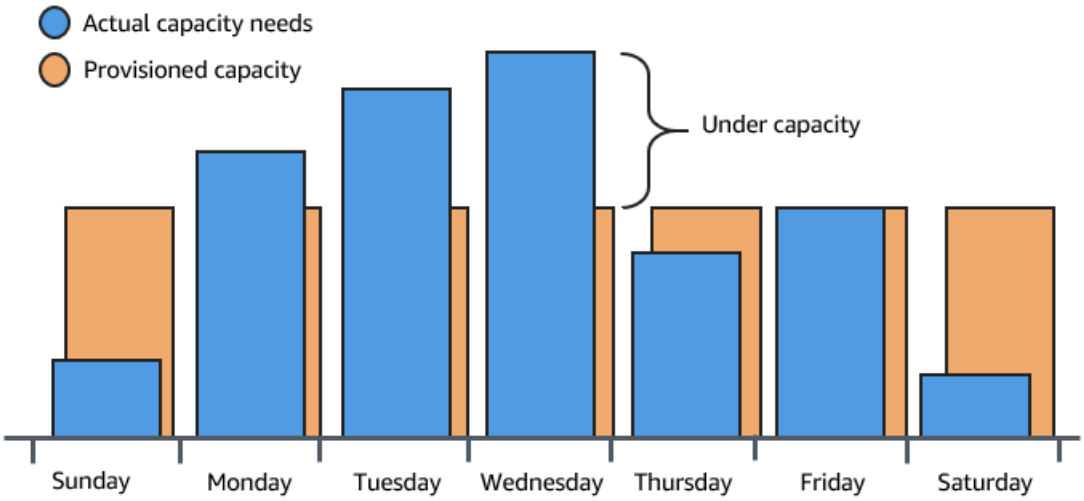
Das folgende Diagramm zeigt, wie viel Kapazität der Anwendung im Lauf der Woche verwendet wird.



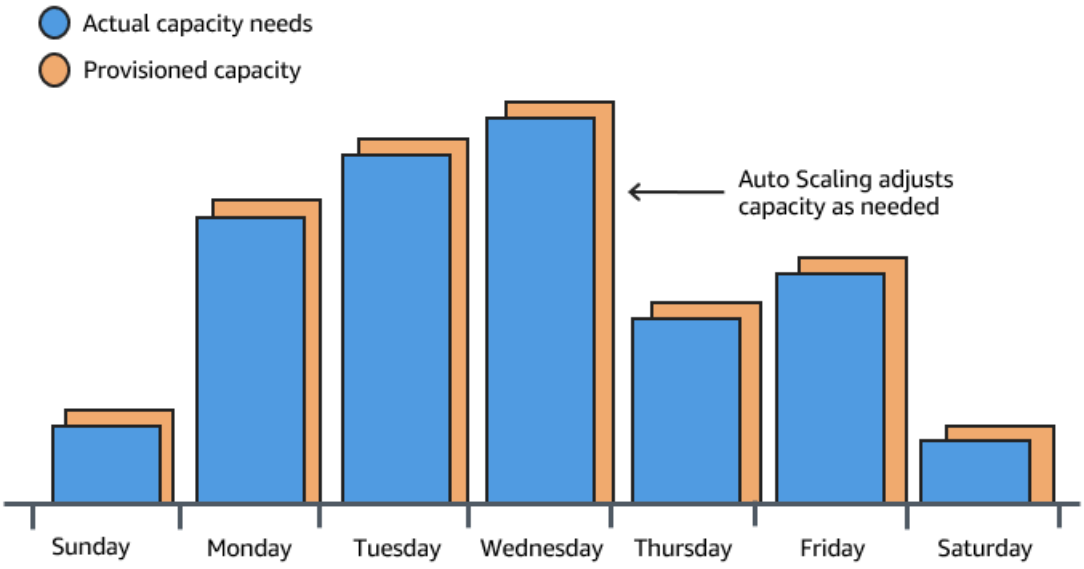
Bisher gab es zwei nur Möglichkeiten, diese Änderungen in der Kapazitätsplanung zu berücksichtigen. Bei der ersten Möglichkeit werden so viele Server hinzugefügt, dass der Kapazitätsbedarf der Anwendung immer gedeckt ist. Diese Möglichkeit hat jedoch einen Nachteil. An machen Tagen benötigt die Anwendung nicht so viel Kapazität. Die zusätzliche Kapazität bleibt ungenutzt und erhöht im Grunde genommen die Kosten für das Ausführen der Anwendung.



Bei der zweiten Möglichkeit wird darauf geachtet, dass die Kapazität für den durchschnittlichen Bedarf der Anwendung ausreicht. Diese Möglichkeit ist kostengünstiger, da Sie keine Geräte kaufen, die Sie nur gelegentlich verwenden. Allerdings riskieren Sie eine schlechte Kundenerfahrung, wenn der Bedarf der Anwendung die Kapazität übersteigt.



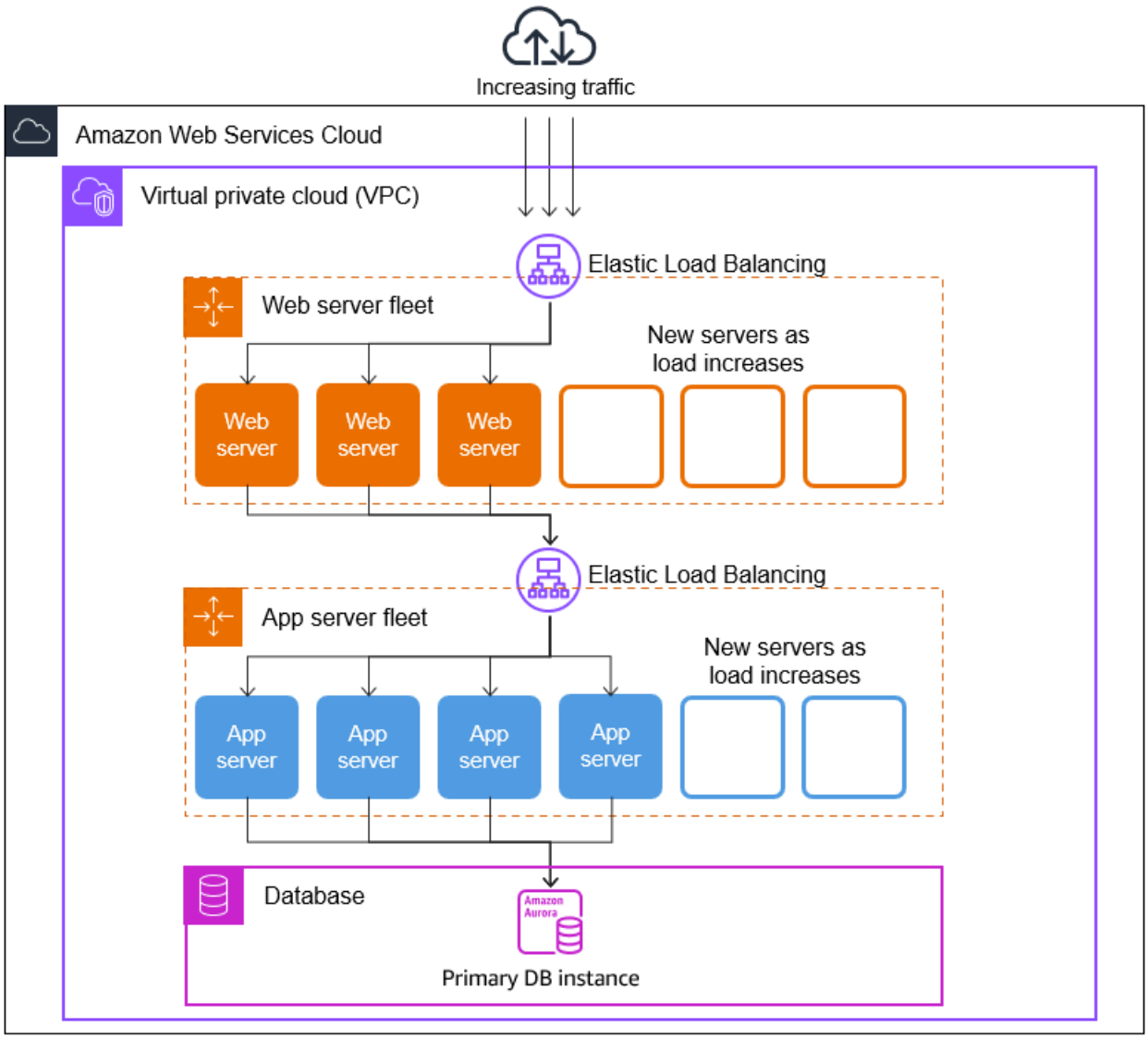
Wenn Sie dieser Anwendung Amazon EC2 Auto Scaling hinzufügen, bietet sich eine dritte Möglichkeit. Sie können der Anwendung bei Bedarf neue Instances hinzufügen und diese wieder beenden, wenn Sie sie nicht mehr benötigen. Da Amazon EC2 Auto Scaling EC2-Instances verwendet, bezahlen Sie nur für die Instances, die Sie tatsächlich nutzen. Sie verfügen jetzt über eine kosteneffektive Architektur, die eine optimale Kundenerfahrung erzielt und gleichzeitig die Kosten gering hält.



Beispiel: Architektur für eine Web-App

Bei den meisten Web-Apps werden mehrere Kopien der App gleichzeitig ausgeführt, um dem Kunden-Datenverkehr gerecht zu werden. Die einzelnen Kopien Ihrer Anwendung werden auf identischen EC2-Instances (Cloud-Servern) gehostet, die die Kundenanfragen jeweils bearbeiten.

Amazon EC2 Auto Scaling verwaltet das Starten und Beenden dieser EC2-Instances für Sie. Sie definieren eine Reihe von Kriterien (z. B. einen CloudWatch Amazon-Alarm), die bestimmen, wann die Auto Scaling Scaling-Gruppe EC2-Instances startet oder beendet. Das Hinzufügen von Auto-Scaling-Gruppen zur Ihrer Netzwerkarchitektur trägt dazu bei, dass sich die Verfügbarkeit und Fehlertoleranz Ihrer Anwendung verbessern.



Sie können so viele Auto-Scaling-Gruppen erstellen, wie Sie benötigen. Sie können beispielsweise für jede Ebene eine Auto-Scaling-Gruppe erstellen.

Damit der Datenverkehr zwischen den Instances in Ihren Auto-Scaling-Gruppen aufgeteilt wird, können Sie einen Load Balancer in Ihre Architektur aufnehmen. Weitere Informationen finden Sie unter Elastic Load Balancing.

Beispiel: Aufteilen von Instances in mehrere Availability Zones

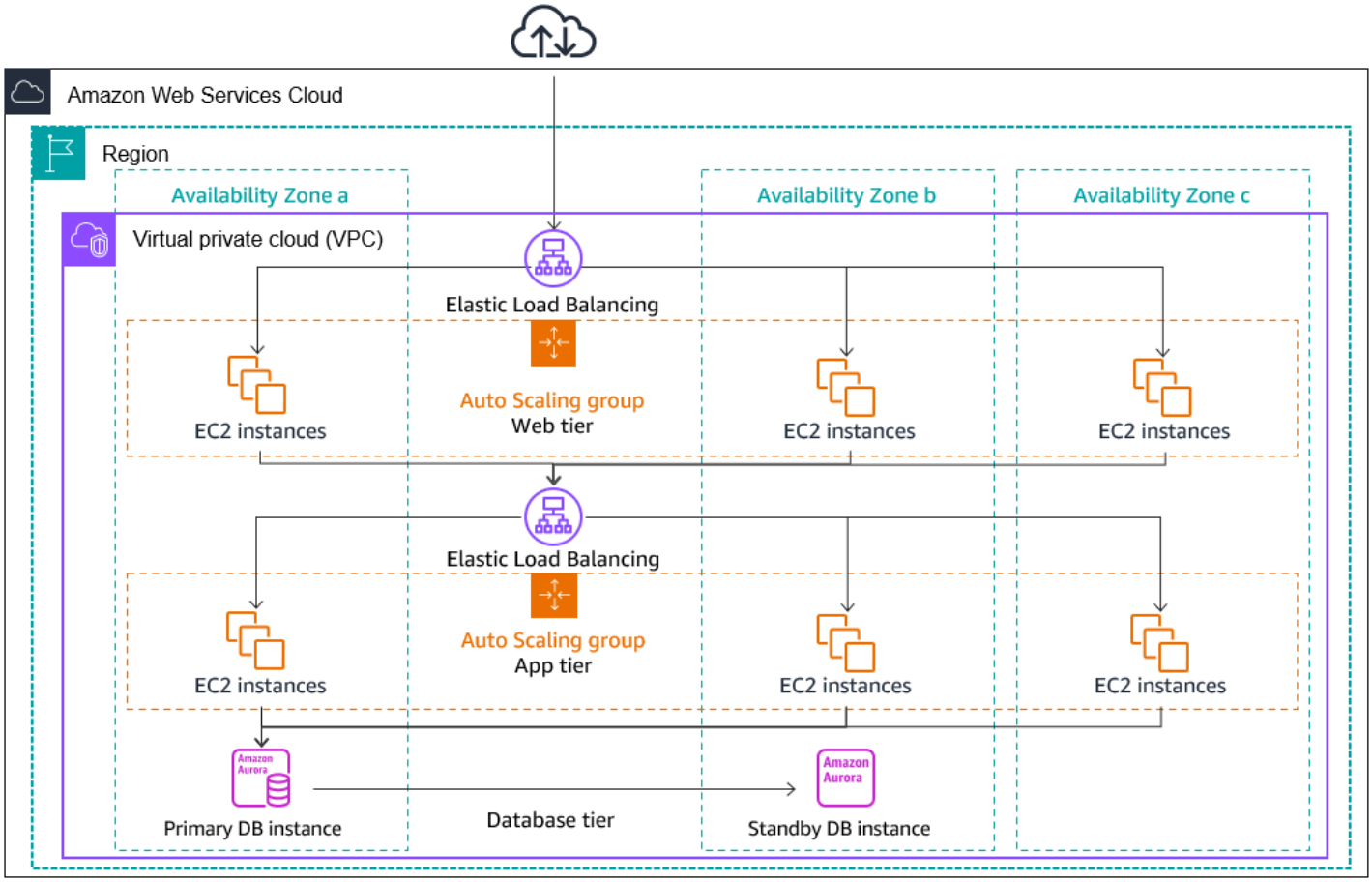
Availability Zones sind isolierte Standorte innerhalb in einer gegebenen AWS-Region-Region. Jede Region verfügt über mehrere Availability Zones, die eine hohe Verfügbarkeit für die Region bieten. Availability Zones sind unabhängig, und daher erhöhen Sie die Anwendungsverfügbarkeit, wenn Sie

Ihre Anwendung so entwerfen, dass sie mehrere Zonen verwendet. Weitere Informationen finden Sie unter Ausfallsicherheit in Amazon EC2 Auto Scaling.

Eine Availability Zone wird durch den AWS-Region Code gefolgt von einer Buchstabenkennung identifiziert (z. B. `us-east-1a`). Wenn Sie die VPC und Subnetze erstellen, anstatt die Standard-VPC zu verwenden, können Sie eines oder mehrere Subnetze in jeder Availability Zone definieren. Jedes Subnetz muss sich vollständig innerhalb einer Availability Zone befinden und darf nicht mehrere Zonen umfassen. Weitere Informationen finden Sie unter Wie funktioniert Amazon VPC im Benutzerhandbuch für Amazon VPC.

Wenn Sie eine Auto-Scaling-Gruppe erstellen, müssen Sie die VPC und die Subnetze auswählen, in denen Sie die Auto-Scaling-Gruppe bereitstellen möchten. Amazon EC2 Auto Scaling erstellt Ihre Instances in den von Ihnen ausgewählten Subnetzen. Jede Instance ist somit einer bestimmten Availability Zone zugeordnet, die von Amazon EC2 Auto Scaling ausgewählt wurde. Wenn Instances gestartet werden, versucht Amazon EC2 Auto Scaling, sie gleichmäßig auf die Zonen zu verteilen, um eine hohe Verfügbarkeit und Zuverlässigkeit zu gewährleisten.

Sie sehen hier einen Überblick über die mehrstufige Architektur, die in drei Availability Zones eingesetzt wird.



Instance-Distribution

Amazon EC2 Auto Scaling versucht automatisch, die gleiche Anzahl von Instances in jeder aktivierten Availability Zone aufrechtzuerhalten. Amazon EC2 Auto Scaling versucht dazu, in der Availability Zone mit den wenigsten Instances neue Instances zu starten. Wenn mehrere Subnetze in einer Availability Zone verwendet werden, wählt Amazon EC2 Auto Scaling das Subnetz der Availability Zone nach dem Zufallsprinzip aus. Falls dies fehlschlägt, versucht Amazon EC2 Auto Scaling so lange, die Instances in einer anderen Availability Zone zu starten, bis dies gelingt.

Unter Umständen, in denen eine Availability Zone nicht mehr funktioniert oder nicht mehr verfügbar ist, kann die Verteilung der Instances auf die Availability Zones ungleichmäßig verteilt werden. Wenn die Availability Zone wiederhergestellt ist, gleicht Amazon EC2 Auto Scaling die Auto-Scaling-Gruppe automatisch neu aus. Dies geschieht, indem Instances in den aktivierten Availability Zones mit den wenigsten Instances gestartet und Instances an anderer Stelle beendet werden.

Wiederherstellen des Gleichgewichts von Aktivitäten

Neuausgleichsaktivitäten fallen in zwei Kategorien: Neuausgleich der Availability Zone und Neuausgleich der Kapazität.

Neuausgleich der Availability Zone

Nach bestimmten Aktionen kann das Verhältnis der Availability Zones Ihrer Auto-Scaling-Gruppe aus dem Gleichgewicht geraten. Amazon EC2 Auto Scaling kann dies kompensieren und das Gleichgewicht der Availability Zones wiederherstellen. Folgende Aktionen können ein Wiederherstellen des Gleichgewichts erforderlich machen:

- Sie ändern die Availability Zones, die Ihrer Auto-Scaling-Gruppe zugeordnet sind.
- Sie beenden oder trennen Instances explizit, oder versetzen Instances in den Standby-Modus, wodurch die Gruppe aus dem Gleichgewicht gerät.
- Eine Availability Zone, die bisher zu wenig Kapazität hatte, wurde wiederhergestellt, wodurch jetzt zusätzliche Kapazität zur Verfügung steht.
- Eine Availability Zone mit einem Spot-Preis, der bisher über Ihrem Höchstpreis lag, liegt jetzt darunter.

Beim Wiederherstellen des Gleichgewichts von Instances startet Amazon EC2 Auto Scaling neue Instances, bevor die alten beendet werden. Auf diese Weise beeinträchtigt ein Wiederherstellen des Gleichgewichts die Leistung und Verfügbarkeit Ihrer Anwendung nicht.

Da Amazon EC2 Auto Scaling vor dem Beenden der vorherigen Instances versucht, neue zu starten, kann das Wiederherstellen des Gleichgewichts beeinträchtigt und sogar gänzlich unterbrochen werden, falls die angegebene maximale Kapazität nahezu oder gänzlich erreicht ist.

Um dieses Problem zu vermeiden, kann das System beim Wiederherstellen des Gleichgewichts die angegebene maximale Kapazität einer Gruppe vorübergehend überschreiten. Standardmäßig kann dies mit einer Marge von 10 Prozent oder einer Instance geschehen, je nachdem, welcher Wert größer ist. Die Marge wird nur verlängert, wenn die Gruppe die maximale Kapazität erreicht oder fast erreicht und eine Neugewichtung erforderlich ist. Die Kapazität wird nur für die Dauer der Wiederherstellung des Gleichgewichts in der Gruppe erhöht, in der Regel sind dies einige Minuten.

Alternativ können Sie mithilfe einer Wartungsrichtlinie für Instances Schwellenwerte für eine Auto-Scaling-Gruppe festlegen, und die Gruppe kann die Kapazität nur innerhalb dieses Schwellenwertbereichs erhöhen oder verringern. Auf diese Weise können Sie kontrollieren, wie

schnell Ihre Gruppe eine erneute Verteilung durchführt. Weitere Informationen finden Sie unter Wartungsrichtlinien für Instances.

Kapazitätsausgleich

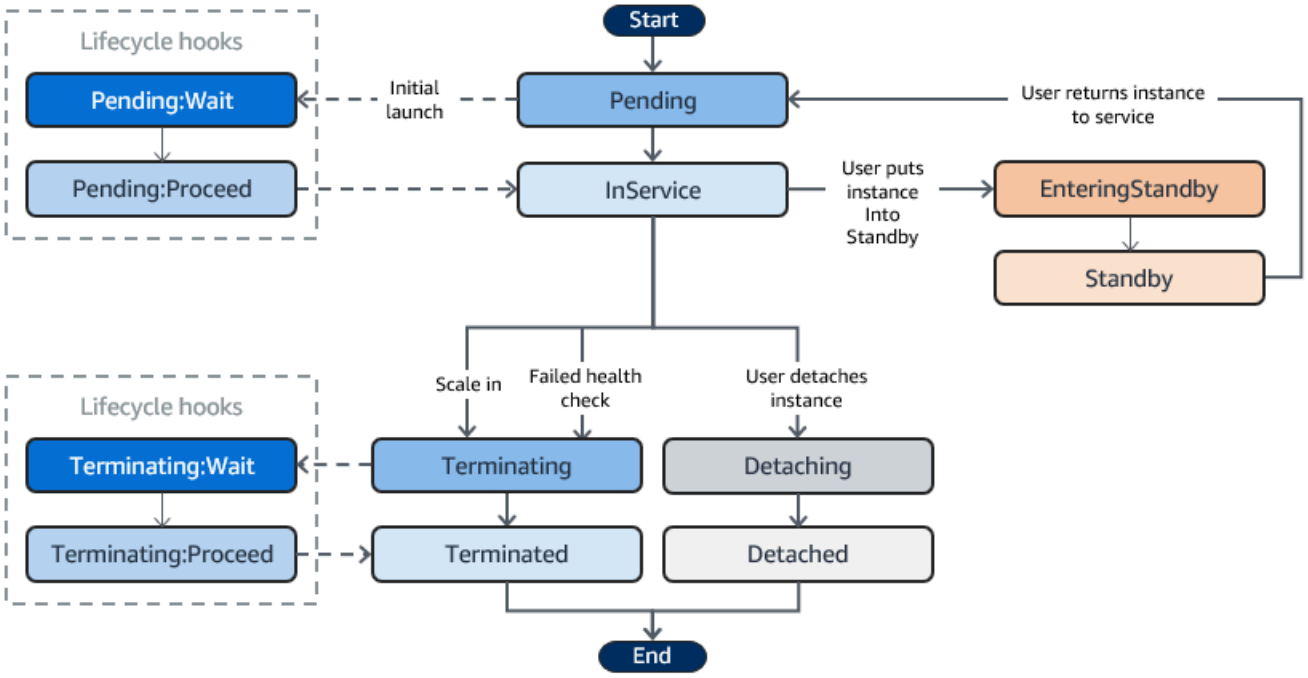
Wenn Sie Spot-Instances verwenden, können Sie den Kapazitätsausgleich für Ihre Auto-Scaling-Gruppen aktivieren. Dadurch kann Amazon EC2 Auto Scaling versuchen, eine Spot-Instance zu starten, wenn Amazon EC2 benachrichtigt, dass eine Spot-Instance einem erhöhten Unterbrechungsrisiko ausgesetzt ist. Nach dem Start einer neuen Instance wird dann eine frühere Instance beendet. Weitere Informationen finden Sie unter Verwenden des Kapazitätsausgleichs, um Amazon-EC2-Spot-Unterbrechungen zu behandeln.

Instance-Lebenszyklus bei Amazon EC2 Auto Scaling

Der Lebenszyklus der EC2-Instances in einer Auto-Scaling-Gruppe unterscheidet sich von dem anderer EC2-Instances. Der Lebenszyklus beginnt, wenn die Auto-Scaling-Gruppe eine Instance startet und sie in Betrieb nimmt. Der Lebenszyklus endet, wenn Sie die Instance beenden oder die Auto-Scaling-Gruppe die Instance ausmustert und beendet.

> **Note**
>
> Instances werden Ihnen in Rechnung gestellt, sobald sie gestartet werden, einschließlich der Zeit, die sie noch nicht in Betrieb sind.

Die folgende Abbildung zeigt die Übergänge zwischen den Instance-Status im Amazon EC2 Auto Scaling-Lebenszyklus.



Horizontale Skalierung

Die folgenden horizontalen Skalierungsereignisse weisen die Auto-Scaling-Gruppe an, EC2-Instances zu starten und sie der Gruppe anzufügen:

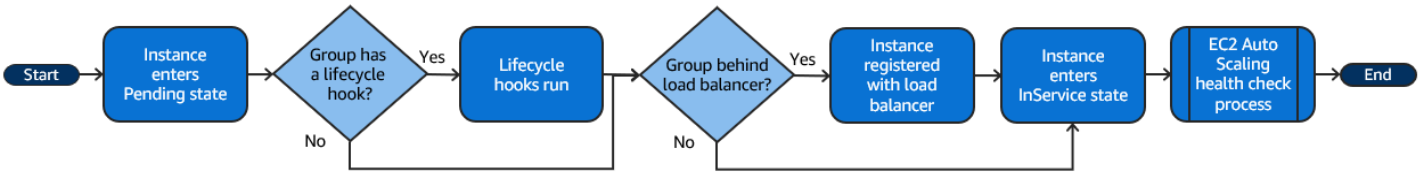
- Sie vergrößern die Gruppe manuell. Weitere Informationen finden Sie unter Ändern der gewünschten Kapazität einer vorhandenen Auto-Scaling-Gruppe.
- Sie erstellen eine Skalierungsrichtlinie, mit der die Gruppe bei einem festgelegten Anstieg des Bedarfs automatisch vergrößert wird. Weitere Informationen finden Sie unter Dynamische Skalierung für Amazon EC2 Auto Scaling.
- Sie richten die Skalierung nach Zeitplan ein, um die Gruppe zu einem bestimmten Zeitpunkt zu vergrößern. Weitere Informationen finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling.

Wenn ein Aufskalierungsereignis eintritt, startet die Auto-Scaling-Gruppe die erforderliche Anzahl von EC2-Instances unter Verwendung der ihr zugewiesenen Startvorlage. Zunächst lautet der Status der Instances `Pending`. Wenn Sie Ihrer Auto-Scaling-Gruppe einen Lebenszyklus-Hook hinzufügen, kann hier eine benutzerdefinierte Aktion ausgeführt werden. Weitere Informationen finden Sie unter Lebenszyklus-Hooks.

Wenn jede Instance vollständig konfiguriert ist und die Amazon EC2-Zustandsprüfungen besteht, wird sie der Auto-Scaling-Gruppe hinzugefügt und erhält den Zustand `InService`. Jede Instance wird auf die gewünschte Kapazität der Auto-Scaling-Gruppe angerechnet.

Wenn Ihre Auto-Scaling-Gruppe so konfiguriert ist, dass sie Datenverkehr von einem Load Balancer von Elastic Load Balancing empfängt, registriert Amazon EC2 Auto Scaling Ihre Instance automatisch beim Load Balancer, bevor es die Instance als `InService` markiert.

Im Folgenden werden die Schritte zur Registrierung einer Instance bei einem Load Balancer für ein Scale-Out-Event zusammengefasst.



In Betrieb genommene Instances

Instances behalten den Status `InService`, bis eines der folgenden Ereignisse eintritt:

- Zur horizontalen Skalierung nach unten beendet Amazon EC2 Auto Scaling diese Instance, um die Auto-Scaling-Gruppe zu verkleinern. Weitere Informationen finden Sie unter Steuern welche Auto-Scaling-Instances beim Abskalieren beendet werden.
- Sie versetzen die Instance in den Status `Standby`. Weitere Informationen finden Sie unter Aktivieren und Deaktivieren des Standby-Status.
- Sie trennen die Instance von der Auto-Scaling-Gruppe. Weitere Informationen finden Sie unter Instanzen trennen oder anhängen.
- Die Instance besteht eine bestimmte Anzahl an Zustandsprüfungen nicht und wird deshalb aus der Auto-Scaling-Gruppe entfernt, beendet und ersetzt. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Scale-In

Die folgenden Abwärtsskalierungsereignisse weisen die Auto-Scaling-Gruppe an, EC2-Instances von der Gruppe zu trennen und zu beenden:

- Sie verkleinern die Gruppe manuell. Weitere Informationen finden Sie unter Ändern der gewünschten Kapazität einer vorhandenen Auto-Scaling-Gruppe.

- Sie erstellen eine Skalierungsrichtlinie, mit der die Gruppe bei einem festgelegten Rückgang des Bedarfs automatisch verkleinert wird. Weitere Informationen finden Sie unter Dynamische Skalierung für Amazon EC2 Auto Scaling.
- Sie richten die Skalierung nach Zeitplan ein, um die Gruppe zu einem bestimmten Zeitpunkt zu verkleinern. Weitere Informationen finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling.

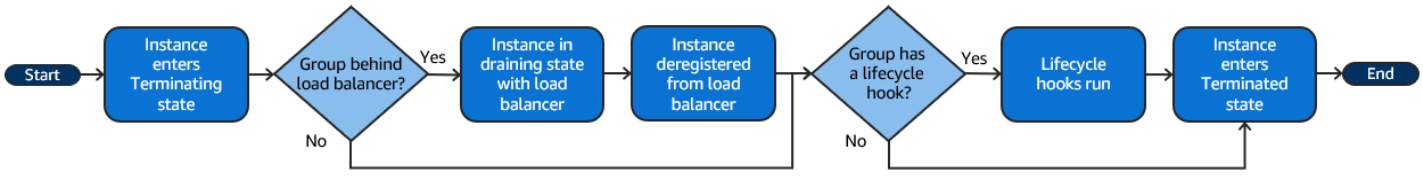
Es ist wichtig, dass Sie für jedes horizontale Skalierungsereignis ein entsprechendes Abwärtsskalierungsereignis erstellen. So stellen Sie sicher, dass die Ressourcen, die Ihrer Anwendung zugewiesen sind, dem Bedarf so gut wie möglich entsprechen.

Wenn ein Abwärtsskalierungsereignis auftritt, trennt die Auto-Scaling-Gruppe eine oder mehrere Instances ab. Die Auto-Scaling-Gruppe ermittelt anhand ihrer Beendigungsrichtlinie, welche Instance beendet werden soll. Instances, die gerade von der Auto-Scaling-Gruppe beendet werden, erhalten den Status `Terminating` und können nicht wieder in Betrieb genommen werden.

Wenn Ihre Auto-Scaling-Gruppe so konfiguriert ist, dass sie Datenverkehr von einem Load Balancer von Elastic Load Balancing empfängt, hebt Amazon EC2 Auto Scaling die Registrierung der beendenden Instance automatisch vom Load Balancer auf. Durch die Abmeldung der Instance wird sichergestellt, dass alle neuen Anforderungen an andere Instance in der Zielgruppe des Load Balancers umgeleitet werden, während bestehende Verbindungen mit der Instance fortgesetzt werden können, bis die Abmeldeverzögerung abläuft.

Wenn Sie Ihrer Auto-Scaling-Gruppe einen Lebenszyklus-Hook hinzufügen, kann eine benutzerdefinierte Aktion auf der Instance ausgeführt werden, die beendet wird. Weitere Informationen finden Sie unter Lebenszyklus-Hooks. Die Instance wird schließlich vollständig beendet und erhält den Status `Terminated`.

Im Folgenden werden die Schritte zum Abmelden einer Instance bei einem Load Balancer für ein Scale-In-Ereignis zusammengefasst.



Trennen einer Instance

Sie können eine Instance von Ihrer Auto-Scaling-Gruppe trennen. Nachdem die Instance getrennt wurde, können Sie sie getrennt von der Auto-Scaling-Gruppe verwalten oder sie an eine andere Auto-Scaling-Gruppe anfügen.

Weitere Informationen finden Sie unter Instanzen trennen oder anhängen.

Hinzufügen einer Instance

Sie können Ihrer Auto-Scaling-Gruppe eine laufende EC2-Instance hinzufügen, wenn sie bestimmte Kriterien erfüllt. Nach dem Hinzufügen der Instance wird sie als Teil der Auto-Scaling-Gruppe verwaltet.

Weitere Informationen finden Sie unter Instanzen trennen oder anhängen.

Lebenszyklus-Hooks

Sie können Ihrer Auto-Scaling-Gruppe einen Lebenszyklus-Hook hinzufügen, damit beim Starten oder Beenden von Instances benutzerdefinierte Aktionen ausgeführt werden.

Wenn Amazon-EC2-Auto-Scaling auf ein horizontales Skalierungsereignis reagiert, werden eine oder mehrere Instances gestartet. Zunächst lautet der Status der Instances `Pending`. Falls Sie Ihrer Auto-ScalingGruppe einen Lebenszyklus-Hook vom Typ `autoscaling:EC2_INSTANCE_LAUNCHING` hinzugefügt haben, ändert sich der Status der Instances von `Pending` zu `Pending:Wait`. Wenn Sie die Lebenszyklusaktion abgeschlossen haben, erhalten die Instances den Status `Pending:Proceed`. Wenn die Instances vollständig konfiguriert sind, werden sie der Auto-Scaling-Gruppe angefügt und erhalten den Status `InService`.

Wenn Amazon EC2 Auto Scaling auf ein Abwärtsskalierungsereignis reagiert, werden eine oder mehrere Instances beendet. Diese Instances werden von der Auto-Scaling-Gruppe getrennt und erhalten den Status `Terminating`. Falls Sie Ihrer Auto-ScalingGruppe einen Lebenszyklus-Hook vom Typ `autoscaling:EC2_INSTANCE_TERMINATING` hinzugefügt haben, ändert sich der Status der Instances von `Terminating` zu `Terminating:Wait`. Wenn Sie die Lebenszyklusaktion abgeschlossen haben, erhalten die Instances den Status `Terminating:Proceed`. Wenn die Instances vollständig beendet werden, erhalten sie den Status `Terminated`.

Weitere Informationen finden Sie unter Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling.

Aktivieren und Deaktivieren des Standby-Status

Sie können eine Instance mit dem Status `InService` in den Status `Standby` versetzen. So können Sie die Instance aus dem Betrieb nehmen, Probleme beheben oder sie ändern und sie dann wieder in Betrieb nehmen.

Instances mit dem Status `Standby` werden weiterhin von der Auto-Scaling-Gruppe verwaltet. Sie werden jedoch erst wieder ein aktiver Teil Ihrer Anwendung, wenn Sie sie wieder in Betrieb nehmen.

Weitere Informationen finden Sie unter Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe.

Kontingente für Auto Scaling Scaling-Ressourcen und Gruppen

Ihr AWS-Konto hat Standardkontingente, früher als Limits bezeichnet, für jeden AWS Dienst. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region. Sie können Erhöhungen für einige Kontingente beantragen und andere Kontingente können nicht erhöht werden.

Um die Kontingente für Amazon EC2 Auto Scaling anzuzeigen, öffnen Sie die Service-Quotas-Konsole. Wählen Sie im Navigationsbereich und dann AWS Services aus und wählen Sie Amazon EC2 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 Limits für Auto Scaling. Die Erhöhung eines Kontingents ist immer an die Region gebunden, für die sie angefragt wurde.

Alle Anfragen werden an gesendet AWS Support. Sie können Ihren Anforderungsfall in der AWS Support -Konsole verfolgen.

Amazon EC2 Auto Scaling-Ressourcen

Für Sie AWS-Konto gelten die folgenden Kontingente in Bezug auf die Anzahl der Auto Scaling Scaling-Gruppen und Startkonfigurationen, die Sie erstellen können.

Ressource	Standardkontingent
Auto-Scaling-Gruppen pro Region	500
Startkonfigurationen pro Region	200

Auto-Scaling-Gruppenkonfiguration

Ihr AWS-Konto hat die folgenden Kontingente in Bezug auf die Konfiguration von Auto Scaling Scaling-Gruppen. Sie können nicht geändert werden.

Ressource	Kontingent
Skalierungsrichtlinien pro Auto-Scaling-Gruppe	50
Geplante Vorgänge pro Auto-Scaling-Gruppe	125
Schrittanpassungen pro Richtlinie zur schrittweisen Skalierung	20
Lebenszyklus-Hooks pro Auto-Scaling-Gruppe	50
SNS-Themen pro Auto-Scaling-Gruppe	10
Classic Load Balancers pro Auto-Scaling-Gruppe	50
Elastic-Load-Balancing-Zielgruppen pro Auto-Scaling-Gruppe	50
VPC-Lattice-Zielgruppen pro Auto-Scaling-Gruppe	5

API-Operationen für Auto-Scaling-Gruppen

Amazon EC2 Auto Scaling bietet API-Vorgänge, mit denen Sie stapelweise Änderungen an Ihren Auto-Scaling-Gruppen vornehmen können. Im Folgenden sind die API-Grenzwerte für die maximale Anzahl von Elementen (maximale Array-Mitglieder) aufgeführt, die in einem einzigen Vorgang zulässig sind. Sie können nicht geändert werden.

Operation	Maximale Array-Mitglieder
AttachInstances	20 Instance-IDs
AttachLoadBalancer	10 Load Balancer
AttachLoadBalancerTargetGruppen	10 Zielgruppen
BatchDeleteScheduledAction	50 geplante Aktionen

Operation	Maximale Array-Mitglieder
BatchPutScheduledUpdateGroupAction	50 geplante Aktionen
DetachInstances	20 Instance-IDs
DetachLoadBalancer	10 Load Balancer
DetachLoadBalancerTargetGruppen	10 Zielgruppen
EnterStandby	20 Instance-IDs
ExitStandby	20 Instance-IDs
SetInstanceSchutz	50 Instance-IDs

Drosselung von Anfragen für die Amazon EC2 Auto Scaling Scaling-API

Amazon EC2 Auto Scaling API-Anfragen werden mithilfe eines Token-Bucket-Schemas gedrosselt, um die Servicebandbreite aufrechtzuerhalten. Weitere Informationen finden Sie unter API-Anforderungsrate in der Amazon EC2 Auto Scaling API-Referenz.

EC2-Beendigungsraten

Amazon EC2 Auto Scaling bestimmt dynamisch die Anzahl der Vorgänge, die zu einem Zeitpunkt ausgeführt werden können, wenn Ihre Auto-Scaling-Gruppe abskaliert. Das bedeutet, dass die Anzahl der gleichzeitig beendeten Instances in den Auto-Scaling-Gruppen variieren kann. Diese Abweichungen werden durch externe Überlegungen verursacht, z. B. ob Amazon EC2 Auto Scaling Instances bei einem Load Balancer abmelden muss.

Sonstige -Services

Kontingente für andere Dienste wie Amazon EC2 und Amazon VPC können sich auf Ihre Auto Scaling Scaling-Gruppen auswirken. Sie können verwenden Service Quotas , um die Kontingente für EC2-Instances und andere Ressourcen in Ihrem zu aktualisieren. AWS-Konto In der Service Quotas Konsole können Sie alle Ihre verfügbaren Servicekontingenten einsehen und Erhöhungen für diese beantragen. Weitere Informationen finden Sie im Service Quotas -Benutzerhandbuch unter Anfordern einer Kontingenterhöhung.

Informationen zu Kontingenten, die spezifisch für Startvorlagen sind, finden Sie unter Einschränkungen für Startvorlagen im Amazon EC2 EC2-Benutzerhandbuch.

Für die Verwendung von Amazon EC2 Auto Scaling einrichten

Bevor Sie mit der Verwendung von Amazon EC2 Auto Scaling beginnen, führen Sie die folgenden Schritte durch.

Aufgaben

- Vorbereitung für die Verwendung von Amazon EC2
- Bereiten Sie sich auf die Verwendung des vor AWS CLI

Vorbereitung für die Verwendung von Amazon EC2

Falls Sie Amazon EC2 noch nicht benutzt haben, vervollständigen Sie die Aufgaben, die in der Amazon EC2-Dokumentation beschrieben sind. Weitere Informationen finden Sie unter Einrichtung mit Amazon EC2 im Amazon EC2-Benutzerhandbuch oder Einrichtung mit Amazon EC2 im Amazon EC2 EC2-Benutzerhandbuch.

Bereiten Sie sich auf die Verwendung des vor AWS CLI

Sie können die AWS Befehlszeilentools verwenden, um Befehle an der Befehlszeile Ihres Systems auszugeben, um Amazon EC2 Auto Scaling und andere AWS Aufgaben auszuführen.

Um die AWS Command Line Interface (AWS CLI) zu verwenden, laden Sie Version 1 oder 2 von herunter, installieren und konfigurieren Sie sie. AWS CLI Version 1 und 2 verfügen über die gleiche Amazon-EC2-Auto-Scaling-Funktionalität. Informationen zum Installieren der AWS CLI -Version 1 finden Sie unter Installieren, Aktualisieren und Deinstallieren der AWS CLI im AWS CLI -Version 1 Benutzerhandbuch. Informationen zur Installation der AWS CLI Version 2 finden Sie unter Installation oder Aktualisierung der neuesten Version von AWS CLI im AWS CLI Version 2-Benutzerhandbuch.

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. Weitere Informationen finden Sie unter Erstellen Sie Auto Scaling Scaling-Gruppen über die Befehlszeile mit AWS CloudShell.

Weitere Informationen finden Sie unter autoscaling in der AWS CLI -Befehlsreferenz.

Erste Schritte mit Amazon EC2 Auto Scaling

Um mit Amazon EC2 Auto Scaling zu beginnen, können Sie Tutorials folgen, die Ihnen den Service vorstellen.

Themen

- Tutorial: Erstellen Sie Ihre erste Auto Scaling Scaling-Gruppe
- Tutorial: Einrichten einer skalierten Anwendung mit Load Balancing

Weitere Tutorials, die sich auf bestimmte Tools für die Verwaltung des Lebenszyklus von Instances in einer Auto Scaling Scaling-Gruppe konzentrieren, finden Sie in den folgenden Themen:

- Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft. Dieses Tutorial zeigt Ihnen, wie Sie Amazon verwenden, um Regeln EventBridge zu erstellen, die Lambda-Funktionen auf der Grundlage von Ereignissen aufrufen, die mit den Instances in Ihrer Auto Scaling Scaling-Gruppe passieren.
- Tutorial: Konfigurieren Sie Benutzerdaten zum Abrufen des Ziellebenszyklusstatus über Instance-Metadaten. Dieses Tutorial zeigt Ihnen, wie Sie den Instance Metadata Service (IMDS) verwenden, um eine Aktion innerhalb der Instance selbst aufzurufen.

Bevor Sie eine Auto-Scaling-Gruppe für die Verwendung mit Ihrer Anwendung erstellen, prüfen Sie die Anwendung gründlich, während sie in der AWS Cloud ausgeführt wird. Berücksichtigen Sie dabei Folgendes:

- Wie viele Availability Zones soll die Auto-Scaling-Gruppe umfassen?
- Welche vorhandenen Ressourcen (z. B. Sicherheitsgruppen oder Amazon Machine Images (AMIs)) können verwendet werden?
- Egal ob Sie eine Skalierung durchführen möchten, um die Kapazität zu erhöhen oder zu verringern, oder ob Sie einfach nur sicherstellen möchten, dass immer eine bestimmte Anzahl von Servern läuft. Beachten Sie, dass Amazon EC2 Auto Scaling diese beiden Aufgaben gleichzeitig erfüllen kann.
- Welche Metriken sind für die Leistung Ihrer Anwendung am relevantesten?
- Wie lange es dauert, einen Server zu starten und bereitzustellen.

Je besser Sie Ihre Anwendung verstehen, desto effektiver können Sie die Auto Scaling-Architektur gestalten.

Tutorial: Erstellen Sie Ihre erste Auto Scaling Scaling-Gruppe

Dieses Tutorial bietet eine praktische Einführung in Amazon EC2 Auto Scaling über die. AWS Management Console Sie erstellen eine Startvorlage, die Ihre EC2-Instances und eine Auto Scaling Scaling-Gruppe mit einer einzigen Instance definiert. Nach dem Start Ihrer Auto Scaling Scaling-Gruppe beenden Sie die Instance und überprüfen, ob die Instance außer Betrieb genommen und ersetzt wurde. Um eine konstante Anzahl von Instances aufrechtzuerhalten, erkennt Amazon EC2 Auto Scaling automatisch die Zustands- und Erreichbarkeitsprüfungen von Amazon EC2 und reagiert darauf.

Wenn Sie sich für Amazon EC2 Auto Scaling anmelden AWS, können Sie das kostenlose Kontingent AWS kostenlos nutzen. Sie können das kostenlose Kontingent verwenden, um eine `t2.micro`-Instance 12 Monate lang kostenlos zu starten und zu verwenden (in Regionen, in denen `t2.micro` nicht verfügbar ist, können Sie eine `t3.micro`-Instance im Rahmen des kostenlosen Kontingents verwenden). Wenn Sie eine Instance starten, die nicht vom kostenlosen Kontingent abgedeckt ist, fallen die standardmäßigen Amazon EC2-Nutzungsgebühren für die Instance an. Weitere Informationen finden Sie unter Amazon EC2 – Preise.

Aufgaben

- Vorbereitung auf den Walkthrough
- Schritt 1: Eine Startvorlage erstellen
- Schritt 2: Eine Auto-Scaling-Gruppe mit einer einzelnen Instance erstellen
- Schritt 3: Überprüfen Ihrer Auto-Scaling-Gruppe
- Schritt 4: Beenden einer Instance in Ihrer Auto-Scaling-Gruppe
- Schritt 5: Nächste Schritte
- Schritt 6: Bereinigen

Vorbereitung auf den Walkthrough

In diesem Walkthrough wird davon ausgegangen, dass Sie sich mit dem Starten von EC2-Instances auskennen und bereits ein Schlüsselpaar und eine Sicherheitsgruppe erstellt haben. Weitere Informationen finden Sie unter Einrichtung mit Amazon EC2 im Amazon EC2 EC2-Benutzerhandbuch.

Um mit der Verwendung von Amazon EC2 Auto Scaling zu beginnen, können Sie die Standard-VPC für Ihre verwenden. AWS-Konto Die Standard-VPC enthält ein öffentliches Standardsubnetz in jeder Availability Zone und ein Internet-Gateway, das Ihrer VPC zugeordnet ist. Sie können Ihre VPCs auf der Seite Your VPCs (Eigene VPCs)der Amazon Virtual Private Cloud (Amazon VPC)-Konsole sehen.

Schritt 1: Eine Startvorlage erstellen

In diesem Schritt erstellen Sie eine Startvorlage, die den Typ der EC2-Instance angibt, die Amazon EC2 Auto Scaling für Sie erstellt. Nehmen Sie Informationen wie die ID des Amazon-Systemabbilds (Amazon Machine Image, AMI), den Instance-Typ, das Schlüsselpaar und die Sicherheitsgruppen auf.

Eine Startvorlage erstellen

1. Öffnen Sie die Amazon EC2 EC2-Konsole und rufen Sie die Seite Launch Templates auf.
2. Wählen Sie in der oberen Navigationsleiste eine AWS-Region aus. Die Startvorlage und die Auto-Scaling-Gruppe, die Sie erstellen, sind an die von Ihnen angegebene Region gebunden.
3. Wählen Sie Startvorlage erstellen.
4. Geben Sie für Startvorlagenname **`my-template-for-auto-scaling`** ein.
5. Unter Auto-Scaling-Anleitung aktivieren Sie das Kontrollkästchen.
6. Wählen Sie für Application and OS Images (Amazon Machine Image) (Anwendungs- und Betriebssystem-Images (Amazon Machine Image)) eine Version von Amazon Linux 2 (HVM) aus der Liste Quick Start (Schnellstart) aus. Das AMI dient als grundlegende Konfigurationsvorlage für Ihre Instances.
7. Wählen Sie für Instance type (Instance-Typ) eine Hardwarekonfiguration aus, die mit dem von Ihnen angegebenen AMI kompatibel ist.
8. (Optional) Wählen Sie für Key pair name (Schlüsselpaarnamen) ein vorhandenes Schlüsselpaar aus. Sie verwenden Schlüsselpaare, um eine Verbindung zu einer Amazon-EC2-Instance mit SSH herzustellen. Informationen dazu, wie eine Verbindung mit einer Instance herstellen, sind nicht Bestandteil dieses Tutorials. Daher müssen Sie kein Schlüsselpaar angeben, es sei denn, Sie möchten eine Verbindung mit der Instance über SSH herstellen.
9. Für Network settings (Netzwerkeinstellungen) erweitern Sie die Option Advanced network configuration (Erweiterte Netzwerkkonfiguration) und tun Folgendes:

a. Wählen Sie zum Konfigurieren der primären Netzwerkschnittstelle Add network interface (Netzwerkschnittstelle hinzufügen) aus.

b. Geben Sie für Auto-Assign Public IP an, ob Ihre Instance eine öffentliche IPv4-Adresse erhält. Standardmäßig weist Amazon EC2 eine öffentliche IPv4-Adresse zu, wenn die EC2-Instance in einem Standardsubnetz gestartet wird oder wenn die Instance in einem Subnetz gestartet wird, das für die automatische Zuweisung einer öffentlichen IPv4-Adresse konfiguriert wurde. Wenn Sie keine Verbindung zu Ihrer Instance herstellen müssen, wählen Sie Disable.

c. Wählen Sie als Sicherheitsgruppen-ID eine Sicherheitsgruppe in derselben VPC aus, die Sie als VPC für Ihre Auto Scaling Scaling-Gruppe verwenden möchten. Wenn Sie keine Sicherheitsgruppe angeben, wird die Instance automatisch der Standard-Sicherheitsgruppe für die VPC zugeordnet.

d. Wählen Sie für Bei Kündigung löschen die Option Ja aus, um die Netzwerkschnittstelle zu löschen, wenn die Instance gelöscht wird.

10. Wählen Sie Startvorlage erstellen.
11. Wählen Sie auf der Bestätigungsseite Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Schritt 2: Eine Auto-Scaling-Gruppe mit einer einzelnen Instance erstellen

Gehen Sie wie folgt vor, um dort fortzufahren, wo Sie nach der Erstellung einer Startvorlage aufgehört haben.

So erstellen Sie eine Auto Scaling-Gruppe

1. Geben Sie auf der Seite Choose launch template or configuration (Startvorlage oder Konfiguration auswählen) als Name der Auto-Scaling-Gruppen **`my-first-asg`** ein.
2. Wählen Sie Weiter aus.

   Die Seite „Instance-Startoptionen auswählen“ wird angezeigt, auf der Sie die VPC-Netzwerkeinstellungen auswählen können, die die Auto Scaling Scaling-Gruppe verwenden soll, und die Ihnen Optionen für den Start von On-Demand- und Spot-Instances bietet.
3. Lassen Sie VPC im Bereich Netzwerk auf die von Ihnen gewählte Standard-VPC eingestellt AWS-Region, oder wählen Sie Ihre eigene VPC aus. Die Standard-VPC wird automatisch so

konfiguriert, dass sie eine Internetverbindung für Ihre Instance bereitstellt. Diese VPC umfasst ein öffentliches Subnetz in jeder Availability Zone in der Region.

4. Wählen Sie für Availability Zones and subnets (Subnetze) ein Subnetz für jede Availability Zone aus, die Sie einschließen möchten. Verwenden Sie Subnetze in mehreren Availability Zones, um eine hohe Verfügbarkeit zu erzielen. Weitere Informationen finden Sie unter Überlegungen bei der Auswahl von VPC-Subnetzen.
5. Verwenden Sie im Abschnitt Instance type requirements (Anforderungen an den Instance-Typ) die Standardeinstellung, um diesen Schritt zu vereinfachen. (Setzen Sie die Startvorlage nicht außer Kraft.) In diesem Tutorial werden Sie nur eine On-Demand-Instance mit dem in Ihrer Startvorlage angegebenen Instance-Typ starten.
6. Behalten Sie die restlichen Standardeinstellungen für dieses Tutorial bei, und wählen Sie Skip to review (Mit Prüfen fortfahren).

> **Note**
>
> Die anfängliche Größe der Gruppe wird durch ihre gewünschte Kapazität bestimmt. Der Standardwert ist 1-Instance.

7. Überprüfen Sie auf der Seite Review (Überprüfen) die Informationen für die Gruppe und wählen Sie dann Create Auto Scaling Group (Auto-Scaling-Gruppe erstellen) aus.

Schritt 3: Überprüfen Ihrer Auto-Scaling-Gruppe

Nun, da Sie eine Auto-Scaling-Gruppe erstellt haben, können Sie überprüfen, ob die Gruppe eine EC2-Instance gestartet hat.

> **Tip**
>
> Im folgenden Verfahren sehen Sie sich die Abschnitte Activity history (Verlauf der Aktivität) und Instances für die Auto- Scaling-Gruppe an. In beiden sollten die benannten Spalten bereits angezeigt werden. Um ausgeblendete Spalten anzuzeigen oder die Anzahl der angezeigten Zeilen zu ändern, wählen Sie das Zahnradsymbol in der oberen rechten Ecke jedes Abschnitts, um die Einstellungen zu öffnen, die Einstellungen nach Bedarf zu aktualisieren und Confirm (Bestätigen) auszuwählen.

So prüfen Sie, dass Ihre Auto-Scaling-Gruppe eine EC2-Instance gestartet hat

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe, die Sie gerade erstellt haben.

   Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet. Die erste verfügbare Registerkarte ist die Registerkarte Details, die Informationen zur Auto-Scaling-Gruppe anzeigt.
3. Wählen Sie die zweite Registerkarte mit de Bezeichnung Activity (Aktivität). Unter Activity history (Aktivitätsverlauf) können Sie sich den Fortschritt der Aktivitäten anzeigen lassen, die der Auto-Scaling-Gruppe zugeordnet sind. In der Status-Spalte wird der aktuelle Status Ihrer Instance angezeigt. Während die Instance gestartet wird, zeigt die Statusspalte `Not yet in service` an. Nach dem Start der Instance ändert sich der Status in `Successful`. Sie können auch die Aktualisierungsschaltfläche verwenden, um den aktuellen Status der Instance anzuzeigen.
4. Auf der Registerkarte Instance management (Instance-Verwaltung), unter Instances, können Sie sich den Status der Instance ansehen.
5. Stellen Sie sicher, dass Ihre Instance erfolgreich gestartet wurde. Es dauert einige Zeit, bis die Instance startet.

   - In der Spalte Lifecycle (Lebenszyklus) wird Ihnen der Zustand Ihrer Instance angezeigt. Die Instance befindet sich zunächst im Status `Pending`. Wenn eine Instance für den Empfang von Datenverkehr bereit ist, lautet der Status `InService`.
   - In der Spalte Health Status wird das Ergebnis der Amazon EC2 Auto Scaling Scaling-Zustandsprüfungen für Ihre Instance angezeigt.

Schritt 4: Beenden einer Instance in Ihrer Auto-Scaling-Gruppe

Sie können diese Schritte nutzen, um mehr über die Funktionsweise von Amazon EC2 Auto Scaling zu erfahren, insbesondere darüber, wie es bei Bedarf neue Instances startet. Die Mindestgröße für die Auto-Scaling-Gruppe, die Sie in diesem Tutorial erstellt haben, ist eine Instance. Deshalb muss Amazon EC2 Auto Scaling eine neue Instance als Ersatz starten, wenn Sie diese laufende Instance beenden.

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

3. Wählen Sie auf der Registerkarte Instance management (Instance-Verwaltung) unter Instances die ID der Instance aus.

   Dadurch gelangen Sie zur Seite Instances in der Amazon-EC2-Konsole, auf der Sie die Instance beenden können.
4. Wählen Sie Actions, Instance State und Terminate aus. Wählen Sie Yes, Terminate aus, wenn Sie zum Bestätigen aufgefordert werden.
5. Wählen Sie im Navigationsbereich unter Auto Scaling Auto Scaling Groups (Auto Scaling-Gruppe) aus. Wählen Sie Ihre Auto-Scaling-Gruppe aus und wählen Sie die Registerkarte Activity (Aktivität).

   Wenn Sie eine Instance auf der Instance-Seite beenden, dauert es nach dem Beenden der Instance ein oder zwei Minuten, bis eine neue Instance gestartet wird. Im Aktivitätsverlauf sehen Sie beim Start der Skalierungsaktivität einen Eintrag für die Beendigung der ersten Instance und einen Eintrag für den Start einer neuen Instance. Verwenden Sie die Schaltfläche „Aktualisieren“, bis Sie die neuen Einträge sehen.
6. Auf der Registerkarte Instance management (Instance-Verwaltung) wird im Abschnitt Instances nur die neue Instance angezeigt.
7. Wählen Sie im Navigationsbereich unter Instances die Option Instances aus. Diese Seite zeigt sowohl die beendete als auch die neue laufende Instance.

Schritt 5: Nächste Schritte

Fahren Sie mit dem nächsten Schritt fort, wenn Sie die Basisinfrastruktur löschen möchten, die Sie gerade erstellt haben. Andernfalls können Sie diese Infrastruktur als Grundlage verwenden und die folgenden Aktionen ausprobieren:

- Herstellen einer Verbindung zu Ihrer Linux-Instance über Session Manager. Weitere Informationen finden Sie unter Connect zu Ihrer Linux-Instance mithilfe von Session Manager und Connect zu Ihrer Linux-Instance von Linux oder macOS mithilfe von SSH im Amazon EC2 EC2-Benutzerhandbuch.
- Konfigurieren Sie eine SNS-Benachrichtigung, die Sie benachrichtigt, sobald Ihre Auto-Scaling-Gruppe Instances startet oder beendet. Weitere Informationen finden Sie unter Amazon SNS-Benachrichtigungsoptionen.
- Skalieren Sie Ihre Auto-Scaling-Gruppe manuell, um die SNS-Benachrichtigung zu testen. Weitere Informationen finden Sie unter Ändern der gewünschten Kapazität einer Auto-Scaling-Gruppe.

Sie können sich auch mit den Konzepten der auto Skalierung vertraut machen, indem Sie Skalierungsrichtlinien für die Ziel-Nachverfolgung. Wenn sich die Auslastung Ihrer Anwendung ändert, kann Ihre Auto Scaling-Gruppe automatisch aufskalieren (Instanzen hinzufügen) und abskalieren (weniger Instanzen ausführen), indem die gewünschte Kapazität der Gruppe zwischen der minimalen und der maximalen Kapazitätsgrenze angepasst wird. Weitere Informationen zu diesen Limits finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.

Schritt 6: Bereinigen

Sie können entweder Ihre Skalierungsinfrastruktur löschen oder nur Ihre Auto Scaling Scaling-Gruppe löschen und Ihre Startvorlage behalten, um sie später zu verwenden.

Wenn Sie eine Instance gestartet haben, die nicht unter das kostenlose Kontingent für AWS fällt, sollten Sie die Instance beenden, um zusätzliche Gebühren zu vermeiden. Wenn Sie die Instance beenden, werden auch die damit verknüpften Daten gelöscht.

So löschen Sie Ihre Auto-Scaling-Gruppe

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe (`my-first-asg`).
3. Wählen Sie Löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`delete`** ein, um das Löschen der angegebenen Auto-Scaling-Gruppe zu löschen, wählen Sie dann Löschen.

   Ein Ladesymbol in der Spalte Name zeigt an, dass die Auto-Scaling-Gruppe gelöscht wird. Wenn der Löschvorgang erfolgt ist, zeigen die Spalten Desired (Gewünscht), Min und Max `0`-Instances für die Auto-Scaling-Gruppe an. Es dauert einige Minuten, bis die Instance beendet und die Gruppe gelöscht werden. Aktualisieren Sie die Liste, um den aktuellen Status anzuzeigen.

Überspringen Sie folgenden Schritte, falls Sie die Startvorlage behalten möchten.

So löschen Sie eine Startvorlage

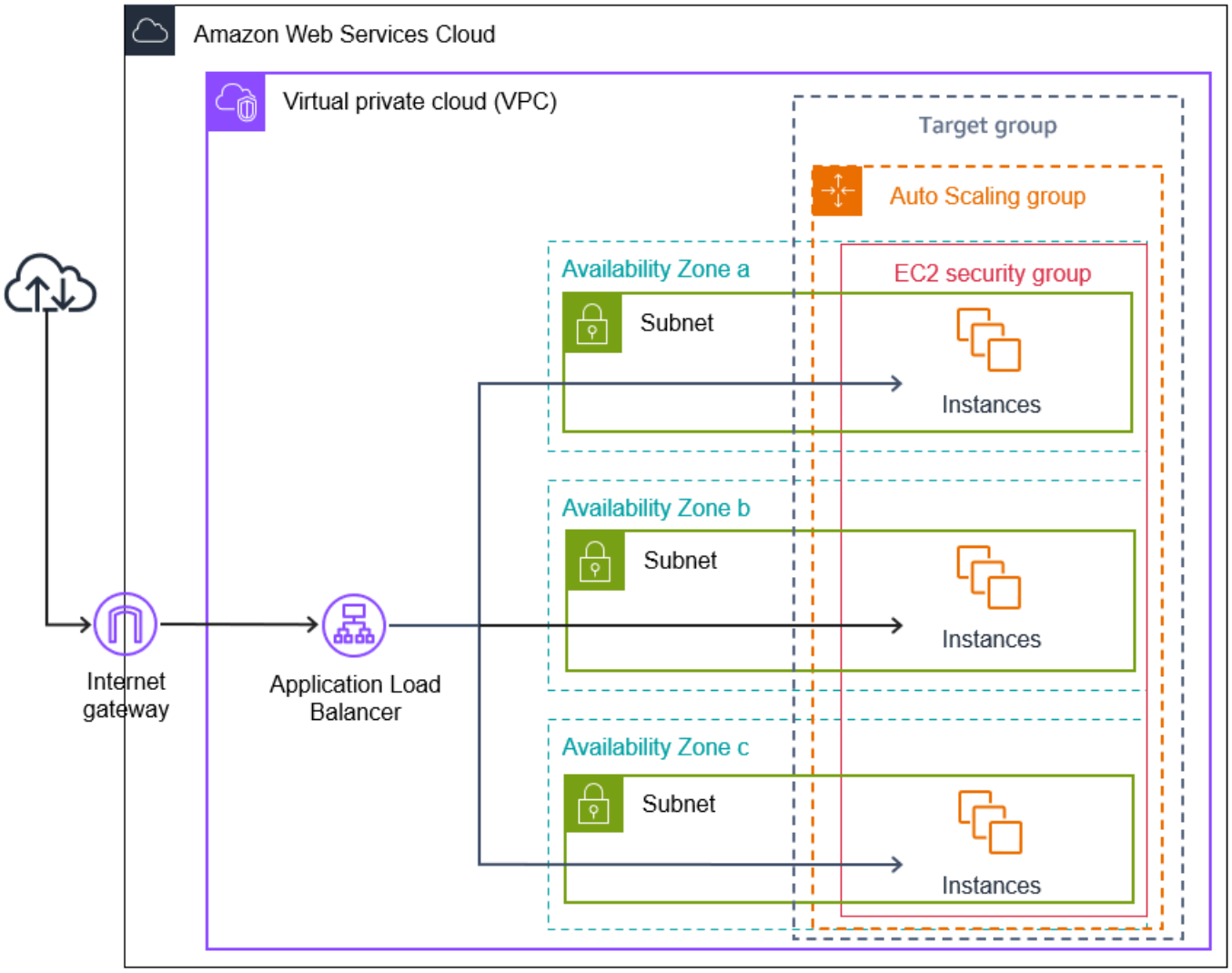
1. Öffnen Sie die Seite Startvorlagen der Amazon EC2 Konsole.
2. Wählen Sie Ihre Startvorlage aus (`my-template-for-auto-scaling`).
3. Wählen Sie Actions (Aktionen) und Delete template (Vorlage löschen) aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`Delete`** ein, um das Löschen der angegebenen Auto-Scaling-Gruppe zu bestätigen, wählen Sie dann Löschen.

Tutorial: Einrichten einer skalierten Anwendung mit Load Balancing

> **Important**
>
> Bevor Sie sich mit diesem Tutorial befassen, empfehlen wir Ihnen, zunächst das folgende einführende Tutorial zu lesen: Erstellen Sie Ihre erste Auto Scaling Scaling-Gruppe.

Wenn Sie Ihre Auto-Scaling-Gruppe bei einem Elastic Load Balancing Load Balancer registrieren, können Sie eine Anwendung mit Lastenausgleich einrichten. Elastic Load Balancing arbeitet mit Amazon EC2 Auto Scaling zusammen, um eingehenden Datenverkehr auf Ihre gesunden Amazon EC2-Instances zu verteilen. Dies erhöht die Skalierbarkeit und Verfügbarkeit Ihrer Anwendung. Sie können Elastic Load Balancing innerhalb mehrerer Availability Zones aktivieren, um die Fehlertoleranz Ihrer Anwendungen zu erhöhen.

In diesem Tutorial werden die grundlegenden Schritte zum Einrichten einer Anwendung mit Lastenausgleich beschrieben, wenn die Auto-Scaling-Gruppe erstellt wird. Wenn Sie fertig sind, sollte Ihre Architektur wie das folgende Diagramm aussehen:



Elastic Load Balancing unterstützt verschiedene Load Balancer-Typen. Wir empfehlen Ihnen, für diese praktische Anleitung einen Application Load Balancer zu verwenden.

Weitere Informationen zum Einführen eines Load Balancer in Ihre Architektur finden Sie unter Um den Datenverkehr über die Instances in Ihrer Auto-Scaling-Gruppe zu verteilen, verwenden Sie Elastic-Load-Balancing..

Aufgaben

- Voraussetzungen
- Schritt 1: Einrichten einer Startvorlage oder Startkonfiguration
- Schritt 2: Erstellen einer Auto-Scaling-Gruppe
- Schritt 3: Überprüfen Sie, ob Ihr Load Balancer angefügt ist
- Schritt 4: Nächste Schritte

- Schritt 5: Bereinigen
- Zugehörige Ressourcen

Voraussetzungen

- Einen Load Balancer und eine Zielgruppe. Stellen Sie sicher, dass Sie die gleiche Availability Zones für den Load Balancer auswählen, die Sie auch für Ihre Auto-Scaling-Gruppe aktivieren möchten. Weitere Informationen finden Sie unter Erste Schritte mit Elastic Load Balancing im Elastic Load Balancing-Benutzerhandbuch.
- Eine Sicherheitsgruppe für Ihre Startvorlage oder Startkonfiguration. Die Sicherheitsgruppe muss den Zugriff vom Load Balancer sowohl auf den Listener-Port (normalerweise Port 80 für HTTP-Datenverkehr) als auch auf den Port, den Elastic Load Balancing für Zustandsprüfungen verwenden soll, erlauben. Weitere Informationen finden Sie in der entsprechenden Dokumentation:
  - Zielsicherheitsgruppen im Benutzerhandbuch für Application Load Balancer
  - Zielsicherheitsgruppen im Benutzerhandbuch für Network Load Balancer

  Wenn die Instances öffentliche IP-Adressen aufweisen, können Sie optional SSH-Datenverkehr zulassen, wenn Sie eine Verbindung zu den Instances herstellen müssen.
- (Optional) Eine IAM-Rolle, die Ihrer Anwendung Zugriff auf AWS gewährt.
- (Optional) Sie haben ein Amazon Machine Image (AMI) als Quellvorlage für Ihre Amazon EC2-Instances definiert. Um jetzt eine zu erstellen, starten Sie eine Instance. Geben Sie die IAM-Rolle (sofern erstellt) und alle benötigten Konfigurationsskripts als Benutzerdaten an. Stellen Sie eine Verbindung mit der Instance her und passen Sie sie an. Beispielsweise können Sie Software und Anwendungen installieren, Daten kopieren und weitere EBS-Volumes anfügen. Testen Sie Ihre Anwendungen auf der Instance, um sicherzustellen, dass sie ordnungsgemäß konfiguriert ist. Speichern Sie diese aktualisierte Konfiguration als benutzerdefiniertes AMI. Sie können die Instance beenden, wenn Sie sie zu einem späteren Zeitpunkt nicht mehr benötigen. Instances, die über dieses neue benutzerdefinierte AMI gestartet werden, enthalten die Anpassungen, die Sie beim Erstellen des AMI vorgenommen haben.
- Eine Virtual Private Cloud (VPC). Dieses Tutorial bezieht sich auf die Standard-VPC, Sie können aber auch Ihre eigene verwenden. Stellen Sie in letzterem Fall sicher, dass Ihre VPC über ein Subnetz verfügt, das jeder Availability Zone der Region zugeordnet ist, in der Sie arbeiten. Mindestens zwei öffentliche Subnetze müssen verfügbar sein, um den Load Balancer zu erstellen. Sie müssen außerdem über zwei private Subnetze oder zwei öffentliche Subnetze verfügen, um Ihre Auto-Scaling-Gruppe zu erstellen und sie beim Load Balancer zu registrieren.

Schritt 1: Einrichten einer Startvorlage oder Startkonfiguration

Verwenden Sie entweder eine Startvorlage oder eine Startkonfiguration für diese praktische Anleitung.

Themen

Wählen oder erstellen Sie eine Startvorlage

Wenn Sie bereits über eine Startvorlage verfügen, die Sie verwenden möchten, wählen Sie sie wie folgt aus.

So wählen Sie eine vorhandene Startvorlage aus

1. Öffnen Sie die Seite Startvorlagen der Amazon EC2 Konsole.
2. Wählen Sie in der Navigationsleiste am oberen Bildschirmrand die Region aus, in der der Load Balancer erstellt wurde.
3. Wählen Sie eine Startvorlage aus.
4. Wählen Sie Actions (Aktionen), Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Alternativ können Sie wie folgt eine neue Startvorlage erstellen.

Eine Startvorlage erstellen

1. Öffnen Sie die Seite Startvorlagen der Amazon EC2 Konsole.
2. Wählen Sie in der Navigationsleiste am oberen Bildschirmrand die Region aus, in der der Load Balancer erstellt wurde.
3. Wählen Sie Startvorlage erstellen.
4. Geben Sie einen Namen und eine Beschreibung für die anfängliche Version der Startvorlage ein.
5. Wählen Sie für Application and OS Images (Amazon Machine Image) (Anwendungs- und Betriebssystem-Images (Amazon Machine Image)) die ID der AMI für Ihre Instances aus. Sie können alle verfügbaren AMIs durchsuchen oder ein AMI aus der Liste Recents (Kürzlich) oder Quickstart (Schnellstart) auswählen. Wenn das benötigte AMI nicht angezeigt wird, wählen

Sie Browse more AMIs (Mehr AMIs durchsuchen), um den vollständigen AMI-Katalog zu durchsuchen.

6. Wählen Sie für Instance type eine Hardwarekonfiguration für Ihre Instances aus, die mit dem von Ihnen angegebenen AMI kompatibel ist.
7. (Optional) Geben Sie für Schlüsselpaar (Anmeldung) das Schlüsselpaar ein, das Sie bei der Verbindung mit Ihren Instances verwenden möchten.
8. Für Network settings (Netzwerkeinstellungen) erweitern Sie die Option Advanced network configuration (Erweiterte Netzwerkkonfiguration) und tun Folgendes:

   a. Wählen Sie zum Konfigurieren der primären Netzwerkschnittstelle Add network interface (Netzwerkschnittstelle hinzufügen) aus.

   b. Geben Sie für Automatische Zuweisung öffentlicher IP-Adressen an, ob Ihre Instances öffentliche IPv4-Adressen erhalten. Standardmäßig weist Amazon EC2 eine öffentliche IPv4-Adresse zu, wenn die EC2-Instance in einem Standardsubnetz gestartet wird oder wenn die Instance in einem Subnetz gestartet wird, das für die automatische Zuweisung einer öffentlichen IPv4-Adresse konfiguriert wurde. Wenn Sie keine Verbindung zu Ihren Instances herstellen müssen, können Sie Disable wählen, um zu verhindern, dass Instances in Ihrer Gruppe Traffic direkt aus dem Internet empfangen. In diesem Fall empfangen sie nur den Datenverkehr vom Load Balancer.

   c. Für Sicherheitsgruppen-ID geben Sie eine Sicherheitsgruppe für Ihre Instances aus derselben VPC wie dem Load Balancer an.

   d. Für Beim Beenden löschen wählen Sie Ja aus. Damit wird die Netzwerkschnittstelle gelöscht, wenn die Auto-Scaling-Gruppe nach unten skaliert wird und die Instance, der die Netzwerkschnittstelle angefügt ist, beendet wird.
9. (Optional) Um Anmeldeinformationen für Ihre Instances sicher zu verteilen, geben Sie bei Advanced details (Erweiterte Details), IAM instance profile (IAM-Instance-Profil) den Amazon-Ressourcennamen (ARN) Ihrer IAM-Rolle ein.
10. (Optional) Wenn Sie Benutzerdaten oder ein Konfigurationsskript für Ihre Instances angeben möchten, fügen Sie dies unter Advanced Details, User data ein.
11. Wählen Sie Startvorlage erstellen.
12. Wählen Sie auf der Bestätigungsseite Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Auswählen oder Erstellen einer Startkonfiguration

> **Note**
>
> Wir raten dringend davon ab, Startkonfigurationen in neuen Anwendungen zu verwenden, da es sich um eine veraltete Funktion handelt, für die keine geplanten Investitionen erforderlich sind. Darüber hinaus haben neue Konten, die am oder nach dem 1. Juni 2023 erstellt wurden, nicht die Möglichkeit, neue Startkonfigurationen über die Konsole zu erstellen. Weitere Informationen finden Sie unter Startkonfigurationen.

So wählen Sie eine vorhandene Startkonfiguration aus

1. Öffnen Sie die Seite Konfigurationen starten der Amazon EC2-Konsole.
2. Wählen Sie in der oberen Navigationsleiste die Region aus, in der der Load Balancer erstellt wurde.
3. Wählen Sie eine Startkonfiguration aus.
4. Wählen Sie Actions (Aktionen), Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Alternativ können Sie wie folgt eine neue Startkonfiguration erstellen.

Erstellen Sie eine Startkonfiguration wie folgt:

1. Öffnen Sie die Seite Konfigurationen starten der Amazon EC2-Konsole. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Startkonfigurationen anzeigen aus, um zu bestätigen, dass Sie die Seite Startkonfigurationen aufrufen möchten.
2. Wählen Sie in der oberen Navigationsleiste die Region aus, in der der Load Balancer erstellt wurde.
3. Klicken Sie auf Erstellen einer Startkonfiguration und geben Sie einen Namen für die Startkonfiguration ein.
4. Geben Sie für Amazon Machine Image (AMI), die ID des AMI für Ihre Instances als Suchkriterien ein.
5. Für Instance-Typ wählen Sie eine Hardware-Konfiguration für Ihre Instance aus.
6. Unter Zusätzliche Konfiguration achten Sie auf die folgenden Felder:

a. (Optional) Wählen Sie zum sicheren Verteilen von Anmeldeinformationen an die EC2-Instance für IAM-Instance-Profil Ihre IAM-Rolle aus. Weitere Informationen finden Sie unter IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden.

b. (Optional) Wenn Sie Benutzerdaten oder ein Konfigurationsskript für Ihre Instance angeben möchten, fügen Sie dies unter Advanced Details, User data ein.

c. (Optional) Behalten Sie für Erweiterte Details, IP-Adresstyp den Standardwert bei. Wenn Sie Ihre Auto-Scaling-Gruppe erstellen, können Sie Instances in Ihrer Auto-Scaling-Gruppe eine öffentliche IP-Adresse zuweisen, indem Sie Subnetze verwenden, für die das Attribut der öffentlichen IP-Adressierung aktiviert ist, z. B. die Standard-Subnetze in der Standard-VPC. Wenn Sie keine Verbindung zu Ihren Instances herstellen müssen, können Sie auch Keine öffentliche IP-Adresse Instances hinzuweisen auswählen, um zu verhindern, dass Instances in Ihrer Gruppe Datenverkehr direkt aus dem Internet empfangen. In diesem Fall empfangen sie nur den Datenverkehr vom Load Balancer.

7. Wählen Sie für Sicherheitsgruppen eine vorhandene Sicherheitsgruppe aus derselben VPC wie dem Load Balancer aus. Wenn Sie die Option Erstellen einer neuen Sicherheitsgruppe ausgewählt lassen, wird eine Standard-SSH-Regel für Amazon-EC2-Instances konfiguriert, auf denen Linux ausgeführt wird. Für Amazon-EC2-Instances, die Windows ausführen, wird eine Standard-RDP-Rolle konfiguriert.

8. Für Schlüsselpaar (Login) wählen Sie eine Option unter Optionen für Schlüsselpaar aus.

   Wenn Sie ein Amazon EC2-Instance-Schlüsselpaar bereits konfiguriert haben, können Sie das Schlüsselpaar hier auswählen.

   Wenn Sie noch kein Amazon EC2-Instance-Schlüsselpaar haben, klicken Sie auf Create a new key pair (Ein neues Schlüsselpaar erstellen) und geben Sie einen wiedererkennbaren Namen ein. Wählen Sie Download Key Pair (Schlüsselpaar herunterladen) aus, um das Schlüsselpaar auf Ihrem Computer herunterzuladen.

   > **Important**
   >
   > Wählen Sie nicht Proceed without a key pair (Ohne Schlüsselpaar fortfahren) aus, wenn Sie eine Verbindung mit Ihrer Instance herstellen müssen.

9. Aktivieren Sie das Bestätigungskontrollkästchen und wählen Sie dann Create launch configuration (Startkonfiguration erstellen) aus.

10. Aktivieren Sie das Kontrollkästchen neben dem Namen der neuen Startkonfiguration und wählen Sie Aktionen, Auto-Scaling-Gruppe erstellen aus.

Schritt 2: Erstellen einer Auto-Scaling-Gruppe

Gehen Sie wie folgt vor, um den Vorgang dort fortzusetzen, wo Sie ihn nach dem Erstellen oder Auswählen der Startvorlage oder der Startkonfiguration abgebrochen haben.

So erstellen Sie eine Auto-Scaling-Gruppe

1. Geben Sie auf der Seite Startvorlage oder -konfiguration auswählen für Auto-Scaling-Gruppenname einen Namen für Ihre Auto-Scaling-Gruppe ein.
2. [Nur Startvorlage] Wählen Sie unter Launch template version (Version der Startvorlage) aus, ob die Auto Scaling-Gruppe beim horizontalen Skalieren nach oben die standardmäßige, die neueste oder eine bestimmte Version der Startvorlage verwenden soll.
3. Wählen Sie Weiter aus.

   Die Seite Choose instance launch options (Startoptionen für Instances auswählen) wird angezeigt. Hier können Sie die VPC-Netzwerkeinstellungen auswählen, welche die Auto-Scaling-Gruppe verwenden soll, und Sie erhalten Optionen für den Start von On-Demand- und Spot-Instances (wenn Sie eine Startvorlage ausgewählt haben).
4. Wählen Sie im Abschnitt Netzwerk für VPC die VPC, die Sie für Ihren Load Balancer verwendet haben. Wenn Sie die Standard-VPC auswählen, wird sie automatisch so konfiguriert, dass sie Internetkonnektivität für Ihre Instances bereitstellt. Diese VPC umfasst ein öffentliches Subnetz in jeder Availability Zone in der Region.
5. Wählen Sie für Availability Zone and Subnets (Subnetze) mindestens ein Subnetz aus jeder Availability Zone aus, das Sie einbeziehen möchten, basierend auf den Availability Zones, in denen sich der Load Balancer befindet. Weitere Informationen finden Sie unter Überlegungen bei der Auswahl von VPC-Subnetzen.
6. [Nur Vorlage starten] Im Abschnitt Anforderungen an den Instance-Typ verwenden Sie die Standardeinstellung, um diesen Schritt zu vereinfachen. (Setzen Sie die Startvorlage nicht außer Kraft.) In diesem Tutorial werden Sie nur On-Demand-Instances mit dem in Ihrer Startvorlage angegebenen Instance-Typ starten.
7. Wählen Sie Next (Weiter), um zur Seite Configure advanced options (Erweiterte Optionen konfigurieren) zu gelangen.

8. Um die Gruppe an einen bestehenden Load-Balancer anzuhängen, wählen Sie im Abschnitt Load balancing (Lastenverteilung vornehmen) die Option Attach to an existing load balancer (An eine bestehende Lastenverteilung anhängen). Sie können Wählen Sie aus Ihren Load-Balancer-Zielgruppen oder Wählen Sie aus den klassischen Load Balancers auswählen. Sie können dann den Namen einer Zielgruppe für den von Ihnen erstellten Application Load Balancer oder Network Load Balancer wählen oder den Namen eines Classic Load Balancers auswählen.
9. (Optional) Um Ihre Elastic Load Balancing-Zustandsprüfungen zu aktivieren, wählen Sie unter Health check type (Zustandsprüfungstyp) für Health checks (Zustandsprüfungen) ELB aus.
10. Wenn Sie die Konfiguration der Auto-Scaling-Gruppe abgeschlossen haben, wählen Sie Skip to review (Mit Prüfung fortfahren) aus.
11. Prüfen Sie auf der Seite Review (Überprüfung) die Details Ihrer Auto-Scaling-Gruppe. Sie können Edit (Bearbeiten) auswählen, um Änderungen vorzunehmen. Wählen Sie Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus, wenn Sie fertig sind.

Nachdem Sie die Auto-Scaling-Gruppe mit dem angefügtem Load Balancer erstellt haben, registriert der Load Balancer automatisch neue Instances, sobald diese online geschaltet werden. Sie haben zu diesem Zeitpunkt nur eine Instance, daher gibt es nicht viel zu registrieren. Sie können jedoch zusätzliche Instances hinzufügen, indem Sie die gewünschte Kapazität der Gruppe aktualisieren. step-by-step Eine Anleitung dazu finden Sie unterÄndern der gewünschten Kapazität einer Auto-Scaling-Gruppe.

Schritt 3: Überprüfen Sie, ob Ihr Load Balancer angefügt ist

So überprüfen Sie, ob Ihr Load Balancer angefügt ist

1. Wählen Sie auf der Seite Auto-Scaling-Gruppen Amazon EC2-Konsole das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe aus.
2. Auf der Registerkarte Details werden unter Load Balancing (Lastenausgleich) alle angefügten Load Balancer-Zielgruppen oder Classic Load Balancers angezeigt.
3. Auf der Registerkarte Aktivität können Sie unter Aktivitätsverlauf überprüfen, ob Ihre Instances erfolgreich gestartet wurden. Die Spalte Status zeigt an, ob Ihre Auto-Scaling-Gruppe erfolgreich Instances gestartet hat. Wenn Ihre Instances nicht gestartet werden können, finden Sie Tipps zur Fehlerbehebung für häufige Instance-Startprobleme unter Fehlersuche bei Amazon EC2 Auto Scaling.
4. Sie können auf der Registerkarte Instance-Verwaltung unter Instances überprüfen, ob Ihre Instances Verkehr empfangen können. Anfänglich sind Ihre Instances im Status `Pending`. Wenn

eine Instance für den Empfang von Datenverkehr bereit ist, lautet der Status `InService`. Die Spalte Health Status (Zustandsstatus) enthält das Ergebnis der Amazon EC2 Auto Scaling-Zustandsprüfungen für Ihre Instances. Obwohl eine Instance als fehlerfrei gekennzeichnet ist, sendet der Load Balancer Datenverkehr nur an Instances, welche die Load Balancer-Zustandsprüfungen bestehen.

5. Vergewissern Sie sich, dass Ihre Instances beim Load Balancer registriert sind. Öffnen Sie die Zielgruppen-Seite der Amazon EC2-Konsole. Wählen Sie Ihre Zielgruppe und danach die Registerkarte Ziele aus. Wenn der Zustand der Instances `initial` ist, liegt das wahrscheinlich daran, dass sie noch im Prozess der Registrierung sind, oder sie werden immer noch einer Zustandsprüfung unterzogen. Wenn der Zustand der Instances `healthy` ist, sind sie bereit zur Verwendung.

Schritt 4: Nächste Schritte

Nachdem Sie nun dieses Tutorial abgeschlossen haben, können Sie hier mehr erfahren:

- Amazon EC2 Auto Scaling bestimmt anhand des Status der von Ihrer Auto-Scaling-Gruppe verwendeten Zustandsprüfungen, ob eine Instance fehlerfrei ist. Wenn Sie Load Balancer-Integritätsprüfungen aktivieren und eine Instance die Integritätsprüfungen nicht besteht, betrachtet Ihre Auto Scaling Scaling-Gruppe die Instance als fehlerhaft und ersetzt sie. Weitere Informationen finden Sie unter Health checks (Zustandsprüfungen).
- Sie können Ihre Anwendung auf eine zusätzliche Availability Zone in derselben Region erweitern, um die Fehlertoleranz im Falle einer Service-Unterbrechung zu erhöhen. Weitere Informationen finden Sie unter Hinzufügen von Availability Zones.
- Sie können die Auto-Scaling-Gruppe für die Verwendung einer Ziel-Nachverfolgung-Skalierungsrichtlinie konfigurieren. Dadurch wird die Anzahl der Instances automatisch erhöht oder verringert, wenn sich der Bedarf für Ihre Instances ändert. Auf diese Weise kann die Gruppe Änderungen an der Menge des Datenverkehrs verarbeiten, den Ihre Anwendung erhält. Weitere Informationen finden Sie unter Skalierungsrichtlinien für die Ziel-Nachverfolgung.

Schritt 5: Bereinigen

Nachdem Sie mit den für diese praktische Anleitung erstellten Ressourcen abgeschlossen haben, sollten Sie sie bereinigen, um unnötige Kosten zu vermeiden.

So löschen Sie Ihre Auto-Scaling-Gruppe

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.
3. Wählen Sie Löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`delete`** ein, um das Löschen der angegebenen Auto-Scaling-Gruppe zu löschen, wählen Sie dann Löschen.

   Ein Ladesymbol in der Spalte Name zeigt an, dass die Auto-Scaling-Gruppe gelöscht wird. Wenn der Löschvorgang erfolgt ist, zeigen die Spalten Desired (Gewünscht), Min und Max `0`-Instances für die Auto-Scaling-Gruppe an. Es dauert einige Minuten, bis die Instance beendet und die Gruppe gelöscht werden. Aktualisieren Sie die Liste, um den aktuellen Status anzuzeigen.

Überspringen Sie folgenden Schritte, falls Sie die Startvorlage behalten möchten.

So löschen Sie eine Startvorlage

1. Öffnen Sie die Seite Startvorlagen der Amazon EC2 Konsole.
2. Wählen Sie Ihre Startvorlage aus.
3. Wählen Sie Actions (Aktionen) und Delete template (Vorlage löschen) aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`Delete`** ein, um das Löschen der angegebenen Auto-Scaling-Gruppe zu bestätigen, wählen Sie dann Löschen.

Überspringen Sie folgenden Schritte, falls Sie die Startkonfiguration behalten möchten.

Löschen Sie eine Startkonfiguration wie folgt:

1. Öffnen Sie die Seite Konfigurationen starten der Amazon EC2-Konsole.
2. Wählen Sie Ihre Startkonfiguration aus.
3. Wählen Sie Actions, Delete launch configuration aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).

Überspringen Sie das folgende Verfahren, wenn Sie den Load Balancer für eine spätere Verwendung behalten möchten.

Löschen Sie den Load Balancer wie folgt:

1. Öffnen Sie die Seite Load Balancers in der Amazon EC2-Konsole.
2. Wählen Sie den Load Balancer aus und klicken Sie auf Actions (Aktionen) und dann auf Delete (Löschen).
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Delete (Ja, löschen).

So löschen Sie Ihre Zielgruppe

1. Öffnen Sie die Zielgruppen-Seite der Amazon EC2-Konsole.
2. Wählen Sie Ihre Zielgruppe aus und klicken Sie dann auf Actions (Aktionen), Delete (Löschen).
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Ja, löschen.

Zugehörige Ressourcen

Mit AWS CloudFormation können Sie AWS Infrastrukturbereitstellungen vorhersehbar und wiederholt erstellen und bereitstellen, indem Sie mithilfe von Vorlagendateien eine Sammlung von Ressourcen als eine Einheit (einen Stapel) erstellen und löschen. Weitere Informationen finden Sie im AWS CloudFormation -Benutzerhandbuch.

Eine exemplarische Vorgehensweise, die Ihnen zeigt, wie Sie mit einer Stack-Vorlage eine Auto-Scaling-Gruppe hinter einem Application Load Balancer bereitstellen, finden Sie unter Exemplarische Vorgehensweise: Erstellen einer skalierten Anwendung mit Lastenausgleich im AWS CloudFormation -Benutzerhandbuch. Verwenden Sie die exemplarische Vorgehensweise und die Mustervorlagen als Ausgangspunkt für die Erstellung ähnlicher Vorlagen, die Ihren Anforderungen entsprechen.

Startvorlagen für Amazon EC2 Auto Scaling

Eine Startvorlage ist ähnlich wie eine Startkonfiguration und gibt wie diese Instance-Konfigurationsinformationen an. Enthalten sind die ID des Amazon Machine Image (AMI), der Instance-Typ, ein Schlüsselpaar, Sicherheitsgruppen und die anderen Parameter, die Sie zum Starten von EC2-Instances verwenden. Durch das Definieren einer Startvorlage anstelle einer Startkonfiguration sind jedenfalls mehrere Versionen einer Startvorlage möglich.

Mit dem Versioning von Startvorlagen können Sie eine Teilmenge des vollständigen Satzes an Parametern erstellen. Anschließend können Sie es erneut verwenden, um andere Versionen derselben Startvorlage zu erstellen. Sie können beispielsweise eine Startvorlage erstellen, die eine Basiskonfiguration ohne AMI- oder Benutzerdatenskript definiert. Nachdem Sie Ihre Startvorlage erstellt haben, können Sie eine neue Version erstellen und die AMI- und Benutzerdaten mit der neuesten Version Ihrer Anwendung zum Testen hinzufügen. Dies führt zu zwei Versionen der Startvorlage. Das Speichern einer Basiskonfiguration hilft Ihnen, die erforderlichen allgemeinen Konfigurationsparameter beizubehalten. Sie können eine neue Version Ihrer Startvorlage aus der Basiskonfiguration erstellen, wann immer Sie möchten. Sie können auch die Versionen löschen, die zum Testen Ihrer Anwendung verwendet werden, wenn Sie sie nicht mehr benötigen.

Es wird empfohlen, Startvorlagen zu verwenden, um sicherzustellen, dass Sie auf die neuesten Funktionen und Verbesserungen zugreifen. Nicht alle Amazon EC2 Auto Scaling-Funktionen sind verfügbar, wenn Sie Startkonfigurationen verwenden. Sie können beispielsweise keine Auto-Scaling-Gruppe erstellen, die Spot- und On-Demand-Instances startet oder mehrere Instance-Typen angibt. Sie müssen eine Startvorlage verwenden, um diese Funktionen zu konfigurieren. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.

Mit Startvorlagen können Sie auch neuere Funktionen von Amazon EC2 verwenden. Dazu gehören Systems Manager Manager-Parameter (AMI-ID), die aktuelle Generation der bereitgestellten EBS IOPS-Volumes (io2), EBS-Volume-Tagging, T2 Unlimited-Instances, Kapazitätsreservierungen und Dedicated HostsCapacity Blocks, um nur einige zu nennen.

Beim Erstellen einer Startvorlage sind alle Parameter optional. Wenn eine Startvorlage jedoch kein AMI angibt, können Sie das AMI nicht hinzufügen, wenn Sie Ihre Auto-Scaling-Gruppe erstellen. Wenn Sie ein AMI angeben, aber keinen Instance-Typ haben, können Sie beim Erstellen der Auto-Scaling-Gruppe einen oder mehrere Instance-Typen hinzufügen.

Inhalt

Berechtigungen für die Arbeit mit Startvorlagen

Bei den Verfahren in diesem Abschnitt wird davon ausgegangen, dass Sie bereits über die erforderlichen Berechtigungen zum Erstellen von Startvorlagen verfügen. Informationen darüber, wie ein Administrator Ihnen Berechtigungen erteilt, finden Sie unter Steuern des Zugriffs auf Startvorlagen mit IAM-Berechtigungen im Amazon EC2 EC2-Benutzerhandbuch.

Wenn Sie nicht über ausreichende Berechtigungen für die Verwendung und Erstellung von Ressourcen verfügen, die in einer Startvorlage angegeben sind, erhalten Sie eine Fehlermeldung, dass Sie nicht berechtigt sind, die Startvorlage zu verwenden, wenn Sie versuchen, sie für eine Auto-Scaling-Gruppe zu spezifizieren. Weitere Informationen finden Sie unter Fehlersuche bei Amazon EC2 Auto Scaling: Startvorlagen.

Beispiele für IAM-Richtlinien, mit denen Sie die `RunInstances` API-Operationen `CreateAutoScalingGroup` `UpdateAutoScalingGroup`, und mit einer Startvorlage aufrufen können, finden Sie unter. Support für Startvorlagen

Von Startvorlagen unterstützte API-Operationen

Eine Liste der API-Operationen, die von Startvorlagen unterstützt werden, finden Sie unter Amazon-EC2-Aktionen in der Amazon-EC2-API-Referenz.

Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe

Bevor Sie eine Auto-Scaling-Gruppe mit einer Startvorlage erstellen können, müssen Sie eine Startvorlage erstellen, die die Konfigurationsinformationen zum Starten einer Instance enthält, einschließlich der ID des Amazon Machine Image (AMI).

Verwenden Sie die folgenden Verfahren, um neue Startvorlagen zu erstellen.

Inhalt

> **Important**
>
> Die Parameter der Startvorlage werden nicht vollständig validiert, wenn Sie die Startvorlage erstellen. Wenn Sie falsche Werte für Parameter angeben oder keine unterstützten Parameterkombinationen verwenden, können keine Instances unter Verwendung dieser Startvorlage gestartet werden. Stellen Sie sicher, dass Sie die richtigen Werte für die Parameter angeben, und verwenden Sie unterstützte Parameterkombinationen. Um beispielsweise Instances mit einem Arm-basierten AWS Graviton- oder Graviton2-AMI zu starten, müssen Sie einen Arm-kompatiblen Instance-Typ angeben. Weitere Informationen finden Sie unter Einschränkungen für Startvorlagen im Amazon EC2 EC2-Benutzerhandbuch.

So erstellen Sie eine Startvorlage (Konsole)

In den folgenden Schritten wird beschrieben, wie Sie eine einfache Startvorlage konfigurieren:

- Geben Sie das Amazon Machine Image (AMI) an, von dem aus die Instances gestartet werden sollen.
- Wählen Sie einen Instance-Typ aus, der mit dem von Ihnen angegebenen AMI kompatibel ist.
- Geben Sie das Schlüsselpaar an, das Sie bei der Verbindung mit Instances verwenden möchten, z. B. mithilfe von SSH.
- Fügen Sie eine oder mehrere Sicherheitsgruppen hinzu, um Netzwerkzugriff auf die Instances zuzulassen.
- Geben Sie an, ob jeder Instance zusätzliche Volumes zugeordnet werden sollen.
- Fügen Sie benutzerdefinierte Tags (Schlüssel-Wert-Paare) zu den Instances und Volumes hinzu.

Eine Startvorlage erstellen

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Launch Templates aus.
3. Wählen Sie Startvorlage erstellen. Geben Sie einen Namen und eine Beschreibung für die anfängliche Version der Startvorlage ein.
4. (Optional) Aktivieren Sie das Kontrollkästchen unter Auto Scaling-Anleitung, damit Amazon EC2 eine hilfreiche Anleitung bei der Erstellung einer Vorlage für die Verwendung mit Amazon EC2 Auto Scaling bereitstellt.
5. Füllen Sie unter Launch template contents (Vorlageninhalte starten), jedes erforderliche Feld und alle optionalen Felder aus.

   a. Application and OS Images (Amazon Machine Image) (Anwendungs- und Betriebssystem-Images (Amazon Machine Image)): (Erforderlich) Wählen Sie die ID des AMI für Ihre Instances aus. Sie können alle verfügbaren AMIs durchsuchen oder ein AMI aus der Liste Recents (Kürzlich) oder Quickstart (Schnellstart) auswählen. Wenn das benötigte AMI nicht angezeigt wird, wählen Sie Browse more AMIs (Mehr AMIs durchsuchen), um den vollständigen AMI-Katalog zu durchsuchen.

      Um ein benutzerdefiniertes AMI auszuwählen, müssen Sie zuerst Ihr AMI aus einer benutzerdefinierten Instance erstellen. Weitere Informationen finden Sie unter Create an AMI im Amazon EC2 EC2-Benutzerhandbuch.

   b. Wählen Sie für Instance type (Instance-Type) einen einzelnen Instance-Typ aus, der mit dem von Ihnen angegebenen AMI kompatibel ist.

      Wenn Sie alternativ die attributbasierte Auswahl des Instance-Typs verwenden möchten, wählen Sie Erweitert, Spezifizieren von Instance-Typen-Attributen aus und geben Sie die folgenden Optionen an:

      - Number of vCPUs (Anzahl vCPUs): Geben Sie die minimale und maximale Anzahl der vCPUs für Ihre Rechenanforderungen ein. Um keine Limits anzugeben, geben Sie mindestens 0 ein und lassen Sie das Maximum leer.
      - Amount of memory (MiB) (Speichermenge (MiB)): Geben Sie in MiB den minimalen und maximalen Speicher für Ihre Rechenanforderungen ein. Um keine Limits anzugeben, geben Sie mindestens 0 ein und lassen Sie das Maximum leer.
      - Erweitern Sie Optional instance type attributes (Optionale Instance-Typ-Attribute) und wählen Sie Add attribute (Attribut hinzufügen), um die Arten von Instances,

die zur Erreichung Ihrer gewünschten Kapazität verwendet werden können, weiter einzuschränken. Informationen zu den einzelnen Attributen finden Sie unter InstanceRequirementsAnfrage in der Amazon EC2 API-Referenz.

- Resultierende Instance-Typen: Sie können die Instance-Typen anzeigen, die den angegebenen Rechenanforderungen entsprechen, z.B. vCPUs, Arbeitsspeicher und Speicher.
- Um Instance-Typen auszuschließen, wählen Sie Add Attribut (Attribut hinzufügen). Wählen Sie in der Liste Attribute (Attribut) Excluded instance types (Ausgeschlossene Instance-Typen). Wählen Sie aus der Liste Attributwert die auszuschließenden Instance-Typen aus.

c. Key pair (login) (Schlüsselpaar (Login)): Wählen Sie für Key pair name (Schlüsselpaarname) ein vorhandenes Schlüsselpaar aus oder wählen Sie Create new key pair (Neues Schlüsselpaar erstellen), um ein neues zu erstellen. Weitere Informationen finden Sie unter Amazon EC2 EC2-Schlüsselpaare und Linux-Instances im Amazon EC2 EC2-Benutzerhandbuch.

d. Network settings (Netzwerkeinstellungen): Wählen Sie für Firewall (security groups) (Firewall (Sicherheitsgruppen) eine oder mehrere Sicherheitsgruppen aus, oder lassen Sie sie leer und konfigurieren Sie eine oder mehrere Sicherheitsgruppen als Teil der Netzwerkschnittstelle. Weitere Informationen finden Sie unter Amazon-EC2-Sicherheitsgruppen für Linux-Instances im Amazon-EC2-Benutzerhandbuch.

   Wenn Sie in Ihrer Startvorlage keine Sicherheitsgruppen angeben, verwendet Amazon EC2 die Standardsicherheitsgruppe für die VPC, in der Ihre Auto-Scaling-Gruppe Instances startet. Standardmäßig lässt diese Sicherheitsgruppe eingehenden Datenverkehr von externen Netzwerken nicht zu. Weitere Informationen finden Sie unter Standardsicherheitsgruppen für Ihre VPC im Benutzerhandbuch zu Amazon VPC.

e. Führen Sie eine der folgenden Aktionen aus:

   - So ändern Sie die Standardeinstellungen für die Netzwerkschnittstelle. Sie können beispielsweise die öffentliche IPv4-Adressierungsfunktion aktivieren oder deaktivieren, die die Einstellung für die automatische Zuweisung öffentlicher IPv4-Adressen im Subnetz außer Kraft setzt. Weitere Informationen finden Sie unter So ändern Sie die Standardeinstellungen für die Netzwerkschnittstelle (Konsole).
   - Überspringen Sie diesen Schritt, um die Standardeinstellungen für die Netzwerkschnittstelle beizubehalten.

f. Führen Sie eine der folgenden Aktionen aus:

- Ändern Sie die Speicherkonfiguration. Weitere Informationen finden Sie unter Ändern Sie die Speicherkonfiguration (Konsole).
- Überspringen Sie diesen Schritt, um die Standardspeicherkonfiguration beizubehalten.

g. Geben Sie für Resource Tags (Ressourcen-Tags) die Tags an, indem Sie Schlüssel-Wert-Kombinationen bereitstellen. Wenn Sie Instance-Tags in Ihrer Startvorlage angeben und sich dann dafür entschieden haben, die Tags Ihrer Auto-Scaling-Gruppe an ihre Instances zu übertragen, werden alle Tags zusammengeführt. Wenn derselbe Tag-Schlüssel für einen Tag in Ihrer Startvorlage und einen Tag in Ihrer Auto-Scaling-Gruppe angegeben wird, hat der Tag-Wert aus der Gruppe Vorrang.

6. (Optional) Konfigurieren erweiterter Einstellungen. Beispielsweise können Sie eine IAM-Rolle auswählen, die Ihre Anwendung verwenden kann, wenn sie auf andere AWS -Ressourcen zugreift. Alternativ können Sie die Instance-Benutzerdaten angeben, die zum Ausführen allgemeiner automatisierter Konfigurationsaufgaben nach dem Start einer Instance verwendet werden können. Weitere Informationen finden Sie unter Erstellen einer Startvorlage mithilfe erweiterter Einstellungen.
7. Wenn Sie bereit sind, die Startvorlage zu erstellen, wählen Sie Create launch template (Startvorlage erstellen) aus.
8. Wählen Sie auf der Bestätigungsseite Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus, um eine Auto-Scaling-Gruppe zu erstellen.

So ändern Sie die Standardeinstellungen für die Netzwerkschnittstelle (Konsole)

Netzwerkschnittstellen bieten eine Konnektivität zu anderen Ressourcen in Ihrer VPC und im Internet. Weitere Informationen finden Sie unter Stellen Sie Netzwerkkonnektivität für Ihre Auto-Scaling-Instances mit Amazon VPC bereit.

In diesem Abschnitt erfahren Sie, wie Sie die Standardeinstellungen für die Netzwerkschnittstelle ändern. Sie können beispielsweise definieren, ob Sie jeder Instance eine öffentliche IPv4-Adresse zuweisen möchten, anstatt standardmäßig die öffentliche IPv4-Adressen im Subnetz automatisch zuweisen zu müssen.

Überlegungen und Einschränkungen

Beachten Sie beim Ändern der Standardeinstellungen für die Netzwerkschnittstelle die folgenden Überlegungen und Einschränkungen:

- Sie müssen die Sicherheitsgruppen als Teil der Netzwerkschnittstelle angeben und nicht im Bereich Security Groups (Sicherheitsgruppen) der Vorlage. Sie können keine Sicherheitsgruppen an beiden Orten festlegen.
- Sie können einer Netzwerkschnittstelle keine sekundären privaten IP-Adressen, auch secondary IP addresses (sekundäre private IP-Adressen) genannt, zuweisen.
- Wenn Sie eine vorhandene Netzwerkschnittstellen-ID angeben, können Sie nur eine Instance starten. Dazu müssen Sie das AWS CLI oder ein SDK verwenden, um die Auto Scaling Scaling-Gruppe zu erstellen. Wenn Sie die Gruppe erstellen, müssen Sie die Availability Zone angeben, jedoch nicht die Subnetz-ID. Außerdem können Sie eine vorhandene Netzwerkschnittstelle nur angeben, wenn sie einen Geräteindex von 0 hat.
- Sie können eine öffentliche IPv4-Adresse nicht automatisch zuweisen, wenn Sie mehr als eine Netzwerkschnittstelle angeben. Sie können auch keine doppelten Geräteindizes über Netzwerkschnittstellen hinweg angeben. Sowohl die primäre als auch die sekundäre Netzwerkschnittstelle befinden sich im selben Subnetz.
- Wenn eine Instance gestartet wird, wird jeder Netzwerkschnittstelle automatisch eine private Adresse zugewiesen. Die Adresse stammt aus dem CIDR-Bereich des Subnetzes, in dem die Instance gestartet wird. Weitere Informationen zum Angeben von CIDR-Blöcken (oder IP-Adressbereichen) für Ihre VPC oder ein Subnetz finden Sie im Amazon-VPC-Benutzerhandbuch.

So ändern Sie die Standardeinstellungen für die Netzwerkschnittstelle

1. Erweitern Sie unter Network settings (Netzwerkeinstellungen) Advanced network configuration (Erweiterte Netzwerkkonfiguration).
2. Wählen Sie Add network interface (Netzwerkschnittstelle hinzufügen) aus, um die primäre Netzwerkschnittstelle zu konfigurieren, und beachten Sie dabei die folgenden Felder:

    a. Device index (Geräteindex): Behalten Sie den Standardwert 0 bei, um Ihre Änderungen auf die primäre Netzwerkschnittstelle (eth0) anzuwenden.

    b. Network interface (Netzwerkschnittstelle): Behalten Sie den Standardwert New interface (Neue Schnittstelle) bei, um Amazon EC2 Auto Scaling automatisch eine neue Netzwerkschnittstelle erstellen zu lassen, wenn eine Instance gestartet wird. Alternativ können Sie eine vorhandene, verfügbare Netzwerkschnittstelle mit einem Geräteindex von 0 auswählen, dies beschränkt Ihre Auto-Scaling-Gruppe jedoch auf eine Instance.

    c. Description (Beschreibung): (Optional) Geben Sie einen beschreibenden Namen ein.

d. Subnet (Subnetz): Behalten Sie die Standardeinstellung Don't include in launch template (Nicht in die Startvorlage einschließen) bei.

   Wenn das AMI ein Subnetz für die Netzwerkschnittstelle angibt, führt dies zu einem Fehler. Es wird empfohlen, Auto Scaling guidance (Anleitung zur automatischen Skalierung) als Problemumgehung zu deaktivieren. Nachdem Sie diese Änderung vorgenommen haben, erhalten Sie keine Fehlermeldung. Unabhängig davon, wo das Subnetz angegeben ist, haben die Subnetzeinstellungen der Auto-Scaling-Gruppe Vorrang und können nicht überschrieben werden.

e. Auto-assign public IP (Auto-zuweisen öffentlicher IP): Legen Sie fest, ob Ihre Netzwerkschnittstelle mit einem Geräteindex von 0 eine öffentliche IPv4-Adresse erhält. Instances in einem Standardsubnetz erhalten standardmäßig eine öffentliche IPv4-Adresse, Instances in einem nicht standardmäßigen Subnetz nicht. Wählen Sie Enable oder Disable aus, um die Standardeinstellungen des Subnetzes zu überschreiben.

f. Security groups (Sicherheitsgruppen): Wählen Sie eine oder mehrere Sicherheitsgruppen für die Netzwerkschnittstelle aus. Jede Sicherheitsgruppe muss für die VPC konfiguriert werden, in der Ihre Auto-Scaling-Gruppe Instances starten wird. Weitere Informationen finden Sie unter Amazon-EC2-Sicherheitsgruppen für Linux-Instances im Amazon-EC2-Benutzerhandbuch.

g. Delete on termination (Beim Beenden löschen): Wählen Sie Yes (Ja), um die Netzwerkschnittstelle zu löschen, wenn die Instance beendet wird, oder wählen Sie No (Nein), um die Netzwerkschnittstelle beizubehalten.

h. Elastic Fabric Adapter: Um Anwendungsfälle für High Performance Computing und Machine Learning zu unterstützen, ändern Sie die Netzwerkschnittstelle in eine Elastic-Fabric-Adapter-Netzwerkschnittstelle. Weitere Informationen finden Sie unter Elastic Fabric Adapter im Amazon EC2 EC2-Benutzerhandbuch.

i. Network card index (Netzwerkkarten-Index): Wählen Sie 0 aus, um die primäre Netzwerkschnittstelle mit einem Geräteindex von 0 an die Netzwerkkarte anzuschließen. Wenn diese Option nicht verfügbar ist, behalten Sie den Standardwert Don't include in launch template (Nicht in die Startvorlage einschließen) bei. Das Anschließen der Netzwerkschnittstelle an eine bestimmte Netzwerkkarte ist nur für unterstützte Instance-Typen verfügbar. Weitere Informationen finden Sie unter Netzwerkkarten im Amazon EC2 EC2-Benutzerhandbuch.

j. ENA Express: Wählen Sie für Instance-Typen, die ENA Express unterstützen, Enable, um ENA Express zu aktivieren, oder Disable, um ENA Express zu deaktivieren. Weitere

Informationen finden Sie unter Verbessern der Netzwerkleistung mit ENA Express auf Linux-Instances im Amazon EC2 EC2-Benutzerhandbuch.

k. ENA Express UDP: Wenn Sie ENA Express aktivieren, können Sie es optional für UDP-Traffic verwenden. Wählen Sie Aktivieren, um ENA Express UDP zu aktivieren, oder Deaktivieren, um es zu deaktivieren.

3. Wählen Sie zum Hinzufügen einer sekundären Netzwerkschnittstelle Netzwerkschnittstelle hinzufügen aus.

Ändern Sie die Speicherkonfiguration (Konsole)

Sie können die Speicherkonfiguration für Instances ändern, die von einem Amazon-EBS-gestützten AMI oder einem Instance-Speicher-gestützten AMI gestartet werden. Sie können auch zusätzliche EBS-Volumes angeben, die an die Instances angefügt werden sollen. Das AMI enthält ein oder mehrere Speicher-Volumes, einschließlich des Root-Volumes (Volume 1 (AMI Root)).

So ändern Sie die Speicherkonfiguration

1. In :Configure storage (Speicher konfigurieren), ändern Sie die Größe oder den Typ des Volumes.

   Wenn der Wert, den Sie für die Volumegröße angeben, außerhalb der Grenzwerte des Volume-Typs liegt oder kleiner als die Snapshotgröße ist, wird eine Fehlermeldung angezeigt. Um Ihnen bei der Behebung des Problems zu helfen, gibt diese Nachricht den minimalen oder maximalen Wert an, den das Feld akzeptieren kann.

   Es werden nur Volumes angezeigt, die einem Amazon-EBS-gestützten AMI zugeordnet sind. Um Informationen zur Speicherkonfiguration für eine Instance anzuzeigen, die von einem Instance-Speicher-gestützten AMI gestartet wurde, wählen Sie Show details (Details anzeigen) aus dem Abschnitt Instance store volumes (Instance-Speicher-Volumes) aus.

   Um alle EBS-Volume-Parameter anzugeben, wechseln Sie zur Advanced(Erweitert)-Ansicht in der rechten oberen Ecke.

2. Erweitern Sie für erweiterte Optionen das Volume, das Sie ändern möchten, und konfigurieren Sie das Volume wie folgt:

   a. Storage type (Speichertyp): Der Typ des Volumes (EBS oder flüchtig), das Ihrer Instance zugeordnet werden soll. Der Volume-Typ Instance-Speicher (flüchtig) ist nur verfügbar, wenn Sie einen Instance-Typ auswählen, der ihn unterstützt. Weitere Informationen finden

Sie unter Amazon EBS-Volumes im Amazon EBS-Benutzerhandbuch und Amazon EC2 EC2-Instance-Speicher im Amazon EC2 EC2-Benutzerhandbuch.

b. Device name (Gerätename): Wählen Sie aus der Liste verfügbarer Gerätenamen für das Volume einen Eintrag aus.

c. Snapshot: Wählen Sie den Snapshot aus, von dem das Volume erstellt werden soll. Sie können nach verfügbaren freigegebenen und öffentlichen Snapshots suchen, indem Sie Text in das Feld Snapshot eingeben.

d. Größe (GiB): Sie können für EBS-Volumes eine Speichergröße angeben. Wenn Sie ein AMI und eine Instance ausgewählt haben, die im kostenlosen Kontingent enthalten sind, dürfen Sie den Grenzwert von 30 GiB Gesamtspeicher nicht überschreiten, um innerhalb des kostenlosen Kontingents zu bleiben. Weitere Informationen finden Sie unter Einschränkungen für die Größe und Konfiguration eines EBS-Volumes im Amazon EBS-Benutzerhandbuch.

e. Volume type (Volume-Typ): Wählen Sie für EBS-Volumes den Volume-Typ aus. Weitere Informationen finden Sie unter Amazon EBS-Volumetypen im Amazon EBS-Benutzerhandbuch.

f. IOPS: Wenn Sie einen Volume des Typs Bereitgestellte IOPS-SSD (`io1` und `io2`) oder Universelle SSD (`gp3`) ausgewählt haben, können Sie die Anzahl der I/O-Operationen pro Sekunde (IOPS) eingeben, die das Volume unterstützen kann. Dies ist für io1-, io2- und gp3-Volumes erforderlich. Wird bei gp2-, st1-, sc1- oder Standard-Volumes nicht unterstützt.

g. Delete on termination (Bei Beendigung löschen): Wählen Sie für EBS-Volumes Yes (Ja), um das Volume zu löschen, wenn die Instance beendet wird, oder wählen Sie No (Nein), um das Volume beizubehalten.

h. Encrypted (Verschlüsselt): Wenn der Instance-Typ die EBS-Verschlüsselung unterstützt, können Sie Ja auswählen, um die Verschlüsselung für das Volume zu aktivieren. Wenn Sie für diese Region die standardmäßige Verschlüsselung aktiviert haben, wird der Standard-CMK für Sie ausgewählt. Weitere Informationen finden Sie unter Amazon EBS-Verschlüsselung und Standardverschlüsselung aktivieren im Amazon EBS-Benutzerhandbuch.

   Der Standardeffekt bei diesem Parameter hängt von der Wahl der Volume-Quelle ab, wie in der folgenden Tabelle beschrieben. In jedem Fall müssen Sie über die Erlaubnis verfügen, die angegebenen Daten zu verwenden. AWS KMS key

Verschlüsselungsergebnisse

Wenn für den Parameter **`Encrypted`** Folgendes festgelegt ist ...	Und wenn die Quelle des Volumes Folgendes ist ...	Dann ist der Standardverschlüsselungsstatus ...	Hinweise
Nein	Neues (leeres) Volume	Unverschlüsselt*	N/A
	Unverschlüsselter eigener Snapshot	Unverschlüsselt*	
	Verschlüsselter eigener Snapshot	Verschlüsselt mit demselben Schlüssel	
	Unverschlüsselter Snapshot, der mit Ihnen geteilt wird	Unverschlüsselt*	
	Verschlüsselter Snapshot, der mit Ihnen geteilt wird	Verschlüsselt durch standardmäßigen KMS-Schlüssel	
Ja	Neues Volume	Verschlüsselt durch standardmäßigen KMS-Schlüssel	Um einen nicht standardmäßigen KMS-Schlüssel zu verwenden, geben Sie einen Wert für den KMS key(KMS-Schlüssel)-Parameter ein.
	Unverschlüsselter eigener Snapshot	Verschlüsselt durch standardmäßigen KMS-Schlüssel	
	Verschlüsselter eigener Snapshot	Verschlüsselt mit demselben Schlüssel	
	Unverschlüsselter Snapshot, der mit Ihnen geteilt wird	Verschlüsselt durch standardmäßigen KMS-Schlüssel	

Wenn für den Parameter **`Encrypted`** Folgendes festgelegt ist ...	Und wenn die Quelle des Volumes Folgendes ist ...	Dann ist der Standardverschlüsselungsstatus ...	Hinweise
	Verschlüsselter Snapshot, der mit Ihnen geteilt wird	Verschlüsselt durch standardmäßigen KMS-Schlüssel	

* Wenn die standardmäßige Verschlüsselung aktiviert ist, werden alle neu erstellten Volumes (unabhängig davon, ob der Encrypted(Verschlüsselt)-Parameter auf Yes (Ja) gesetzt ist) mit dem standardmäßigen KMS-Schlüssel verschlüsselt. Wenn Sie sowohl den Encrypted(Verschlüsselt)- als auch den Key(Schlüssel)-Parameter festlegen, können Sie einen nicht standardmäßigen KMS-Schlüssel angeben.

i. KMS key (KMS-Schlüssel): Wenn Sie Yes (Ja) für Encrypted (Verschlüsselt) auswählen, müssen Sie einen kundenverwalteten Schlüssel zum Verschlüsseln des Volumes auswählen. Wenn die für diese Region die standardmäßige Verschlüsselung aktiviert haben, wird der Standard-CMK für Sie ausgewählt. Sie können einen anderen Schlüssel auswählen oder den ARN eines beliebigen kundenverwalteten Schlüssels angeben, den Sie zuvor mit dem AWS Key Management Service erstellt haben.

3. Um zusätzliche Volumes anzugeben, die an die von dieser Startvorlage gestarteten Instances angehängt werden sollen, wählen Sie Add new volume (Neues Volume hinzufügen) aus.

Erstellen Sie eine Startvorlage anhand einer vorhandenen Instance (Konsole)

So erstellen Sie eine Startvorlage anhand einer vorhandenen Instance

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances (Instances) die Option Instances (Instances) aus.
3. Wählen Sie die Instance und anschließend Aktionen, Image und Vorlagen, Eine Vorlage aus einer Instance erstellen aus.

4. Geben Sie einen Namen und eine Beschreibung ein.
5. Unter Auto-Scaling-Anleitung aktivieren Sie das Kontrollkästchen.
6. Passen Sie alle Einstellungen wie erforderlich an und wählen Sie Startvorlage erstellen aus.
7. Wählen Sie auf der Bestätigungsseite Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus, um eine Auto-Scaling-Gruppe zu erstellen.

Zugehörige Ressourcen

Wir stellen einige JSON- und YAML-Vorlagenausschnitte zur Verfügung, anhand derer Sie verstehen können, wie Sie Startvorlagen in Ihren AWS CloudFormation Stack-Vorlagen deklarieren. Weitere Informationen finden Sie in den AWS CloudFormation Abschnitten AWS::EC2::LaunchTemplateund Erstellen von Startvorlagen mit Hilfe des AWS CloudFormation Benutzerhandbuchs.

Weitere Informationen zu Startvorlagen finden Sie unter Starten einer Instance aus einer Startvorlage im Amazon EC2 EC2-Benutzerhandbuch.

Einschränkungen

- Sie können zwar ein Subnetz in einer Startvorlage spezifizieren, aber das ist nicht notwendig, wenn Sie die Startvorlage nur zum Erstellen von Auto-Scaling-Gruppen verwenden. Sie können das Subnetz für eine Auto-Scaling-Gruppe nicht durch Spezifizierung des Subnetzes in einer Startvorlage festlegen. Die Subnetze für die Auto-Scaling-Gruppe werden aus der eigenen Ressourcendefinition der Auto-Scaling-Gruppe übernommen.
- Weitere Einschränkungen für benutzerdefinierte Netzwerkschnittstellen finden Sie unter So ändern Sie die Standardeinstellungen für die Netzwerkschnittstelle (Konsole).

Erstellen einer Startvorlage mithilfe erweiterter Einstellungen

In diesem Thema wird beschrieben, wie Sie eine Startvorlage mit erweiterten Einstellungen aus dem AWS Management Console erstellen.

Um eine Startvorlage mit erweiterten Einstellungen zu erstellen

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Launch Templates und dann Create Launch Template aus.

3. Konfigurieren Sie Ihre Startvorlage wie in den folgenden Themen beschrieben:

   - Erforderliche Einstellungen
   - Erweiterte Einstellungen

4. Wählen Sie Startvorlage erstellen.

Erforderliche Einstellungen

Wenn Sie eine Startvorlage erstellen, müssen Sie die folgenden erforderlichen Einstellungen angeben.

Name der Startvorlage

Geben Sie einen eindeutigen Namen ein, der die Startvorlage beschreibt.

Anwendungs- und Betriebssystem-Images (Amazon Machine Image)

Wählen Sie das Amazon Machine Image (AMI) aus, das Sie verwenden möchten. Sie können entweder nach dem AMI suchen oder nach dem AMI suchen, das Sie verwenden möchten. Wählen Sie für eine optimale Skalierungseffizienz ein benutzerdefiniertes AMI, das vollständig so konfiguriert ist, dass es eine Instance mit Ihrem Anwendungscode startet und beim Start nur wenige Änderungen erfordert.

Instance-Typ

Wählen Sie einen Instance-Typ, der mit Ihrem AMI kompatibel ist. Sie können das Hinzufügen eines Instance-Typs zu Ihrer Startvorlage überspringen, wenn Sie mehrere Instance-Typen verwenden möchten, die in die eigene Ressourcendefinition der Auto Scaling Scaling-Gruppe eingebettet sind. Ein Instanztyp ist nur erforderlich, wenn Sie nicht vorhaben, eine gemischte Instanzgruppe zu erstellen.

Erweiterte Einstellungen

Die erweiterten Einstellungen sind optional. Wenn Sie keine erweiterten Einstellungen konfigurieren, werden die spezifischen Funktionen nicht zu Ihren Instances hinzugefügt.

Erweitern Sie den Abschnitt Erweiterte Details, um die erweiterten Einstellungen anzuzeigen. In den folgenden Abschnitten werden die nützlichsten erweiterten Einstellungen beschrieben, auf die Sie sich bei der Erstellung einer Startvorlage für eine Auto Scaling Scaling-Gruppe konzentrieren sollten. Weitere Informationen finden Sie unter Erweiterte Details im Amazon EC2 EC2-Benutzerhandbuch.

IAM-Instance-Profil

Das Instance-Profil enthält die IAM-Rolle, die Sie verwenden möchten. Wenn Ihre Auto Scaling Scaling-Gruppe eine EC2-Instance startet, werden die in der zugehörigen IAM-Rolle definierten Berechtigungen Anwendungen gewährt, die auf der Instance ausgeführt werden. Weitere Informationen finden Sie unter IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden.

Termination protection

Wenn diese Funktion aktiviert ist, verhindert sie, dass Benutzer eine Instance mithilfe der Amazon EC2 EC2-Konsole, CLI-Befehlen und API-Operationen beenden. Der Kündigungsschutz bietet zusätzlichen Schutz vor versehentlicher Kündigung. Es verhindert nicht, dass Amazon EC2 Auto Scaling eine Instance beendet. Informationen zur Steuerung, welche Instances Amazon EC2 Auto Scaling beenden kann, finden Sie unterInstance-Abskalierungsschutz verwenden.

Detaillierte Überwachung CloudWatch

Sie können eine detaillierte Überwachung für Ihre EC2-Instances aktivieren, damit sie in Intervallen von 1 Minute Metrikdaten CloudWatch an Amazon senden können. Standardmäßig senden EC2-Instances Metrikdaten in Intervallen von 5 Minuten CloudWatch an. Es fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Überwachung für Auto-Scaling-Instances konfigurieren.

Kreditspezifikation

Amazon EC2 bietet Burstable-Performance-Instances wie T2, T3 und T3a, die es Anwendungen ermöglichen, bei Bedarf die CPU-Basisleistung zu überschreiten. Standardmäßig können diese Instances für eine begrenzte Zeit ausgelastet werden, bevor ihre CPU-Auslastung gedrosselt wird. Sie können optional den Modus „Unlimited“ aktivieren, sodass die Instances so lange wie nötig über den Basiswert hinaus übersteigen können. Auf diese Weise können Anwendungen bei Bedarf eine hohe CPU-Leistung aufrechterhalten. Es können zusätzliche Gebühren anfallen. Weitere Informationen finden Sie unter Verwenden einer Auto Scaling Scaling-Gruppe zum Starten einer Burstable-Performance-Instance als Unlimited im Amazon EC2 EC2-Benutzerhandbuch.

Name einer Platzierungsgruppe

Sie können eine Platzierungsgruppe angeben und mithilfe einer Cluster- oder Partitionsstrategie beeinflussen, wie sich Ihre Instances physisch im AWS Rechenzentrum befinden. Für kleine Auto Scaling Scaling-Gruppen können Sie auch die Spread-Strategie verwenden. Weitere Informationen finden Sie unter Placement-Gruppen im Amazon EC2 EC2-Benutzerhandbuch.

Bei der Verwendung von Platzierungsgruppen mit Auto Scaling Scaling-Gruppen sind einige Überlegungen zu beachten:

- Wenn eine Platzierungsgruppe sowohl in der Startvorlage als auch in der Auto Scaling Scaling-Gruppe angegeben ist, hat die Platzierungsgruppe für die Auto Scaling Scaling-Gruppe Vorrang. Nachdem die Gruppe erstellt wurde, kann die in den Auto Scaling-Gruppeneinstellungen angegebene Platzierungsgruppe nicht geändert werden.
- Seien Sie vorsichtig AWS CloudFormation, wenn Sie in der Startvorlage eine Platzierungsgruppe definieren. Amazon EC2 Auto Scaling startet Instances in der angegebenen Platzierungsgruppe. Empfangen CloudFormation Sie jedoch keine Signale von diesen Instances, wenn Sie eine UpdatePolicymit Ihrer Auto Scaling Scaling-Gruppe verwenden (obwohl sich dies in future ändern könnte).

Kaufoption

Sie können Spot-Instances anfordern wählen, um Spot-Instances zum Spot-Preis, begrenzt auf den On-Demand-Preis, anzufordern, und „Anpassen“ wählen, um die Standardeinstellungen für Spot-Instances zu ändern. Bei einer Auto-Scaling-Gruppe müssen Sie eine einmalige Anforderung ohne Enddatum angeben (Standardeinstellung). Weitere Informationen finden Sie unter Spot-Instances für fehlertolerante und flexible Anwendungen anfordern. Diese Einstellung kann unter besonderen Umständen nützlich sein, aber im Allgemeinen ist es am besten, sie nicht festzulegen und stattdessen eine gemischte Instances-Gruppe zu erstellen. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.

Wenn Sie in Ihrer Startvorlage eine Spot Instance-Anfrage angeben, können Sie keine gemischte Instances-Gruppe erstellen. Wenn Sie versuchen, eine Startvorlage zu verwenden, die Spot Instances mit einer gemischten Instance-Gruppe anfordert, erhalten Sie die folgende Fehlermeldung: `Incompatible launch template: You cannot use a launch template that is set to request Spot Instances (InstanceMarketOptions) when you configure an Auto Scaling group with a mixed instances policy. Add a different launch template to the group and try again.`

Capacity Reservation

Kapazitätsreservierungen ermöglichen es Ihnen, Kapazität für Ihre Amazon EC2 EC2-Instances in einer bestimmten Availability Zone für einen beliebigen Zeitraum zu reservieren. Weitere Informationen finden Sie unter Kapazitätsreservierungen auf Abruf im Amazon EC2 EC2-Benutzerhandbuch.

Sie können wählen, ob Sie Instances starten möchten in:

- jede offene Kapazitätsreservierung (Offen)
- eine bestimmte Kapazitätsreservierung (Ziel nach ID)
- eine Gruppe von Kapazitätsreservierungen (Ziel nach Gruppe)

Um auf eine bestimmte Kapazitätsreservierung abzielen zu können, muss der Instance-Typ in Ihrer Startvorlage mit dem Instance-Typ der Reservierung übereinstimmen. Wenn Sie Ihre Auto Scaling Scaling-Gruppe erstellen, verwenden Sie dieselbe Availability Zone wie die Kapazitätsreservierung. Je nachdem, AWS-Region was Sie wählen, können Sie sich stattdessen für einen Kapazitätsblock entscheiden. Weitere Informationen finden Sie unter Capacity BlocksFür Machine-Learning-Workloads verwenden.

Informationen zur gezielten Ausrichtung auf eine Gruppe von Kapazitätsreservierungen finden Sie unterVerwenden Sie On-Demand-Kapazitätsreservierungen, um Kapazitäten in bestimmten Availability Zones zu reservieren.. Wenn Sie auf eine Gruppe von Kapazitätsreservierungen abzielen, können Sie die Kapazität auf mehrere Availability Zones verteilen, um die Ausfallsicherheit zu verbessern.

Tenancy

Amazon EC2 bietet drei Optionen für die Wartung Ihrer EC2-Instances:

- Gemeinsam genutzt (gemeinsam genutzt) — Mehrere AWS-Konten können sich dieselbe physische Hardware teilen. Dies ist die Standard-Tenancy-Option beim Starten einer Instance.
- Dedizierte Instances (Dedicated) — Ihre Instance wird auf Single-Tenant-Hardware ausgeführt. Kein anderer AWS Kunde teilt sich denselben physischen Server. Weitere Informationen finden Sie unter Dedicated Instances im Amazon EC2-Benutzerhandbuch.
- Dedicated Hosts (Dedicated Host) — Die Instance wird auf einem physischen Server ausgeführt, der für Sie reserviert ist. Die Verwendung von Dedicated Hosts macht es einfacher, Ihre eigenen Lizenzen (BYOL) mit speziellen Hardwareanforderungen für EC2 zu verwenden und Compliance-Anwendungsfälle zu erfüllen. Wenn Sie diese Option wählen, müssen Sie eine Host-Ressourcengruppe für die Tenancy-Host-Ressourcengruppe angeben. Weitere Informationen finden Sie unter Dedicated Hosts im Amazon EC2 EC2-Benutzerhandbuch.

Support für Dedicated Hosts ist nur verfügbar, wenn Sie eine Host-Ressourcengruppe angeben. Sie können keinen bestimmten Host als Ziel festlegen oder eine Host-Platzierungsaffinität verwenden.

- Wenn Sie versuchen, eine Startvorlage zu verwenden, die eine Host-ID angibt, erhalten Sie die folgende Fehlermeldung: `Incompatible launch template: Tenancy host ID is not supported for Auto Scaling.`

- Wenn Sie versuchen, eine Startvorlage zu verwenden, die die Hostplatzierungsaffinität angibt, wird die folgende Fehlermeldung angezeigt: `Incompatible launch template: Auto Scaling does not support host placement affinity.`

Hostressourcengruppe „Tenancy“

Mit AWS License Manager können Sie Ihre eigenen Lizenzen verwenden AWS und diese zentral verwalten. Eine Host-Ressourcengruppe ist eine Gruppe von Dedicated Hosts, die mit einer bestimmten License Manager Manager-Lizenzkonfiguration verknüpft sind. Host-Ressourcengruppen ermöglichen es Ihnen, EC2-Instances einfach auf Dedicated Hosts zu starten, die Ihren Softwarelizenzanforderungen entsprechen. Sie müssen Dedicated Hosts nicht vorab manuell zuweisen. Sie werden bei Bedarf automatisch erstellt. Beachten Sie, dass, wenn Sie ein AMI mit einer Lizenzkonfiguration verknüpfen, dieses AMI jeweils nur einer Host-Ressourcengruppe zugeordnet werden kann. Weitere Informationen finden Sie unter Host-Ressourcengruppen AWS License Manager im License Manager Manager-Benutzerhandbuch.

Lizenzkonfigurationen

Mit dieser Einstellung können Sie eine Lizenzkonfiguration für Ihre Instances angeben, ohne deren Tenancy auf Dedicated Hosts zu beschränken. Die Lizenzkonfiguration verfolgt die auf den Instances bereitgestellten Softwarelizenzen, sodass Sie Ihre Lizenznutzung und die Einhaltung der Vorschriften überwachen können. Weitere Informationen finden Sie unter Erstellen einer selbstverwalteten Lizenz im License Manager Manager-Benutzerhandbuch.

Auf Metadaten kann zugegriffen werden

Sie können wählen, ob Sie den Zugriff auf den HTTP-Endpunkt des Instanz-Metadatendienstes aktivieren oder deaktivieren möchten. Standardmäßig ist der HTTP-Endpunkt aktiviert. Wenn Sie den Endpunkt deaktivieren, wird der Zugriff auf Ihre Instance-Metadaten deaktiviert. Sie können die Bedingung angeben, dass IMDSv2 nur erforderlich ist, wenn der HTTP-Endpunkt aktiviert ist. Weitere Informationen finden Sie unter Konfiguration der Instance-Metadatenoptionen im Amazon EC2 EC2-Benutzerhandbuch.

Version der Metadaten

Sie können sich dafür entscheiden, die Verwendung von Instance Metadata Service Version 2 (IMDSv2) vorzuschreiben, wenn Sie Instanz-Metadaten anfordern. Wenn Sie keinen Wert angeben, unterstützt standardmäßig IMDSv1 und IMDSv2. Weitere Informationen finden Sie unter Konfiguration der Instance-Metadatenoptionen im Amazon EC2 EC2-Benutzerhandbuch.

Antwort-Hop-Limit für Metadatentok

Sie können die zulässige Anzahl von Netzwerk-Hops für das Metadaten-Token festlegen. Wenn Sie keinen Wert angeben, wird der Standard auf 1 festgelegt. Weitere Informationen finden Sie unter Konfiguration der Instance-Metadatenoptionen im Amazon EC2 EC2-Benutzerhandbuch.

Benutzerdaten

Sie können Ihre Instances beim Start anpassen und die Konfiguration abschließen, indem Sie Shell-Skripte oder Cloud-Init-Direktiven als Benutzerdaten angeben. Die Benutzerdaten werden beim ersten Start der Instanz ausgeführt, sodass Sie Anwendungen, Abhängigkeiten oder Anpassungen beim Start automatisch installieren können. Weitere Informationen finden Sie unter Befehle auf Ihrer Linux-Instance beim Start ausführen im Amazon EC2 EC2-Benutzerhandbuch.

Wenn Sie umfangreiche Downloads oder komplexe Skripts haben, verlängert dies die Zeit, die benötigt wird, bis die Instance einsatzbereit ist. In diesem Fall müssen Sie möglicherweise einen Lifecycle-Hook konfigurieren, um zu verhindern, dass eine Instanz den `InService` Status erreicht, bis sie vollständig bereitgestellt ist. Weitere Informationen zum Hinzufügen eines Lifecycle-Hooks zu Ihrer Auto Scaling Scaling-Gruppe finden Sie unterLebenszyklus-Hooks bei Amazon EC2 Auto Scaling.

Spot-Instances für fehlertolerante und flexible Anwendungen anfordern

In Ihrer Startvorlage können Sie optional Spot-Instances ohne Enddatum oder Dauer anfordern. Amazon EC2-Spot-Instances sind freie Kapazitäten mit hohen Rabatten im Vergleich zum EC2 On-Demand-Preis verfügbar. Spot-Instances sind eine kostengünstige Wahl, sofern Sie bei der Ausführung Ihrer Anwendungen zeitlich flexibel sind und Unterbrechungen verschmerzen können. Weitere Informationen zum Erstellen einer Startvorlage, die Spot-Instanzen anfordert, finden Sie unter Erstellen einer Startvorlage mithilfe erweiterter Einstellungen.

> **Important**
>
> Normalerweise werden Spot-Instances zur Ergänzung von On-Demand-Instances verwendet In diesem Szenario können Sie die gleichen Einstellungen, die auch für den Start von Spot-Instances verwendet werden, als Teil der Einstellungen Ihrer Auto-Scaling-Gruppe festlegen. Wenn Sie die Einstellungen als Teil der Auto-Scaling-Gruppe angeben, können Sie Spot-Instances erst nach dem Start einer bestimmten Anzahl von On-Demand-Instances starten und dann eine Kombination aus On-Demand-Instances und Spot-Instances starten, während

die Gruppe skaliert wird. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.

In diesem Thema wird beschrieben, wie Sie nur Spot-Instances in Ihrer Auto-Scaling-Gruppe starten können, indem Sie die Einstellungen in einer Startvorlage und nicht in der Auto-Scaling-Gruppe selbst festlegen. Die Informationen in diesem Thema gelten auch für Auto-caling-Gruppen, die Spot-Instances mit einer Startkonfiguration anfordern. Der Unterschied besteht darin, dass für eine Startkonfiguration ein Höchstpreis erforderlich ist, bei Startvorlagen ist der Höchstpreis jedoch optional.

Wenn Sie eine Startvorlage erstellen, um nur Spot-Instanzen zu starten, sollten Sie die folgenden Punkte beachten:

- Spot-Preis. Sie zahlen nur den aktuellen Spot-Preis für die Spot-Instances, die Sie starten. Dieser Preis ändert sich im Laufe der Zeit langsam, basierend auf den langfristigen Trends bei Angebot und Nachfrage. Weitere Informationen finden Sie unter Spot-Instances und Preise und Einsparungen im Amazon EC2 EC2-Benutzerhandbuch.
- Festlegen des Höchstpreises. Sie können optional einen Höchstpreis pro Stunde für Spot-Instances in Ihre Startvorlage aufnehmen. Wenn Ihr Maximalpreis den aktuellen Spot-Preis übersteigt, erfüllt der Amazon-EC2-Spot-Service Ihre Anfrage sofort, sofern Kapazität verfügbar ist. Wenn der Preis für Spot-Instances Ihren Maximalpreis für eine laufende Instance in Ihrer Auto-Scaling-Gruppe übersteigt, wird Ihre Instance beendet.

  Warning

  Ihre Anwendung läuft möglicherweise nicht, wenn Sie keine Spot-Instances erhalten, z. B. wenn Ihr Höchstpreis zu niedrig ist. Um so lange wie möglich von den verfügbaren Spot-Instances zu profitieren, legen Sie Ihren Maximalpreis nahe dem On-Demand-Preis fest.

- Ausgleichen zwischen den Availability Zones. Wenn Sie mehrere Availability Zones angeben, verteilt Amazon EC2 Auto Scaling die Spot-Anfragen über die angegebenen Zonen. Ist Ihr Höchstpreis in einer Availability Zone zu niedrig, um Anfragen zu erfüllen, prüft Amazon-EC2-Auto-Scaling, ob Anfragen in anderen Availability Zones erfolgreich waren. Ist dies der Fall, beendet Amazon EC2 Auto Scaling die nicht erfolgreichen Anfragen und verteilt sie auf die Availability Zones, deren Anfragen erfolgreich waren. Fallen die Preise in einer Availability Zone ohne erfolgreiche Anfragen so weit, dass künftige Anfragen erfolgreich sind, gleicht Amazon EC2 Auto Scaling die Kapazitäten der Availability Zones wieder aus.

- Spot-Instance-Beendigung. Spot-Instances können jederzeit gekündigt werden. Der Amazon-EC2-Spot-Service kann Spot-Instances in Ihrer Auto-Scaling-Gruppe beenden, wenn sich die Verfügbarkeit von Spot-Instances oder der Preis für diese ändert. Bei der Skalierung oder bei der Durchführung von Gesundheitsprüfungen kann Amazon EC2 Auto Scaling Spot-Instances auf die gleiche Weise beenden wie On-Demand-Instances. Wenn eine Instance beendet wird, wird jeglicher Speicher gelöscht.
- Behalten Sie Ihre gewünschte Kapazität bei. Wenn eine Spot-Instance beendet wird, versucht Amazon EC2 Auto Scaling, eine andere Spot-Instance zu starten, um die gewünschte Kapazität für die Gruppe aufrechtzuerhalten. Wenn der aktuelle Spot-Preis unter Ihrem Höchstpreis liegt, wird eine Spot-Instance gestartet. Wenn die Anfrage nach einer Spot-Instance erfolglos ist, versucht sie es weiter.
- Ändern des Höchstpreises. Um Ihren Höchstpreis zu ändern, erstellen Sie eine neue Startvorlage oder aktualisieren Sie eine vorhandene Startvorlage mit dem neuen Höchstpreis und verknüpfen Sie sie dann mit Ihrer Auto-Scaling-Gruppe. Die bestehenden Spot-Instances laufen weiter, solange der in der für diese Instances verwendeten Startvorlage angegebene Höchstpreis höher ist als der aktuelle Spot-Preis. Wenn Sie keinen Höchstpreis festgelegt haben, ist der Standardhöchstpreis der Preis auf Abruf.

Capacity BlocksFür Machine-Learning-Workloads verwenden

Capacity Blockshelfen Ihnen dabei, stark nachgefragte GPU-Instances zu einem future Zeitpunkt zu reservieren, um Ihre kurzfristigen Machine-Learning-Workloads (ML) zu unterstützen.

Einen Überblick über Capacity Blocks und wie sie funktionieren, finden Sie unter Capacity Blocksfür ML im Amazon EC2 EC2-Benutzerhandbuch.

Um mit der Nutzung zu beginnenCapacity Blocks, erstellen Sie eine Kapazitätsreservierung in einer bestimmten Availability Zone. Capacity Blockswerden als `targeted` Kapazitätsreservierungen in einer einzigen Availability Zone bereitgestellt. Wenn Sie Ihre Startvorlage erstellen, geben Sie die Reservierungs-ID und den Instanztyp des Kapazitätsblocks an. Aktualisieren Sie dann Ihre Auto Scaling Scaling-Gruppe so, dass sie die von Ihnen erstellte Startvorlage und die Availability Zone des Capacity Blocks verwendet. Wenn Ihre Capacity Block-Reservierung beginnt, verwenden Sie die geplante Skalierung, um dieselbe Anzahl von Instances wie Ihre Capacity Block-Reservierung zu starten.

> **Important**
>
> Capacity Blockssind nur für bestimmte Amazon EC2 EC2-Instance-Typen und AWS-Regionen verfügbar. Weitere Informationen finden Sie unter Voraussetzungen im Amazon EC2 EC2-Benutzerhandbuch.

Inhalt

Betriebliche Richtlinien

Nachfolgend finden Sie grundlegende Richtlinien, die Sie bei der Verwendung eines Kapazitätsblocks mit einer Auto-Scaling-Gruppe beachten sollten.

- Skalieren Sie Ihre Auto-Scaling-Gruppe mehr als 30 Minuten vor der Endzeit der Kapazitätsblockreservierung auf Null herunter. Amazon EC2 beendet alle Instances, die noch in Betrieb sind, 30 Minuten vor dem Ende des Kapazitätsblocks.
- Wir empfehlen Ihnen, die geplante Skalierung zu verwenden, um zu den entsprechenden Reservierungszeiten die horizontale Skalierung (Hinzufügen von Instances) und die Skalierung (Instances entfernen) durchzuführen. Weitere Informationen finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling.
- Fügen Sie bei Bedarf Lebenszyklus-Hooks hinzu, um Ihre Anwendung beim Skalieren innerhalb der Instances ordnungsgemäß herunterzufahren. Lassen Sie genügend Zeit, bis die Lebenszyklus-Aktion abgeschlossen ist, bevor Amazon EC2 beginnt, Ihre Instances 30 Minuten vor dem Ende der Kapazitätsblockreservierung zwangsweise zu beenden. Weitere Informationen finden Sie unter Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling.
- Stellen Sie sicher, dass die Auto-Scaling-Gruppe für die gesamte Dauer der Reservierung auf die richtige Version der Startvorlage verweist. Wir empfehlen, auf eine bestimmte Version der Startvorlage statt auf die Version `$Default` oder `$Latest` zu verweisen.

> **Note**
>
> Wenn Sie eine Capacity Block-Instance bis zum Ende der Reservierung laufen lassen und Amazon EC2 sie zurückfordert, geben die Skalierungsaktivitäten für Ihre Auto Scaling Scaling-Gruppe an, dass sie "`taken out of service in response to an EC2 health check that indicated it had been terminated or stopped`„war, obwohl sie am Ende des Kapazitätsblocks absichtlich zurückgefordert wurde. In ähnlicher Weise versucht Amazon EC2 Auto Scaling, die Instance auf dieselbe Weise zu ersetzen, wie es bei jeder Instance der Fall ist, die eine Zustandsprüfung nicht besteht. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Geben Sie in Ihrer Startvorlage einen Kapazitätsblock an

Verwenden Sie eine der folgenden Methoden, um eine Startvorlage zu erstellen, die auf einen bestimmten Kapazitätsblock für Ihre Auto Scaling Scaling-Gruppe abzielt:

Console

Angabe eines Kapazitätsblocks in Ihrer Startvorlage (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen Navigationsleiste den Ort aus, AWS-Region an dem Sie Ihren Kapazitätsblock erstellt haben.
3. Wählen Sie im Navigationsbereich unter Instances die Option Launch Templates aus.
4. Wählen Sie Startvorlage erstellen und erstellen Sie die Startvorlage. Schließen Sie bei Bedarf die ID des Amazon Machine Image (AMI), den Instance-Typ und alle anderen Startvorlagen ein.
5. Erweitern Sie den Abschnitt Erweiterte Details, um die erweiterten Einstellungen anzuzeigen.
6. Wählen Sie als Kaufoption Kapazitätsblöcke aus.
7. Wählen Sie für Kapazitätsreservierung die Option Ziel nach ID und dann für Kapazitätsreservierung – Ziel nach ID die Kapazitätsreservierungs-ID eines vorhandenen Kapazitätsblocks aus.
8. Klicken Sie danach auf Startvorlage erstellen.

Hilfe zum Erstellen einer Auto Scaling Scaling-Gruppe mit einer Startvorlage finden Sie unterErstellen einer Auto-Scaling-Gruppe mithilfe einer Startvorlage.

AWS CLI

Angabe eines Kapazitätsblocks in Ihrer Startvorlage (AWS CLI)

Verwenden Sie den folgenden Befehl create-launch-template, um eine Startvorlage zu erstellen, die eine vorhandene Reservierungs-ID eines Kapazitätsblocks angibt. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

```
aws ec2 create-launch-template --launch-template-name my-template-for-capacity-block
  \
  --version-description AutoScalingVersion1 --region us-east-2 \
  --launch-template-data file://config.json
```

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

Inhalt von `config.json`.

```
{
    "ImageId": "ami-04d5cc9b88example",
    "InstanceType": "p4d.24xlarge",
    "SecurityGroupIds": [
        "sg-903004f88example"
    ],
    "KeyName": "MyKeyPair",
    "InstanceMarketOptions": {
        "MarketType": "capacity-block"
    },
    "CapacityReservationSpecification": {
        "CapacityReservationTarget": {
            "CapacityReservationId": "cr-02168da1478b509e0"
        }
    }
}
```

Es folgt eine Beispielausgabe.

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-068f72b724example",
        "LaunchTemplateName": "my-template-for-capacity-block",
        "CreateTime": "2023-10-27T15:12:44.000Z",
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

Sie können den folgenden describe-launch-template-versions-Befehl verwenden, um die Reservierungs-ID für den Kapazitätsblock zu überprüfen, die der Startvorlage zugeordnet ist.

```
aws ec2 describe-launch-template-versions --launch-template-names my-template-for-capacity-block \
  --region us-east-2
```

Es folgt eine Beispielausgabe für eine Startvorlage mit Angabe einer Kapazitätsblockreservierung.

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-068f72b724example",
            "LaunchTemplateName": "my-template-for-capacity-block",
            "VersionNumber": 1,
            "CreateTime": "2023-10-27T15:12:44.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-04d5cc9b88example",
                "InstanceType": "p5.48xlarge",
                "SecurityGroupIds": [
                    "sg-903004f88example"
                ],
                "KeyName": "MyKeyPair",
                "InstanceMarketOptions": {
                    "MarketType": "capacity-block"
                },
                "CapacityReservationSpecification": {
```

```
                    "CapacityReservationTarget": {
                        "CapacityReservationId": "cr-02168da1478b509e0"
                    }
                }
            }
        }
    ]
}
```

Einschränkungen

- Support für Capacity Blocks ist nur verfügbar, wenn Ihre Auto Scaling Scaling-Gruppe über eine kompatible Konfiguration verfügt. Gruppen mit gemischten Instances und warmen Pools werden nicht unterstützt.
- Sie können jeweils nur einen Kapazitätsblock als Ziel festlegen.

Zugehörige Ressourcen

- Die Voraussetzungen und Empfehlungen für die Verwendung von P5-Instances finden Sie unter Erste Schritte mit P5-Instances im Amazon EC2 EC2-Benutzerhandbuch.
- Amazon EKS unterstützt die Verwendung Capacity Blocks zur Unterstützung Ihrer kurzfristigen Workloads für maschinelles Lernen (ML) auf Amazon EKS-Clustern. Weitere Informationen finden Sie unter Capacity Blocksfür ML im Amazon EKS-Benutzerhandbuch.
- Sie können es Capacity Blocks mit unterstützten Instance-Typen und Regionen verwenden. Kapazitätsreservierungen auf Abruf bieten jedoch die Flexibilität, Kapazität für andere Instance-Typen und Regionen zu reservieren. Ein Tutorial, das Ihnen zeigt, wie Sie die Option Kapazitätsreservierung auf Abruf verwenden, finden Sie unterVerwenden Sie On-Demand-Kapazitätsreservierungen, um Kapazitäten in bestimmten Availability Zones zu reservieren..

Migrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten

Ab 2023 können Sie `CreateLaunchConfiguration` nicht mit neuen Typen von Amazon-EC2-Instances aufrufen, die nach dem 31. Dezember 2022 veröffentlicht wurden. Weitere Informationen finden Sie unter Startkonfigurationen.

Gehen Sie wie folgt vor, um Ihre Auto Scaling Scaling-Gruppen von Startkonfigurationen zu Startvorlagen zu migrieren.

> **Important**
>
> Stellen Sie sicher, dass Sie über die erforderlichen Berechtigungen zum Arbeiten mit Startvorlagen verfügen. Weitere Informationen finden Sie unter Berechtigungen für die Arbeit mit Startvorlagen.

Schritt 1: Suchen Sie Auto-Scaling-Gruppen, die Startkonfigurationen verwenden

Um festzustellen, ob Sie Auto-Scaling-Gruppen haben, die noch Startkonfigurationen verwenden, führen Sie den folgenden describe-auto-scaling-groups-Befehl mit AWS CLI aus. Ersetzen Sie *REGION* durch Ihre AWS-Region.

```
aws autoscaling describe-auto-scaling-groups --region REGION \
  --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

Es folgt eine Beispielausgabe.

```
[
    {
 "AutoScalingGroupName": "group-1",
 "AutoScalingGroupARN": "arn",
 "LaunchConfigurationName": "my-launch-config",
 "MinSize": 1,
 "MaxSize": 5,
 "DesiredCapacity": 2,
 "DefaultCooldown": 300,
 "AvailabilityZones": [
           "us-west-2a",
           "us-west-2b",
           "us-west-2c"
       ],
 "LoadBalancerNames": [],
 "TargetGroupARNs": [],
 "HealthCheckType": "EC2",
 "HealthCheckGracePeriod": 300,
 "Instances": [
```

```
            {
                "ProtectedFromScaleIn": false,
                "AvailabilityZone": "us-west-2a",
                "LaunchConfigurationName": "my-launch-config",
                "InstanceId": "i-05b4f7d5be44822a6",
                "InstanceType": "t3.micro",
                "HealthStatus": "Healthy",
                "LifecycleState": "InService"
            },
            {
                "ProtectedFromScaleIn": false,
                "AvailabilityZone": "us-west-2b",
                "LaunchConfigurationName": "my-launch-config",
                "InstanceId": "i-0c20ac468fa3049e8",
                "InstanceType": "t3.micro",
                "HealthStatus": "Healthy",
                "LifecycleState": "InService"
            }
 ],
 "CreatedTime": "2023-03-09T22:15:11.611Z",
 "SuspendedProcesses": [],
 "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
 "EnabledMetrics": [],
 "Tags": [
            {
  "ResourceId": "group-1",
  "ResourceType": "auto-scaling-group",
  "Key": "environment",
  "Value": "production",
  "PropagateAtLaunch": true
            }
       ],
 "TerminationPolicies": [
     "Default"
 ],
 "NewInstancesProtectedFromScaleIn": false,
 "ServiceLinkedRoleARN": "arn",
       "TrafficSources": []
    },

    ... additional groups ...

]
```

Führen Sie alternativ den folgenden Befehl aus, um alles außer den Auto-Scaling-Gruppe-Namen mit den Namen ihrer jeweiligen Startkonfigurationen und Tags in der Ausgabe zu entfernen:

```
aws autoscaling describe-auto-scaling-groups --region REGION \
  --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`].{AutoScalingGroupName:
 AutoScalingGroupName, LaunchConfigurationName: LaunchConfigurationName, Tags: Tags}'
```

Das folgende Beispiel zeigt eine Ausgabe.

```
[
    {
        "AutoScalingGroupName": "group-1",
        "LaunchConfigurationName": "my-launch-config",
        "Tags": [
            {
                "ResourceId": "group-1",
                "ResourceType": "auto-scaling-group",
                "Key": "environment",
                "Value": "production",
                "PropagateAtLaunch": true
            }
        ]
    },

    ... additional groups ...

]
```

Weitere Informationen zum Filtern finden Sie im AWS Command Line Interface Benutzerhandbuch unter Filtern der AWS CLI Ausgabe.

Schritt 2: Kopieren einer Startkonfiguration in eine Startvorlage

Mit dem folgenden Verfahren können Sie eine Startkonfiguration in eine Startvorlage kopieren. Dann können Sie sie zu Ihrer Auto-Scaling-Gruppe hinzufügen.

Das Kopieren mehrerer Startkonfigurationen führt zu Startvorlagen mit identischem Namen. Um den Namen zu ändern, der einer Startvorlage während des Kopiervorgangs gegeben wurde, müssen Sie die Startkonfigurationen eine nach der anderen kopieren.

> **Note**
>
> Die Kopierfunktion steht nur über die Konsole zur Verfügung.

Kopieren einer Startkonfiguration in eine Startvorlage (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im linken Navigationsbereich unter Auto Scaling Auto-Scaling-Gruppen aus.
3. Wählen Sie oben auf der Seite Startkonfigurationen aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Startkonfigurationen anzeigen aus, um zu bestätigen, dass Sie die Seite Startkonfigurationen aufrufen möchten.
4. Wählen Sie die zu kopierende Startkonfiguration und Copy to launch template, Copy selected (In Startvorlage kopieren, Kopie ausgewählt) aus. Dadurch wird eine neue Startvorlage mit demselben Namen und denselben Optionen wie bei der ausgewählten Startkonfiguration eingerichtet.
5. Unter New launch template name (Neuer Startvorlagenname) können Sie den Namen der Startkonfiguration (Standard) verwenden oder einen neuen Namen eingeben. Die Namen von Startvorlagen müssen eindeutig sein.
6. (Optional) Wählen Sie Eine Auto-Scaling-Gruppe mithilfe der neuen Vorlage erstellen aus.

   Sie können diesen Schritt überspringen, wenn Sie das Kopieren der Startkonfiguration abschließen möchten. Sie müssen keine neue Auto-Scaling-Gruppe erstellen.
7. Wählen Sie die Option Kopieren aus.

So kopieren Sie alle Startkonfigurationen in Startvorlagen (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Auto Scaling die Option Launch Configurations (Startkonfigurationen) aus.
3. Klicken Sie auf Kopieren zur Startvorlage, Alle kopieren. Dadurch wird jede Startkonfiguration in der aktuellen Region in eine neue Startvorlage mit demselben Namen und denselben Optionen kopiert.
4. Wählen Sie die Option Kopieren aus.

Schritt 3: Aktualisieren einer Auto-Scaling-Gruppe zum Verwenden einer Startvorlage

Wenn Sie eine Startvorlage erstellt haben, können Sie sie zu Ihrer Auto-Scaling-Gruppe hinzufügen.

Aktualisieren einer Auto-Scaling-Gruppe zum Verwenden einer Startvorlage (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet, in dem Informationen über die ausgewählte Gruppe angezeigt werden.
3. Wählen Sie auf der Registerkarte Details die Option Konfiguration starten, Bearbeiten aus.
4. Wählen Sie So wechseln Sie zur Startvorlage aus.
5. Wählen Sie als Launch Template (Startvorlage) Ihre Startvorlage aus.
6. Als Version wählen Sie ggf. die Version der Startvorlage aus. Nachdem Sie Versionen einer Startvorlage erstellt haben, können Sie auswählen, ob die Auto-Scaling-Gruppe beim Hochskalieren die standardmäßige oder die neueste Version der Startvorlage verwenden soll.
7. Wählen Sie Aktualisieren.

Aktualisieren einer Auto-Scaling-Gruppe zum Verwenden einer Startvorlage (AWS CLI)

Der folgende update-auto-scaling-group-Befehl aktualisiert die angegebene Auto-Scaling-Gruppe, um die ursprüngliche Version der angegebenen Startvorlage zu verwenden.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='1'
```

Weitere Beispiele für die Verwendung von CLI-Befehlen, um eine Auto-Scaling-Gruppe zur Verwendung einer Startvorlage zu aktualisieren, finden Sie unter Aktualisieren einer Auto-Scaling-Gruppe zum Verwenden einer Startvorlage.

Schritt 4: Ersetzen Ihrer Instances

Wenn Sie die Startkonfiguration durch eine Startvorlage ersetzt haben, verwenden alle neuen Instances die neue Startvorlage. Bestehende Instances sind nicht betroffen.

Um vorhandene Instances zu aktualisieren, können Sie eine Instance-Aktualisierung verwenden, um die Instances in der Auto-Scaling-Gruppe zu ersetzen, anstatt Instances gleichzeitig manuell zu ersetzen. Weitere Informationen finden Sie unter Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren. Eine Instance-Aktualisierung kann besonders hilfreich sein, wenn die Gruppe groß ist.

Alternativ können Sie die automatische Skalierung zulassen, um vorhandene Instances auf Grundlage der Beendigungsrichtlinien der Gruppe schrittweise durch neue Instances zu ersetzen, oder Sie können sie beenden. Das manuelle Beenden zwingt Ihre Auto-Scaling-Gruppe, neue Instances zu starten, um die gewünschte Kapazität der Gruppe aufrechtzuerhalten. Weitere Informationen finden Sie unter Terminate an Instance im Amazon EC2 EC2-Benutzerhandbuch.

Zusätzliche Informationen

Weitere Informationen finden Sie im AWS Compute-Blog unter Amazon EC2 Auto Scaling wird keine Unterstützung mehr für neue EC2-Funktionen zu Launch-Konfigurationen hinzufügen.

Ein Thema, das Ihnen zeigt, wie Sie AWS CloudFormation Stacks von Startkonfigurationen zu Startvorlagen migrieren, finden Sie unter. Migrieren Sie AWS CloudFormation Stacks zu Startvorlagen

Migrieren Sie AWS CloudFormation Stacks zu Startvorlagen

Sie können Ihre vorhandenen AWS CloudFormation Stack-Vorlagen von Startkonfigurationen zu Startvorlagen migrieren. Fügen Sie dazu eine Startvorlage direkt zu einer vorhandenen Stack-Vorlage hinzu und verknüpfen Sie die Startvorlage dann mit der Auto-Scaling-Gruppe in der Stack-Vorlage. Verwenden Sie anschließend die geänderte Vorlage zum Aktualisieren Ihres Stacks.

Bei der Migration zu Startvorlagen spart Ihnen dieses Thema Zeit, da es Anweisungen zum Umschreiben der Startkonfigurationen in Ihren CloudFormation Stack-Vorlagen als Startvorlagen enthält. Weitere Informationen zum Migrieren von Startkonfigurationen in Startvorlagen finden Sie unter Migrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten.

Themen

- Auto-Scaling-Gruppen finden, die eine Startkonfiguration verwenden
- Aktualisieren eines Stacks zur Verwendung einer Startvorlage
- Das Aktualisierungsverhalten von Stack-Ressourcen verstehen

- Verfolgen Sie die Migration
- Referenz für die Abbildung der Startkonfiguration

Auto-Scaling-Gruppen finden, die eine Startkonfiguration verwenden

So finden Sie Auto-Scaling-Gruppen, die eine Startkonfiguration verwenden

- Verwenden Sie den folgenden Befehl describe-auto-scaling-groups, um die Namen der Auto-Scaling-Gruppen aufzulisten, die Startkonfigurationen in der angegebenen Region verwenden. Fügen Sie die `--filters` Option hinzu, die Ergebnisse auf Gruppen einzugrenzen, die einem CloudFormation Stack zugeordnet sind (durch Filtern nach dem `aws:cloudformation:stack-name` Tag-Schlüssel).

```
aws autoscaling describe-auto-scaling-groups --region REGION \
  --filters Name=tag-key,Values=aws:cloudformation:stack-name \
  --query 'AutoScalingGroups[?LaunchConfigurationName!
=`null`].AutoScalingGroupName'
```

Das folgende Beispiel zeigt eine Ausgabe.

```
[
    "{stack-name}-group-1",
    "{stack-name}-group-2",
    "{stack-name}-group-3"
]
```

Sie finden weitere nützliche AWS CLI Befehle, um Auto Scaling Scaling-Gruppen für die Migration zu finden und die Ausgabe zu filternMigrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten.

> **Important**
>
> Wenn Ihre Stack-Ressourcen `AWSEB` in ihrem Namen stehen, bedeutet das, dass sie durch erstellt wurden AWS Elastic Beanstalk. In diesem Fall müssen Sie die Beanstalk-Umgebung aktualisieren, um Elastic Beanstalk anzuweisen, die Startkonfiguration zu entfernen und sie durch eine Startvorlage zu ersetzen.

Aktualisieren eines Stacks zur Verwendung einer Startvorlage

Befolgen Sie die Schritte in diesem Abschnitt, um Folgendes zu tun:

- Schreiben Sie die Startkonfiguration als Startvorlage um und verwenden Sie die entsprechenden Eigenschaften der Startvorlage.
- Verknüpfen Sie die neue Startvorlage mit der Auto-Scaling-Gruppe.
- Stellen Sie diese Updates bereit.

So ändern Sie die Stack-Vorlage und aktualisieren den Stack

1. Folgen Sie den gleichen allgemeinen Verfahren zum Ändern der Stack-Vorlage, die im AWS CloudFormation Benutzerhandbuch unter Ändern einer Stack-Vorlage beschrieben sind.
2. Schreiben Sie die Startkonfiguration in eine Startvorlage um. Sehen Sie sich das folgende Beispiel an:

   Beispiel: Eine einfache Startkonfiguration

```
---
Resources:
  myLaunchConfig:
    Type: AWS::AutoScaling::LaunchConfiguration
    Properties:
      ImageId: ami-02354e95b3example
      InstanceType: t3.micro
      SecurityGroups:
        - !Ref EC2SecurityGroup
      KeyName: MyKeyPair
      BlockDeviceMappings:
        - DeviceName: /dev/xvda
          Ebs:
            VolumeSize: 150
            DeleteOnTermination: true
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash -xe
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource myASG
 --region ${AWS::Region}
```

Beispiel: Das Äquivalent zur Startvorlage

```
---
Resources:
  myLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
      LaunchTemplateData:
        ImageId: ami-02354e95b3example
        InstanceType: t3.micro
        SecurityGroupIds:
          - Ref! EC2SecurityGroup
        KeyName: MyKeyPair
        BlockDeviceMappings:
          - DeviceName: /dev/xvda
            Ebs:
              VolumeSize: 150
              DeleteOnTermination: true
        UserData:
          Fn::Base64: !Sub |
            #!/bin/bash -x
            yum install -y aws-cfn-bootstrap
            /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
 myASG --region ${AWS::Region}
```

Referenzinformationen zu allen Eigenschaften, die Amazon EC2 unterstützt, finden Sie AWS::EC2::LaunchTemplateim AWS CloudFormation Benutzerhandbuch.

Beachten Sie, dass die Startvorlage die Eigenschaft `LaunchTemplateName` mit einem Wert von `!Sub ${AWS::StackName}-launch-template` enthält. Dies ist erforderlich, wenn der Name der Startvorlage den Namen des Stacks enthalten soll.

3. Wenn die Eigenschaft **`IamInstanceProfile`** in Ihrer Startkonfiguration vorhanden ist, müssen Sie sie in eine Struktur umwandeln und entweder den Namen oder den ARN des Instance-Profils angeben. Ein Beispiel finden Sie unter AWS::EC2::LaunchTemplate.
4. Wenn die Eigenschaften **`AssociatePublicIpAddress`**, **`InstanceMonitoring`** oder **`PlacementTenancy`** in Ihrer Startkonfiguration vorhanden sind, müssen Sie diese in eine Struktur umwandeln. Beispiele finden Sie unter AWS::EC2::LaunchTemplate.

Eine Ausnahme besteht, wenn der Wert für die Eigenschaft `MapPublicIpOnLaunch` in den Teilnetzen, die Sie für Ihre Auto-Scaling-Gruppe verwendet haben, mit dem Wert für die Eigenschaft `AssociatePublicIpAddress` in Ihrer Startkonfiguration übereinstimmt. In diesem Fall können Sie die `AssociatePublicIpAddress`-Eigenschaft ignorieren. Die `AssociatePublicIpAddress`-Eigenschaft wird nur verwendet, um die `MapPublicIpOnLaunch`-Eigenschaft zu überschreiben und zu ändern, ob Instances beim Start eine öffentliche IPv4-Adresse erhalten.

5. Sie können Sicherheitsgruppen aus der **`SecurityGroups`**-Eigenschaft an eine von zwei Stellen in Ihrer Startvorlage kopieren. Normalerweise kopieren Sie die Sicherheitsgruppen in die `SecurityGroupIds`-Eigenschaft. Wenn Sie jedoch in Ihrer Startvorlage eine `NetworkInterfaces`-Struktur erstellen, um die `AssociatePublicIpAddress`-Eigenschaft anzugeben, müssen Sie stattdessen die Sicherheitsgruppen in die `Groups`-Eigenschaft der Netzwerkschnittstelle kopieren.
6. Wenn in Ihrer Startkonfiguration `BlockDeviceMapping`-Strukturen vorhanden sind und **`NoDevice`** auf `true` gesetzt ist, müssen Sie in Ihrer Startvorlage eine leere Zeichenfolge für `NoDevice` angeben, damit Amazon EC2 das Gerät auslässt.
7. Wenn die Eigenschaft **`SpotPrice`** in Ihrer Startkonfiguration vorhanden ist, empfehlen wir Ihnen, sie in Ihrer Startvorlage wegzulassen. Ihre Spot Instance wird zum aktuellen Spot-Preis gestartet. Dieser Preis wird niemals den On-Demand-Preis überschreiten.

   Um Spot-Instances anzufordern, haben Sie zwei Optionen, die sich gegenseitig ausschließen:

   - Die erste Möglichkeit besteht darin, die `InstanceMarketOptions`-Struktur in Ihrer Startvorlage zu verwenden (nicht empfohlen). Weitere Informationen finden Sie AWS::EC2::LaunchTemplate InstanceMarketOptionsim AWS CloudFormation Benutzerhandbuch.
   - Die andere besteht darin, Ihrer Auto-Scaling-Gruppe eine `MixedInstancesPolicy`-Struktur hinzuzufügen. Auf diese Weise stehen Ihnen mehr Optionen zur Verfügung, wie Sie die Anfrage stellen können. Eine Spot-Instance-Anfrage in Ihrer Startvorlage unterstützt nicht mehr als eine Instance-Typauswahl pro Auto-Scaling-Gruppe. Eine Richtlinie für gemischte Instances unterstützt jedoch die Auswahl von mehr als einem Instance-Typ pro Auto-Scaling-Gruppe. Spot-Instance-Anfragen profitieren davon, dass mehr als ein einziger Instance-Typ zur Auswahl steht. Weitere Informationen finden Sie MixedInstancesPolicy im AWS CloudFormation Benutzerhandbuch unter AWS::AutoScaling::AutoScaling MixedInstancesPolicy.

8. Entfernen Sie die **`LaunchConfigurationName`** Eigenschaft aus der AWS::AutoScaling::AutoScaling. Fügen Sie stattdessen die Startvorlage hinzu.

   In den folgenden Beispielen ruft die intrinsische Funktion Ref die ID der AWS::EC2::LaunchTemplateRessource mit der logischen ID ab. `myLaunchTemplate` Die GetAttFunktion ruft die neueste Versionsnummer (z. B.1) der Startvorlage für die `Version` Eigenschaft ab.

   Beispiel: Ohne eine Richtlinie für gemischte Instances

   ```
   ---
   Resources:
     myASG:
       Type: AWS::AutoScaling::AutoScalingGroup
       Properties:
         LaunchTemplate:
           LaunchTemplateId: !Ref myLaunchTemplate
           Version: !GetAtt myLaunchTemplate.LatestVersionNumber
         ...
   ```

   Beispiel: Mit einer Richtlinie für gemischte Instances

   ```
   ---
   Resources:
     myASG:
       Type: AWS::AutoScaling::AutoScalingGroup
       Properties:
         MixedInstancesPolicy:
           LaunchTemplate:
             LaunchTemplateSpecification:
               LaunchTemplateId: !Ref myLaunchTemplate
               Version: !GetAtt myLaunchTemplate.LatestVersionNumber
         ...
   ```

   Referenzinformationen zu allen Eigenschaften, die Amazon EC2 Auto Scaling unterstützt, finden Sie unter AWS::AutoScaling::AutoScalingAWS::AutoScaling::AutoScalingGroup im AWS CloudFormation Benutzerhandbuch.

9. Wenn Sie bereit sind, diese Updates bereitzustellen, folgen Sie den CloudFormation Verfahren, um den Stack mit Ihrer geänderten Stack-Vorlage zu aktualisieren. Weitere Informationen finden Sie unter Ändern einer Stack-Vorlage im AWS CloudFormation Benutzerhandbuch.

Das Aktualisierungsverhalten von Stack-Ressourcen verstehen

CloudFormation aktualisiert die Stack-Ressourcen, indem die Änderungen zwischen der von Ihnen bereitgestellten aktualisierten Vorlage und den Ressourcenkonfigurationen, die Sie in der vorherigen Version Ihrer Stack-Vorlage beschrieben haben, verglichen werden. Nicht geänderte Ressourcenkonfigurationen bleiben während des Updates davon unberührt.

CloudFormation unterstützt das UpdatePolicyAttribut für Auto Scaling Scaling-Gruppen. Wenn während eines Updates auf eingestellt `UpdatePolicy` ist`AutoScalingRollingUpdate`, werden `InService` Instanzen CloudFormation ersetzt, nachdem Sie die Schritte in diesem Verfahren ausgeführt haben. Wenn auf gesetzt `UpdatePolicy` ist`AutoScalingReplacingUpdate`, CloudFormation ersetzt die Auto Scaling Scaling-Gruppe und ihren Warmpool (falls vorhanden).

Wenn Sie kein `UpdatePolicy` Attribut für Ihre Auto Scaling Scaling-Gruppe angegeben haben, wird die Startvorlage auf ihre Richtigkeit überprüft, aber CloudFormation es werden keine Änderungen für die Instances in der Auto Scaling Scaling-Gruppe bereitgestellt. Alle neuen Instances verwenden Ihre Startvorlage, aber bestehende Instances werden weiterhin mit der Startkonfiguration ausgeführt, mit der sie ursprünglich gestartet wurden (obwohl die Startkonfiguration nicht mehr existiert). Die Ausnahme ist, wenn Sie Ihre Kaufoptionen ändern, z. B. indem Sie eine Police für gemischte Instances hinzufügen. In diesem Fall ersetzt Ihre Auto-Scaling-Gruppe die vorhandenen Instances nach und nach durch neue Instances, die den neuen Kaufoptionen entsprechen.

Verfolgen Sie die Migration

So verfolgen Sie die Migration

1. Wählen Sie in der AWS CloudFormation -Konsole den Stack aus, den Sie aktualisiert haben, und klicken Sie dann auf die Registerkarte Events (Ereignisse), um die Stack-Ereignisse anzuzeigen.
2. Um die Ereignisliste mit den neuesten Ereignissen zu aktualisieren, klicken Sie in der CloudFormation Konsole auf die Schaltfläche „Aktualisieren“.
3. Während Ihr Stack aktualisiert wird, werden Sie mehrere Ereignisse für jede Ressourcenaktualisierung feststellen. Wenn Sie in der Spalte Statusgrund eine Ausnahme sehen, die auf ein Problem beim Erstellen der Startvorlage hinweist, finden Sie weitere Informationen Fehlersuche bei Amazon EC2 Auto Scaling: Startvorlagen zu möglichen Ursachen.
4. (Optional) Je nach Verwendung des `UpdatePolicy`-Attributs können Sie den Fortschritt Ihrer Auto-Scaling-Gruppe auf der Seite Auto-Scaling-Gruppen der Amazon EC2-Konsole überwachen. Wählen Sie die Auto-Scaling-Gruppe aus. Auf der Registerkarte Activity (Aktivität)

unter Activity history (Aktivitätsverlauf) zeigt die Spalte Status an, ob Ihre Auto-Scaling-Gruppe-Instances erfolgreich gestartet oder beendet hat oder ob die Skalierungsaktivität noch im Gange ist.

5. Wenn das Stack-Update abgeschlossen ist, wird CloudFormation ein `UPDATE_COMPLETE` Stack-Ereignis ausgelöst. Weitere Informationen finden Sie unter Überwachung des Fortschritts einer Stack-Aktualisierung im AWS CloudFormation Benutzerhandbuch.
6. Nachdem das Stack-Update abgeschlossen ist, öffnen Sie die Seite Startvorlagen und Startkonfigurationen der Amazon EC2-Konsole. Sie werden feststellen, dass eine neue Startvorlage erstellt und die Startkonfiguration gelöscht wurde.

Referenz für die Abbildung der Startkonfiguration

Zu Referenzzwecken sind in der folgenden Tabelle alle Eigenschaften der AWS::AutoScaling::LaunchConfigurationRessource auf oberster Ebene mit den entsprechenden Eigenschaften in der AWS::EC2::LaunchTemplateRessource aufgeführt.

Startkonfiguration Quelleigenschaft	Ziel-Eigenschaft der Startvorlage
`AssociatePublicIpAddress`	`NetworkInterfaces.AssociatePublicIpAddress`
`BlockDeviceMappings`	`BlockDeviceMappings`
`ClassicLinkVPCId`	Nicht verfügbar[1]
`ClassicLinkVPCSecurityGroups`	Nicht verfügbar[1]
`EbsOptimized`	`EbsOptimized`
`IamInstanceProfile`	Entweder `IamInstanceProfile.Arn` oder `IamInstanceProfile.Name`, jedoch nicht beides.
`ImageId`	`ImageId`
`InstanceId`	`InstanceId`
`InstanceMonitoring`	`Monitoring.Enabled`

Startkonfiguration Quelleigenschaft	Ziel-Eigenschaft der Startvorlage
`InstanceType`	`InstanceType`
`KernelId`	`KernelId`
`KeyName`	`KeyName`
`LaunchConfigurationName`	`LaunchTemplateName`
`MetadataOptions`	`MetadataOptions`
`PlacementTenancy`	`Placement.Tenancy`
`RamDiskId`	`RamDiskId`
`SecurityGroups`	Entweder `SecurityGroupIds` oder `NetworkInterfaces.Groups`, jedoch nicht beides.
`SpotPrice`	`InstanceMarketOptions.SpotOptions.MaxPrice`
`UserData`	`UserData`

[1] Die `ClassicLinkVPCSecurityGroups` Eigenschaften `ClassicLinkVPCId` und können nicht in einer Startvorlage verwendet werden, da EC2-Classic nicht mehr verfügbar ist.

Beispiele für die Erstellung und Verwaltung von Startvorlagen mit dem AWS CLI

Sie können Startvorlagen mit den SDKs AWS Management Console, AWS Command Line Interface (AWS CLI) oder verwalten. In diesem Abschnitt finden Sie Beispiele für die Erstellung und Verwaltung von Startvorlagen für Amazon EC2 Auto Scaling aus dem AWS CLI.

Inhalt

- Angeben von Tags, die Instances beim Start kennzeichnen
- Angeben einer IAM-Rolle, die an Instances übergeben wird
- Zuweisen einer öffentlichen IP-Adresse
- Angeben eines Benutzerdatenskripts, das Instances beim Start konfiguriert
- Angeben einer Blockgerät-Zuweisung für ein AMI
- Festlegen von Dedicated Hosts zur Bereitstellung von Softwarelizenzen externer Anbieter
- Angeben einer vorhandenen Netzwerkschnittstelle
- Erstellen mehrerer Netzwerkschnittstellen
- Verwalten Ihrer Startvorlagen
- Aktualisieren einer Auto-Scaling-Gruppe zum Verwenden einer Startvorlage

Beispielverwendung

```
{
    "LaunchTemplateName": "my-template-for-auto-scaling",
    "VersionDescription": "test description",
    "LaunchTemplateData": {
        "ImageId": "ami-04d5cc9b88example",
        "InstanceType": "t2.micro",
        "SecurityGroupIds": [
            "sg-903004f88example"
        ],
        "KeyName": "MyKeyPair",
        "Monitoring": {
            "Enabled": true
        },
        "Placement": {
            "Tenancy": "dedicated"
        },
        "CreditSpecification": {
            "CpuCredits": "unlimited"
        },
        "MetadataOptions": {
            "HttpTokens": "required",
            "HttpPutResponseHopLimit": 1,
            "HttpEndpoint": "enabled"
        }
    }
```

```
}
```

Erstellen einer grundlegenden Startvorlage

Um eine grundlegende Startvorlage zu erstellen, verwenden Sie den create-launch-template-Befehl wie folgt, mit diesen Änderungen:

- Ersetzen Sie `ami-04d5cc9b88example` mit der ID des AMI, von dem aus die Instances gestartet werden sollen.
- Ersetzen Sie `t2.micro` mit einem Instance-Typ, der kompatibel mit dem angegebenen AMI ist.

Dieses Beispiel erstellt eine Startvorlage mit dem Namen *my-template-for-auto-scaling*. Wenn die mit dieser Einführungsvorlage erstellten Instances in einer Standard-VPC gestartet werden, erhalten sie standardmäßig eine öffentliche IP-Adresse. Wenn die Instances in einer nicht standardmäßigen VPC gestartet werden, erhalten sie keine öffentliche Adresse.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data
 '{"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

Weitere Informationen zum Ansetzen von JSON-formatierten Parametern finden Sie unter Verwenden von Anführungszeichen mit Zeichenfolgen in der AWS CLI im AWS Command Line Interface -Benutzerhandbuch.

Alternativ können Sie die JSON-formatierten Parameter in einer Konfigurationsdatei angeben.

Im folgenden Beispiel wird eine einfache Startvorlage erstellt, die auf eine Konfigurationsdatei für Startvorlagenparameterwerte verweist.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data file://config.json
```

Inhalt von `config.json`:

```
{
    "ImageId":"ami-04d5cc9b88example",
```

```
    "InstanceType":"t2.micro"
}
```

Angeben von Tags, die Instances beim Start kennzeichnen

Im folgenden Beispiel wird Instances beim Start ein Tag hinzugefügt (z. B. `purpose=webserver`).

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"TagSpecifications":[{"ResourceType":"instance","Tags":
[{"Key":"purpose","Value":"webserver"}]}],"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.
```

> **Note**
>
> Wenn Sie Instance-Tags in Ihrer Startvorlage angeben und sich dann dafür entschieden haben, die Tags Ihrer Auto-Scaling-Gruppe an ihre Instances zu übertragen, werden alle Tags zusammengeführt. Wenn derselbe Tag-Schlüssel für einen Tag in Ihrer Startvorlage und einen Tag in Ihrer Auto-Scaling-Gruppe angegeben wird, hat der Tag-Wert aus der Gruppe Vorrang.

Angeben einer IAM-Rolle, die an Instances übergeben wird

Im folgenden Beispiel wird der Name des Instance-Profils angegeben, das der IAM-Rolle zugeordnet ist, das beim Start an Instances übergeben wird. Weitere Informationen finden Sie unter IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
--launch-template-data '{"IamInstanceProfile":{"Name":"my-instance-
profile"},"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

Zuweisen einer öffentlichen IP-Adresse

Im folgenden create-launch-template-Beispiel wird eine Startvorlage erstellt und dafür konfiguriert, Instances, die in einer nicht standardmäßigen VPC gestartet werden, öffentliche Adressen zuzuweisen.

> **Note**
>
> Wenn Sie eine Netzwerkschnittstelle angeben, geben Sie einen Wert für `Groups` an, der den Sicherheitsgruppen der VPC, in der Ihre Auto-Scaling-Gruppe Instances starten wird, entspricht. Geben Sie die VPC-Subnetze als Eigenschaften der Auto-Scaling-Gruppe an.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":
[{"DeviceIndex":0,"AssociatePublicIpAddress":true,"Groups":
["sg-903004f88example"],"DeleteOnTermination":true}],"ImageId":"ami-04d5cc9b88example","Instanc
```

Angeben eines Benutzerdatenskripts, das Instances beim Start konfiguriert

Im folgenden Beispiel wird ein Benutzerdatenskript als base64-kodierte Zeichenfolge angegeben, die Instances beim Start konfiguriert. Der create-launch-template-Befehl benötigt base64-verschlüsselte Benutzerdaten.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
--launch-template-data
 '{"UserData":"IyEvYmluL2Jhc...","ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

Angeben einer Blockgerät-Zuweisung für ein AMI

Im folgenden create-launch-template-Befehl wird eine Startvorlage mit einer Blockgeräte-Zuweisung erstellt: ein 22-Gigabyte-EBS-Volume, das `/dev/xvdcz` zugeordnet ist. Das Volume `/dev/xvdcz` verwendet den Volume-Typ General Purpose SSD (gp2) und wird beim Beenden der Instance, an die es angehängt ist, gelöscht.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"BlockDeviceMappings":[{"DeviceName":"/dev/xvdcz","Ebs":
{"VolumeSize":22,"VolumeType":"gp2","DeleteOnTermination":true}}],"ImageId":"ami-04d5cc9b88exam
```

Festlegen von Dedicated Hosts zur Bereitstellung von Softwarelizenzen externer Anbieter

Wenn Sie eine Host-Tenancy angeben, können Sie eine Host-Ressourcengruppe und eine License Manager-Konfiguration angeben, um berechtigte Softwarelizenzen von externen Anbietern bereitzustellen. Anschließend können Sie die Lizenzen auf EC2-Instances verwenden, indem Sie den folgenden create-launch-template-Befehl verwenden.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"Placement":
{"Tenancy":"host","HostResourceGroupArn":"arn"},"LicenseSpecifications":
[{"LicenseConfigurationArn":"arn"}],"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"
```

Angeben einer vorhandenen Netzwerkschnittstelle

Das folgende create-launch-template-Beispiel konfiguriert die primäre Netzwerkschnittstelle so, dass eine vorhandene Netzwerkschnittstelle verwendet wird.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":
[{"DeviceIndex":0,"NetworkInterfaceId":"eni-
b9a5ac93","DeleteOnTermination":false}],"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.mi
```

Erstellen mehrerer Netzwerkschnittstellen

Das folgende create-launch-template-Beispiel fügt eine sekundäre Netzwerkschnittstelle hinzu. Die primäre Netzwerkschnittstelle hat einen Geräteindex von 0, und die sekundäre Netzwerkschnittstelle hat einen Geräteindex von 1.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":[{"DeviceIndex":0,"Groups":
["sg-903004f88example"],"DeleteOnTermination":true},{"DeviceIndex":1,"Groups":
["sg-903004f88example"],"DeleteOnTermination":true}],"ImageId":"ami-04d5cc9b88example","Instanc
```

Wenn Sie einen Instance-Typ verwenden, der mehrere Netzwerkkarten und Elastic Fabric Adapter (EFAs) unterstützt, können Sie einer sekundären Netzwerkkarte eine sekundäre Schnittstelle

hinzufügen und EFA mithilfe des folgenden create-launch-template-Befehls aktivieren. Weitere Informationen finden Sie unter Hinzufügen einer EFA zu einer Startvorlage im Amazon EC2 EC2-Benutzerhandbuch.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":
[{"NetworkCardIndex":0,"DeviceIndex":0,"Groups":
["sg-7c2270198example"],"InterfaceType":"efa","DeleteOnTermination":true},
{"NetworkCardIndex":1,"DeviceIndex":1,"Groups":
["sg-7c2270198example"],"InterfaceType":"efa","DeleteOnTermination":true}],"ImageId":"ami-09d95
```

Warning

Der Instance-Typ p4d.24xlarge verursacht höhere Kosten als die anderen Beispiele in diesem Abschnitt. Weitere Informationen zu Preisen für P4d-Instances erhalten Sie unter Preise für Amazon-EC2-P4d-Instances.

Note

Das Anfügen mehrerer Netzwerkschnittstellen aus demselben Subnetz an eine Instance kann asymmetrisches Routing einführen, insbesondere bei Instances, die eine Linux-Variante verwenden, die nicht von Amazon stammt. Wenn Sie diese Art von Konfiguration benötigen, müssen Sie die sekundäre Netzwerkschnittstelle innerhalb des Betriebssystems konfigurieren. Ein Beispiel finden Sie unter Wie kann ich dafür sorgen, dass meine sekundäre Netzwerkschnittstelle in meiner Ubuntu EC2-Instance funktioniert? im AWS Knowledge Center.

Verwalten Ihrer Startvorlagen

Das AWS CLI beinhaltet mehrere andere Befehle, mit denen Sie Ihre Startvorlagen verwalten können.

Inhalt

- Löschen einer Startvorlagenversion
- Löschen einer Startvorlage

Auflisten und Beschreiben Ihrer Startvorlagen

Sie können zwei AWS CLI Befehle verwenden, um Informationen zu Ihren Startvorlagen abzurufen: describe-launch-templates und describe-launch-template-versions.

Mit dem describe-launch-templates-Befehl können Sie eine Liste einer von Ihnen erstellten Startvorlagen abrufen. Sie können eine Option verwenden, um Ergebnisse nach Namen einer Startvorlage, Zeit, Tag-Schlüssel oder einer Kombination aus Tag und Schlüssel-Wert filtern. Mit diesem Befehl werden zusammenfassende Informationen zu allen Ihren Startvorlagen zurückgegeben, einschließlich der Startvorlagenkennung, der neuesten Version und der Standardversion.

Das folgende Beispiel bietet eine Zusammenfassung der angegebenen Startvorlage.

```
aws ec2 describe-launch-templates --launch-template-names my-template-for-auto-scaling
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "LaunchTemplates": [
        {
            "LaunchTemplateId": "lt-068f72b729example",
            "LaunchTemplateName": "my-template-for-auto-scaling",
            "CreateTime": "2020-02-28T19:52:27.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersionNumber": 1,
            "LatestVersionNumber": 1
        }
    ]
}
```

Wenn Sie nicht die `--launch-template-names`-Option verwenden, um die Ausgabe auf eine Startvorlage zu beschränken, werden Informationen zu allen Ihren Startvorlagen zurückgegeben.

Der folgende Befehl describe-launch-template-versions liefert Informationen zu den Versionen der angegebenen Startvorlage.

```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b729example
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "LaunchTemplateVersions": [
        {
            "VersionDescription": "version1",
            "LaunchTemplateId": "lt-068f72b729example",
            "LaunchTemplateName": "my-template-for-auto-scaling",
            "VersionNumber": 1,
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "LaunchTemplateData": {
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "purpose",
                                "Value": "webserver"
                            }
                        ]
                    }
                ],
                "ImageId": "ami-04d5cc9b88example",
                "InstanceType": "t2.micro",
                "NetworkInterfaces": [
                    {
                        "DeviceIndex": 0,
                        "DeleteOnTermination": true,
                        "Groups": [
                            "sg-903004f88example"
                        ],
                        "AssociatePublicIpAddress": true
                    }
                ]
            },
            "DefaultVersion": true,
            "CreateTime": "2020-02-28T19:52:27.000Z"
        }
    ]
}
```

Erstellen einer Startvorlagenversion

Der folgende create-launch-template-version-Befehl wird eine neue Startvorlagenversion basierend auf Version 1 der Startvorlage erstellt und eine andere AMI-ID angegeben.

```
aws ec2 create-launch-template-version --launch-template-id lt-068f72b729example --
version-description version2 \
  --source-version 1 --launch-template-data "ImageId=ami-c998b6b2example"
```

Um die Standardversion der Startvorlage festzulegen, verwenden Sie den launch-template-Befehl.

Löschen einer Startvorlagenversion

Der folgende launch-template-versions-Befehl löscht die angegebene Startvorlagenversion.

```
aws ec2 delete-launch-template-versions --launch-template-id lt-068f72b729example --
versions 1
```

Löschen einer Startvorlage

Wenn eine Startvorlage nicht mehr benötigt wird, können Sie sie mit dem folgenden launch-template-Befehl löschen. Beim Löschen einer Startvorlage werden alle ihre Versionen gelöscht.

```
aws ec2 delete-launch-template --launch-template-id lt-068f72b729example
```

Aktualisieren einer Auto-Scaling-Gruppe zum Verwenden einer Startvorlage

Sie können den update-auto-scaling-group-Befehl verwenden, um eine Startvorlage zu einer bestehenden Auto-Scaling-Gruppe hinzuzufügen.

Aktualisieren einer Auto-Scaling-Gruppe, um die neueste Version einer Startvorlage zu verwenden

Der folgende update-auto-scaling-group-Befehl aktualisiert die angegebene Auto-Scaling-Gruppe, um die neueste Version der angegebenen Startvorlage zu verwenden.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateId=lt-068f72b729example,Version='$Latest'
```

Eine Auto-Scaling-Gruppe aktualisieren, um eine bestimmte Version einer Startvorlage zu verwenden

Der folgende update-auto-scaling-group-Befehl aktualisiert die angegebene Auto-Scaling-Gruppe, um eine bestimmte Version der angegebenen Startvorlage zu verwenden.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

Verwenden Sie AWS Systems Manager Parameter anstelle von AMI-IDs in Startvorlagen

In diesem Abschnitt erfahren Sie, wie Sie eine Startvorlage erstellen, die einen AWS Systems Manager Parameter angibt, der auf eine Amazon Machine Image (AMI) -ID verweist. Sie können einen Parameter verwenden, der in Ihrem eigenen gespeichert ist AWS-Konto, einen Parameter, der von einem anderen gemeinsam genutzt wird AWS-Konto, oder einen öffentlichen Parameter für ein öffentliches AMI, das von verwaltet wird, verwenden AWS.

Mit Systems-Manager-Parametern können Sie Ihre Auto-Scaling-Gruppen aktualisieren, um neue AMI-IDs zu verwenden, ohne jedes Mal, wenn sich eine AMI-ID ändert, neue Startvorlagen oder neue Versionen von Startvorlagen erstellen zu müssen. Diese IDs können sich regelmäßig ändern, z. B. wenn ein AMI mit den neuesten Betriebssystem- oder Software-Updates aktualisiert wird.

Sie können Ihre eigenen Systems Manager Manager-Parameter mithilfe des Parameterspeichers, einer Funktion von, erstellen, aktualisieren oder löschen AWS Systems Manager. Sie müssen einen Systems-Manager-Parameter erstellen, bevor Sie ihn in einer Startvorlage verwenden können. Erstellen Sie zunächst einen Parameter mit dem Datentyp `aws:ec2:image` und geben Sie als Wert die ID eines AMI ein. Die AMI-ID nimmt Form `ami-<identifier>` an, zum Beispiel `ami-123example456`. Die korrekte AMI-ID ist vom Instance-Typ und der AWS-Region abhängig, in der Sie Ihre Auto-Scaling-Gruppe starten.

Weitere Informationen zum Erstellen eines gültigen Parameters für eine AMI-ID finden Sie unter Systems Manager Manager-Parameter erstellen.

Erstellen Sie eine Startvorlage, die einen Parameter für das AMI angibt

Verwenden Sie eine der folgenden Methoden, um eine Startvorlage zu erstellen, die einen Parameter für das AMI angibt:

Console

Um eine Startvorlage mit einem AWS Systems Manager Parameter zu erstellen

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Launch Templates (Startvorlagen) und dann Create launch template (Startvorlage erstellen) aus.
3. Geben Sie für Launch template name (Startvorlagenname) einen aussagekräftigen Namen für die Startvorlage ein.
4. Wählen Sie unter Application and OS Images (Amazon Machine Image) (Anwendungs- und Betriebssystem-Images (Amazon Machine Image)) die Option Browse more AMIs (Weitere AMIs durchsuchen) aus.
5. Wählen Sie die Pfeiltaste rechts neben der Suchleiste und wählen Sie dann Benutzerdefinierten Wert/Systems-Manager-Parameter angeben aus.
6. Gehen Sie im Dialogfeld Benutzerdefinierten Wert oder Systems-Manager-Parameter angeben wie folgt vor:

   a. Geben Sie für AMI-ID oder Systems-Manager-Parameterzeichenfolge den Systems-Manager-Parameternamen in einem der folgenden Formate ein:

      Um auf einen öffentlichen Parameter zu verweisen:

      - **resolve:ssm:*public-parameter***

      Um auf einen Parameter zu verweisen, der im selben Konto gespeichert ist:

      - **resolve:ssm:*parameter-name***
      - **resolve:ssm:*parameter-name*:*version-number***
      - **resolve:ssm:*parameter-name*:*label***

      Um auf einen Parameter zu verweisen, der von einem anderen gemeinsam genutzt wird AWS-Konto:

      - **resolve:ssm:*parameter-ARN***
      - **resolve:ssm:*parameter-ARN*:*version-number***
      - **resolve:ssm:*parameter-ARN*:*label***

b. Wählen Sie Speichern.

7. Konfigurieren Sie nach Bedarf weitere Startvorlageneinstellungen und wählen Sie dann Startvorlage erstellen aus. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.

AWS CLI

Um eine Startvorlage zu erstellen, die einen Systems Manager Manager-Parameter angibt, können Sie einen der folgenden Beispielbefehle verwenden. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Beispiel: Erstellen Sie eine Startvorlage, die einen öffentlichen Parameter „ AWS-own“ angibt

Verwenden Sie die folgende Syntax: `resolve:ssm:public-parameter`, wobei `resolve:ssm` das Standardpräfix und *`public-parameter`* der Pfad und Name des öffentlichen Parameters ist.

In diesem Beispiel verwendet die Startvorlage einen von AWS-bereitgestellten öffentlichen Parameter, um Instances mit dem neuesten Amazon Linux 2-AMI in dem zu starten AWS-Region , das für Ihr Profil konfiguriert ist.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling
 --version-description version1 \
  --launch-template-data file://config.json
```

Inhalt von `config.json`:

```
{
    "ImageId":"resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2",
    "InstanceType":"t2.micro"
}
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-089c023a30example",
        "LaunchTemplateName": "my-template-for-auto-scaling",
```

```
        "CreateTime": "2022-12-28T19:52:27.000Z",
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

Beispiel: Erstellen Sie eine Startvorlage, die einen Parameter spezifiziert, der im selben Konto gespeichert ist

Verwenden Sie die folgende Syntax: `resolve:ssm:`*`parameter-name`*, wobei `resolve:ssm` das Standardpräfix und *`parameter-name`* der Systems-Manager-Parametername ist.

Im folgenden Beispiel wird eine Startvorlage erstellt, die die AMI-ID aus einem vorhandenen Systems-Manager-Parameter mit dem Namen *`golden-ami`* abruft.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling \
  --launch-template-data file://config.json
```

Inhalt von `config.json`:

```
{
    "ImageId":"resolve:ssm:golden-ami",
    "InstanceType":"t2.micro"
}
```

Die Standardversion des Parameters ist, falls nicht angegeben, die neueste Version.

Das folgende Beispiel verweist auf eine bestimmte Version des *`golden-ami`*-Parameters. Das Beispiel verwendet Version *`3`* des *`golden-ami`*-Parameters, Sie können jedoch jede gültige Versionsnummer verwenden.

```
{
    "ImageId":"resolve:ssm:golden-ami:3",
    "InstanceType":"t2.micro"
}
```

Das folgende ähnliche Beispiel verweist auf die Parameterbezeichnung*`prod`*, die einer bestimmten Version des *`golden-ami`*-Parameters zugeordnet ist.

```
{
    "ImageId":"resolve:ssm:golden-ami:prod",
    "InstanceType":"t2.micro"
}
```

Es folgt eine Beispielausgabe.

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-068f72b724example",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "CreateTime": "2022-12-27T17:11:21.000Z",
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

Beispiel: Erstellen Sie eine Startvorlage, die einen Parameter angibt, der von einem anderen gemeinsam genutzt wird AWS-Konto

Verwenden Sie die folgende Syntax:`resolve:ssm:`*`parameter-ARN`*, wobei `resolve:ssm` das Standardpräfix und *`parameter-ARN`* der ARN des Systems Manager Manager-Parameters sind.

Im folgenden Beispiel wird eine Startvorlage erstellt, die die AMI-ID aus einem vorhandenen Systems Manager Manager-Parameter mit dem ARN von abruft*`arn:aws:ssm:us-east-2:123456789012:parameter/MyParameter`*.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling
 --version-description version1 \
  --launch-template-data file://config.json
```

Inhalt von `config.json`:

```
{
    "ImageId":"resolve:ssm:arn:aws:ssm:us-east-2:123456789012:parameter/
MyParameter",
    "InstanceType":"t2.micro"
}
```

Die Standardversion des Parameters ist, falls nicht angegeben, die neueste Version.

Das folgende Beispiel verweist auf eine bestimmte Version des *MyParameter*-Parameters. Das Beispiel verwendet Version *3* des *MyParameter*-Parameters, Sie können jedoch jede gültige Versionsnummer verwenden.

```
{
    "ImageId":"resolve:ssm:arn:aws:ssm:us-east-2:123456789012:parameter/
MyParameter:3",
    "InstanceType":"t2.micro"
}
```

Das folgende ähnliche Beispiel verweist auf die Parameterbezeichnung*prod*, die einer bestimmten Version des *MyParameter*-Parameters zugeordnet ist.

```
{
    "ImageId":"resolve:ssm:arn:aws:ssm:us-east-2:123456789012:parameter/
MyParameter:prod",
    "InstanceType":"t2.micro"
}
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-00f93d4588example",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "CreateTime": "2024-01-08T12:43:21.000Z",
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

Um einen Parameter aus dem Parameterspeicher in einer Startvorlage anzugeben, benötigen Sie die `ssm:GetParameters` Berechtigung für den angegebenen Parameter. Jeder, der die Startvorlage verwendet, benötigt auch die `ssm:GetParameters` Erlaubnis, damit der Parameterwert validiert werden kann. Weitere Informationen finden Sie unter Beschränken des Zugriffs auf Systems Manager Manager-Parameter mithilfe von IAM-Richtlinien im AWS Systems Manager Benutzerhandbuch.

Stellen Sie sicher, dass eine Startvorlage die richtige AMI-ID erhält

Verwenden Sie den Befehl describe-launch-template-versions und fügen Sie die `--resolve-alias` Option hinzu, den Parameter in die tatsächliche AMI-ID aufzulösen.

```
aws ec2 describe-launch-template-versions --launch-template-name my-template-for-auto-scaling \
   --versions $Default --resolve-alias
```

Das Beispiel gibt die AMI-ID für `ImageId` zurück. Wenn eine Instance mit dieser Startvorlage gestartet wird, wird die AMI-ID zu `ami-0ac394d6a3example` aufgelöst.

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-089c023a30example",
            "LaunchTemplateName": "my-template-for-auto-scaling",
            "VersionNumber": 1,
            "CreateTime": "2022-12-28T19:52:27.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0ac394d6a3example",
                "InstanceType": "t2.micro",
            }
        }
    ]
}
```

Zugehörige Ressourcen

Weitere Informationen zur Angabe eines Systems Manager Manager-Parameters in Ihrer Startvorlage finden Sie unter Verwenden eines Systems Manager Manager-Parameters anstelle einer AMI-ID im Amazon EC2 EC2-Benutzerhandbuch.

Weitere Informationen zur Arbeit mit Systems Manager Manager-Parametern finden Sie in den folgenden Referenzmaterialien in der Systems Manager Manager-Dokumentation.

- Informationen zum Erstellen von Parameterversionen und Labels finden Sie unter Arbeiten mit Parameterversionen und Arbeiten mit Parameterbeschriftungen.

- Informationen zum Nachschlagen der öffentlichen AMI-Parameter, die von Amazon EC2 unterstützt werden, finden Sie unter Öffentliche AMI-Parameter aufrufen.
- Informationen zur gemeinsamen Nutzung von Parametern mit anderen AWS Konten oder über AWS Organizations finden Sie unter Arbeiten mit gemeinsam genutzten Parametern.
- Informationen zur Überwachung, ob Ihre Parameter erfolgreich erstellt wurden, finden Sie unter Native Parameterunterstützung für Amazon Machine Image IDs.

Einschränkungen

Beachten Sie bei der Arbeit mit Systems Manager Manager-Parametern die folgenden Einschränkungen:

- Amazon EC2 Auto Scaling unterstützt nur die Angabe von AMI-IDs als Parameter.
- Das Erstellen oder Aktualisieren von Gruppen mit gemischten Instanzen mithilfe einer Startvorlage, die einen Systems Manager Manager-Parameter angibt, wird derzeit nicht unterstützt.
- Wenn Ihre Auto Scaling Scaling-Gruppe eine Startvorlage verwendet, die einen Systems Manager Manager-Parameter spezifiziert, können Sie eine Instance-Aktualisierung nicht mit der gewünschten Konfiguration oder mithilfe von Skip Matching starten.
- Bei jedem Aufruf zur Erstellung oder Aktualisierung Ihrer Auto-Scaling-Gruppe löst Amazon EC2 Auto Scaling den Systems-Manager-Parameter in der Startvorlage auf. Wenn Sie erweiterte Parameter oder höhere Durchsatzgrenzen verwenden, können die häufigen Aufrufe des Parameterspeichers (d. h. der `GetParameters`-Vorgang) Ihre Kosten für Systems Manager erhöhen, da Gebühren pro Parameterspeicher-API-Interaktion anfallen. Weitere Informationen finden Sie unter AWS Systems Manager Preise.

Startkonfigurationen

> **Important**
>
> Sie können `CreateLaunchConfiguration` nicht mit neuen Typen von Amazon-EC2-Instances aufrufen, die nach dem 31. Dezember 2022 veröffentlicht wurden. Darüber hinaus besteht für alle neuen Konten, die nach dem 1. Juni 2023 erstellt werden, nicht die Möglichkeit, neue Startkonfigurationen über die Konsole zu erstellen. In future werden neue Konten nicht mehr in der Lage sein, neue Startkonfigurationen mithilfe der Konsole, API, CLI und zu erstellen CloudFormation. Migrieren Sie zu Startvorlagen, um sicherzustellen, dass Sie weder jetzt noch in future neue Startkonfigurationen erstellen müssen. Informationen zum Migrieren Ihrer Auto-Scaling-Gruppen zu Startvorlagen finden Sie unter Migrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten.

Eine Startkonfiguration ist eine Instance-Konfigurationsvorlage, die eine Auto-Scaling-Gruppe zum Starten von EC2-Instances verwendet. Beim Erstellen einer Startkonfiguration geben Sie Informationen für die Instances an. Fügen Sie die ID des Amazon Machine Image (AMI), den Instance-Typ, ein Schlüsselpaar, mindestens eine Sicherheitsgruppe und eine Blockgerät-Zuweisung hinzu. Beim Start einer EC2-Instance geben Sie die gleichen Informationen an.

Sie können die Startkonfiguration für mehrere Auto-Scaling-Gruppen verwenden. Sie können jedoch nur eine Startkonfiguration pro Auto-Scaling-Gruppe angeben und diese nach der Erstellung nicht mehr ändern. Um die Startkonfiguration für eine Auto-Scaling-Gruppe zu ändern, müssen Sie eine Startkonfiguration erstellen und dann damit Ihre Auto-Scaling-Gruppe aktualisieren.

Inhalt

Erstellen einer Startkonfiguration

> **Important**
>
> Sie können `CreateLaunchConfiguration` nicht mit neuen Typen von Amazon-EC2-Instances aufrufen, die nach dem 31. Dezember 2022 veröffentlicht wurden. Darüber

hinaus besteht für alle neuen Konten, die nach dem 1. Juni 2023 erstellt werden, nicht die Möglichkeit, neue Startkonfigurationen über die Konsole zu erstellen. In future werden neue Konten nicht mehr in der Lage sein, neue Startkonfigurationen mithilfe der Konsole, API, CLI und zu erstellen CloudFormation. Migrieren Sie zu Startvorlagen, um sicherzustellen, dass Sie weder jetzt noch in future neue Startkonfigurationen erstellen müssen. Informationen zum Migrieren Ihrer Auto-Scaling-Gruppen zu Startvorlagen finden Sie unter Migrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten.

In diesem Thema wird beschrieben, wie Sie eine Startkonfiguration erstellen.

Nachdem Sie eine Startkonfiguration erstellt haben, können Sie sie nicht mehr ändern. Stattdessen müssen Sie eine neue Startkonfiguration erstellen.

Informationen zum Zuordnen einer neuen Startkonfiguration zu einer vorhandenen Auto Scaling Scaling-Gruppe finden Sie unterÄndern der Startkonfiguration für eine Auto-Scaling-Gruppe. Informationen zum Erstellen einer neuen Auto Scaling Scaling-Gruppe finden Sie unterEine Auto-Scaling-Gruppe mithilfe einer Startkonfiguration erstellen.

Inhalt

Erstellen einer Startkonfiguration

So erstellen Sie eine Startkonfiguration (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen Navigationsleiste Ihre AWS Region aus.
3. Wählen Sie im linken Navigationsbereich unter Auto Scaling Auto-Scaling-Gruppen aus.
4. Wählen Sie oben auf der Seite Startkonfigurationen aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Startkonfigurationen anzeigen aus, um zu bestätigen, dass Sie die Seite Startkonfigurationen aufrufen möchten.
5. Klicken Sie auf Erstellen einer Startkonfiguration und geben Sie einen Namen für die Startkonfiguration ein.

6. Wählen Sie für Amazon Machine Image (AMI) ein AMI aus. Um ein bestimmtes AMI zu finden, können Sie ein passendes AMI finden, sich die ID notieren und als Suchkriterium eingeben.

   So rufen Sie die ID des Amazon Linux 2-AMI ab:

   a. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.

   b. Wählen Sie im Navigationsbereich unter Instances Instances und dann Instances starten aus.

   c. Notieren Sie auf der Registerkarte Schnellstart der Seite Auswählen eines Amazon Machine Image die ID des AMI neben Amazon Linux 2-AMI.

7. Für Instance-Typ wählen Sie eine Hardware-Konfiguration für Ihre Instances aus.

8. Unter Zusätzliche Konfiguration achten Sie auf die folgenden Felder:

   a. (Optional) Für Kaufoption können Sie Spot-Instances anfordern auswählen, um Spot-Instances zum aktuellen Spot-Preis anzufordern, dessen Obergrenze der On-Demand-Preis ist. Optional können Sie einen Höchstpreis pro Spot-Instance-Stunde angeben.

   > **Note**
   >
   > Spot-Instances sind eine kostengünstige Wahl im Vergleich zu On-Demand-Instances, sofern Sie bei der Nutzung Ihrer Anwendungen zeitlich flexibel sind und Unterbrechungen verschmerzen können. Weitere Informationen finden Sie unter Spot-Instances für fehlertolerante und flexible Anwendungen anfordern.

   b. (Optional) Wählen Sie unter IAM-Instance-Profil eine Rolle aus, die mit den Instances verknüpft werden soll. Weitere Informationen finden Sie unter IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden.

   c. (Optional) Wählen Sie für Monitoring aus, ob die Instances Metrikdaten in Intervallen von 1 Minute an Amazon veröffentlichen können, CloudWatch indem Sie die detaillierte Überwachung aktivieren. Es fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Überwachung für Auto-Scaling-Instances konfigurieren.

   d. (Optional) Für Erweiterte Details, Benutzerdaten können Sie Benutzerdaten angeben, um eine Instance während des Starts zu konfigurieren, oder um nach dem Starten der Instance ein Konfigurationsskript auszuführen.

   e. (Optional) Für Erweiterte Details, IP-Adresstyp, wählen Sie aus, ob Sie den Instances der Gruppe eine Öffentliche IP-Adresse zuweisen. Wenn Sie keinen Wert festlegen, verwenden

Sie standardmäßig die automatische Zuweisung öffentlicher IP-Einstellungen der Subnetze, in denen Ihre Instances gestartet werden.

9. (Optional) Bei Speicher (Volumes) können Sie, wenn Sie keinen zusätzlichen Speicher benötigen, diesen Abschnitt überspringen. Wenn Sie Volumes angeben, die den Instances zusätzlich zu den von AMI angegebenen Volumes hinzugefügt werden sollen, wählen Sie Hinzufügen eines neuen Volumes aus. Wählen Sie dann die gewünschten Optionen und zugeordneten Werte für Geräte, Snapshot, Größe, Volume-Typ, IOPS, Durchsatz, Beim Beenden löschen und Verschlüsselt aus.
10. Für Sicherheitsgruppen erstellen Sie die Sicherheitsgruppe, die den Instances der Gruppe zugeordnet werden soll, oder wählen Sie sie aus. Wenn Sie die Option Erstellen einer neuen Sicherheitsgruppe ausgewählt lassen, wird eine Standard-SSH-Regel für Amazon-EC2-Instances konfiguriert, auf denen Linux ausgeführt wird. Für Amazon-EC2-Instances, die Windows ausführen, wird eine Standard-RDP-Rolle konfiguriert.
11. Für Schlüsselpaar (Login) wählen Sie eine Option unter Optionen für Schlüsselpaar aus.

    Wenn Sie ein Amazon EC2-Instance-Schlüsselpaar bereits konfiguriert haben, können Sie das Schlüsselpaar hier auswählen.

    Wenn Sie noch kein Amazon EC2-Instance-Schlüsselpaar haben, klicken Sie auf Create a new key pair (Ein neues Schlüsselpaar erstellen) und geben Sie einen wiedererkennbaren Namen ein. Wählen Sie Download Key Pair (Schlüsselpaar herunterladen) aus, um das Schlüsselpaar auf Ihrem Computer herunterzuladen.

    > **Important**
    >
    > Wählen Sie nicht Proceed without a key pair (Ohne Schlüsselpaar fortfahren) aus, wenn Sie eine Verbindung mit Ihrer Instance herstellen müssen.

12. Aktivieren Sie das Bestätigungskontrollkästchen und wählen Sie dann Create launch configuration (Startkonfiguration erstellen) aus.

Um eine Startkonfiguration aus einer vorhandenen Startkonfiguration (Konsole) zu erstellen

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen Navigationsleiste Ihre AWS Region aus.
3. Wählen Sie im linken Navigationsbereich unter Auto Scaling Auto-Scaling-Gruppen aus.

4. Wählen Sie oben auf der Seite Startkonfigurationen aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Startkonfigurationen anzeigen aus, um zu bestätigen, dass Sie die Seite Startkonfigurationen aufrufen möchten.
5. Wählen Sie die Startkonfiguration und anschließend Actions, Copy launch configuration aus. So wird eine mit dem Original identische Startkonfiguration (mit Ausnahme des Zusatzes "Kopie" im Namen) erstellt.
6. Bearbeiten Sie auf der Seite Copy Launch Configuration nach Bedarf die Konfigurationsoptionen und wählen Sie anschließend Create launch configuration aus.

Mithilfe der Befehlszeile erstellen Sie eine Startkonfiguration wie folgt:

Verwenden Sie einen der folgenden Befehle:

- create-launch-configuration (AWS CLI)
- Neu-AS LaunchConfiguration ()AWS Tools for Windows PowerShell

Konfigurieren der Instance-Metadaten-Optionen

Amazon EC2 Auto Scaling unterstützt die Konfiguration des Instance Metadata Service (IMDS) in Startkonfigurationen. Auf diese Weise haben Sie die Möglichkeit, Startkonfigurationen zu verwenden, um die Amazon EC2-Instances in Ihren Auto-Scaling-Gruppen so zu konfigurieren, dass Instance Metadata Service Version 2 (IMDSv2) erforderlich ist. Dies ist eine sitzungsorientierte Methode zum Anfordern von Instance-Metadaten. Weitere Informationen zu den Vorteilen von IMDSv2 finden Sie im AWS -Blog zu Enhancements to add defense in depth to the EC2 Instance Metadata Service.

Sie können IMDS so konfigurieren, dass IMDSv2 und IMDSv1 (Standardeinstellung) unterstützt oder IMDSv2 verwendet werden muss. Wenn Sie das AWS CLI oder eines der SDKs zur Konfiguration von IMDS verwenden, müssen Sie die neueste Version des AWS CLI oder des SDK verwenden, um die Verwendung von IMDSv2 zu erfordern.

Sie können Ihre Startkonfiguration wie folgt konfigurieren:

- Erzwingen der Verwendung von IMDSv2 beim Anfordern von Instance-Metadaten
- Angeben des PUT-Antwort-Hop-Limits
- Deaktivieren des Zugriffs auf Instance-Metadaten

Weitere Informationen zur Konfiguration des Instance-Metadaten-Service finden Sie im folgenden Thema: Konfiguration des Instance-Metadaten-Service im Amazon EC2 EC2-Benutzerhandbuch.

Konfigurieren Sie mit den folgenden Schritten IMDS-Optionen in einer Startkonfiguration. Nach dem Erstellen Ihrer Startkonfiguration können Sie diese mit Ihrer Auto-Scaling-Gruppe verknüpfen. Wenn Sie die Startkonfiguration einer vorhandenen Auto-Scaling-Gruppe zuordnen, wird die vorhandene Startkonfiguration von der Auto-Scaling-Gruppe getrennt, und vorhandene Instances müssen ersetzt werden, um die IMDS-Optionen zu verwenden, die Sie in der neuen Startkonfiguration angegeben haben. Weitere Informationen finden Sie unter Ändern der Startkonfiguration für eine Auto-Scaling-Gruppe.

So konfigurieren Sie IMDS in einer Startkonfiguration (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen Navigationsleiste Ihre AWS Region aus.
3. Wählen Sie im linken Navigationsbereich unter Auto Scaling Auto-Scaling-Gruppen aus.
4. Wählen Sie oben auf der Seite Startkonfigurationen aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Startkonfigurationen anzeigen aus, um zu bestätigen, dass Sie die Seite Startkonfigurationen aufrufen möchten.
5. Klicken Sie auf Erstellen einer Startkonfiguration und erstellen Sie die Startkonfiguration wie gewohnt. Enthalten sind die ID des Amazon Machine Image (AMI), der Instance-Typ und optional ein Schlüsselpaar, mindestens eine Sicherheitsgruppe und alle zusätzlichen EBS-Volumes oder Instance-Speicher-Volumes für Ihre Instances.
6. Um Instance-Metadatenoptionen für alle Instances zu konfigurieren, die dieser Startkonfiguration zugeordnet sind, finden Sie bei Zusätzliche Konfiguration unter Erweiterte Details wie folgt:

    a. Für Metadata accessible (Metadaten zugänglich) wählen Sie aus, ob der Zugriff auf den HTTP-Endpunkt des Instance-Metadatenservices aktiviert oder deaktiviert werden soll. Standardmäßig ist der HTTP-Endpunkt aktiviert. Wenn Sie den Endpunkt deaktivieren, wird der Zugriff auf Ihre Instance-Metadaten deaktiviert. Sie können die Bedingung angeben, dass IMDSv2 nur erforderlich ist, wenn der HTTP-Endpunkt aktiviert ist.

    b. Für Metadata version (Metadatenversion) können Sie auswählen, dass die Verwendung von Instance-Metadatenservice Version 2 (IMDSv2) beim Anfordern von Instance-Metadaten erforderlich ist. Wenn Sie keinen Wert angeben, unterstützt standardmäßig IMDSv1 und IMDSv2.

c. Für Metadata token response hop limit (Antwort-Hop-Limit des Metadaten-Tokens) können Sie die zulässige Anzahl von Netzwerk-Hops für das Metadaten-Token festlegen. Wenn Sie keinen Wert angeben, wird der Standard auf 1 festgelegt.

7. Wählen Sie danach Erstellen einer Startkonfiguration aus.

Die Verwendung von IMDSv2 in einer Startkonfiguration mit dem Befehl AWS CLI anfordern

Verwenden Sie den Befehl create-launch-configuration mit der auf `HttpTokens=required` eingestellten Option `--metadata-options`. Wenn Sie einen Wert für `HttpTokens` angeben, müssen Sie auch `HttpEndpoint` aktivieren. Da der sichere Token-Header für Metadaten-Abrufanforderungen auf „erforderlich“ festgelegt ist, muss beim Anfordern von Instance-Metadaten in der Instance IMDSv2 verwendet werden.

```
aws autoscaling create-launch-configuration \
  --launch-configuration-name my-lc-with-imdsv2 \
  --image-id ami-01e24be29428c15b2 \
  --instance-type t2.micro \
      ...
  --metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

So deaktivieren Sie den Zugriff auf Instance-Metadaten

Verwenden Sie die folgenden create-launch-configuration-Befehl, um den Zugriff auf Instance-Metadaten zu deaktivieren. Sie können den Zugriff zu einem späteren Zeitpunkt mit dem Befehl modify-instance-metadata-options wieder aktivieren.

```
aws autoscaling create-launch-configuration \
  --launch-configuration-name my-lc-with-imds-disabled \
  --image-id ami-01e24be29428c15b2 \
  --instance-type t2.micro \
      ...
  --metadata-options "HttpEndpoint=disabled"
```

Erstellen einer Startkonfiguration aus einer EC2-Instance

Sie haben auch die Möglichkeit, eine Startkonfiguration mit den Attributen einer laufenden EC2-Instance zu erstellen.

Das Erstellen einer Startkonfiguration von Grund auf ist nicht dasselbe wie das Erstellen einer Startkonfiguration aus einer vorhandenen EC2-Instance. Bei der Erstellung einer Startkonfiguration von Grund auf geben Sie Image-ID, Instance-Typ, optionale Ressourcen (z. B. Speichergeräte) und optionale Einstellungen (z. B. Überwachung) an. Beim Erstellen einer Startkonfiguration aus einer ausgeführten Instance übernimmt Amazon EC2 Auto Scaling die Attribute für die Startkonfiguration von der angegebenen Instance. Attribute werden auch aus der Blockgerät-Zuweisung für das AMI abgeleitet, von dem die Instance gestartet wurde; dabei werden alle weiteren Blockgeräte, die nach dem Start hinzugefügt wurden, ignoriert.

Bei der Erstellung einer Startkonfiguration mithilfe einer laufenden Instance können Sie die folgenden Attribute überschreiben, indem Sie sie als Teil derselben Anforderung markieren: AMI, Blockgeräte, Schlüsselpaar, Instance-Profil, Instance-Typ, Kernel, Instance-Überwachung, Platzierungs-Tenancy, Ramdisk, Sicherheitsgruppen, (max) Spot-Preis, Benutzerdaten, ob die Instance über eine öffentliche IP-Adresse verfügt und ob die Instance für EBS optimiert ist.

> **Note**
>
> Weist die angegebene Instance Eigenschaften auf, die derzeit von Startkonfigurationen nicht unterstützt werden, dann sind die von der Auto-Scaling-Gruppe gestarteten Instances möglicherweise nicht identisch mit der ursprünglichen EC2-Instance.

> **Important**
>
> Das AMI zum Starten der angegebenen Instance muss noch vorhanden sein.

Themen

Erstellen einer Startkonfiguration aus einer EC2-Instance (AWS CLI)

Verwenden Sie den folgenden create-launch-configuration-Befehl zum Erstellen einer Startkonfiguration aus einer Instance mit den Attributen der Instance. Alle Blockgeräte, die nach dem Start hinzugefügt wurden, werden ignoriert.

```
aws autoscaling create-launch-configuration --launch-configuration-name my-lc-from-instance --instance-id i-a8e09d9c
```

Verwenden Sie den folgenden describe-launch-configurations-Befehl, um die Startkonfiguration zu beschreiben und zu überprüfen, ob die Attribute der Instance übereinstimmen:

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-lc-from-instance
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "LaunchConfigurations": [
        {
            "UserData": null,
            "EbsOptimized": false,
            "LaunchConfigurationARN": "arn",
            "InstanceMonitoring": {
                "Enabled": false
            },
            "ImageId": "ami-05355a6c",
            "CreatedTime": "2014-12-29T16:14:50.382Z",
            "BlockDeviceMappings": [],
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                "sg-8422d1eb"
            ],
            "LaunchConfigurationName": "my-lc-from-instance",
            "KernelId": "null",
            "RamdiskId": null,
            "InstanceType": "t1.micro",
            "AssociatePublicIpAddress": true
        }
    ]
}
```

Erstellen einer Startkonfiguration aus einer Instance und Überschreiben der Blockgeräte (AWS CLI)

Standardmäßig verwendet Amazon EC2 Auto Scaling die Attribute der angegebenen EC2-Instance zum Erstellen der Startkonfiguration. Die Blockgeräte kommen jedoch aus dem AMI, das zum Starten

der Instance verwendet wird, nicht aus der Instance. Überschreiben Sie die Blockgerät-Zuweisung der Startkonfiguration, um ihr Blockgeräte hinzuzufügen.

Verwenden Sie den folgenden create-launch-configuration-Befehl, um eine Startkonfiguration aus einer EC2-Instance und mit einer benutzerdefinierten Blockgerät-Zuweisung zu erstellen:

```
aws autoscaling create-launch-configuration --launch-configuration-name my-lc-from-instance-bdm --instance-id i-a8e09d9c \
  --block-device-mappings "[{\"DeviceName\":\"/dev/sda1\",\"Ebs\":{\"SnapshotId\":\"snap-3decf207\"}},{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"SnapshotId\":\"snap-eed6ac86\"}}]"
```

Verwenden Sie den folgenden describe-launch-configurations-Befehl, um die Startkonfiguration zu beschreiben und zu überprüfen, ob sie die benutzerdefinierte Blockgerät-Zuweisung verwendet:

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-lc-from-instance-bdm
```

Die folgende Beispielantwort beschreibt die Startkonfiguration:

```
{
    "LaunchConfigurations": [
        {
            "UserData": null,
            "EbsOptimized": false,
            "LaunchConfigurationARN": "arn",
            "InstanceMonitoring": {
                "Enabled": false
            },
            "ImageId": "ami-c49c0dac",
            "CreatedTime": "2015-01-07T14:51:26.065Z",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "Ebs": {
                        "SnapshotId": "snap-3decf207"
                    }
                },
                {
                    "DeviceName": "/dev/sdf",
                    "Ebs": {
                        "SnapshotId": "snap-eed6ac86"
```

```
                    }
                }
            ],
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                "sg-8637d3e3"
            ],
            "LaunchConfigurationName": "my-lc-from-instance-bdm",
            "KernelId": null,
            "RamdiskId": null,
            "InstanceType": "t1.micro",
            "AssociatePublicIpAddress": true
        }
    ]
}
```

Erstellen einer Startkonfiguration und Überschreiben des Instance-Typs (AWS CLI)

Standardmäßig verwendet Amazon EC2 Auto Scaling die Attribute der angegebenen EC2-Instance zum Erstellen der Startkonfiguration. Je nach Ihren Anforderungen möchten Sie vielleicht Attribute aus der Instance überschreiben und die benötigten Werte verwenden. Sie können beispielsweise den Instance-Typ überschreiben.

Verwenden Sie den folgenden create-launch-configuration-Befehl, um eine Startkonfiguration aus einer EC2-Instance und mit einem anderen Instance-Typ (z. B. `t2.medium`) als die Instance (z. B. `t2.micro`) zu erstellen:

```
aws autoscaling create-launch-configuration --launch-configuration-name my-lc-from-instance-changetype \
  --instance-id i-a8e09d9c --instance-type t2.medium
```

Verwenden Sie den folgenden describe-launch-configurations-Befehl, um die Startkonfiguration zu beschreiben und zu überprüfen, ob der Instance-Typ überschrieben wurde:

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-lc-from-instance-changetype
```

Die folgende Beispielantwort beschreibt die Startkonfiguration:

```
{
```

```
    "LaunchConfigurations": [
        {
            "UserData": null,
            "EbsOptimized": false,
            "LaunchConfigurationARN": "arn",
            "InstanceMonitoring": {
                "Enabled": false
            },
            "ImageId": "ami-05355a6c",
            "CreatedTime": "2014-12-29T16:14:50.382Z",
            "BlockDeviceMappings": [],
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                "sg-8422d1eb"
            ],
            "LaunchConfigurationName": "my-lc-from-instance-changetype",
            "KernelId": "null",
            "RamdiskId": null,
            "InstanceType": "t2.medium",
            "AssociatePublicIpAddress": true
        }
    ]
}
```

Ändern der Startkonfiguration für eine Auto-Scaling-Gruppe

> **Important**
>
> Wir stellen Informationen zu Startkonfigurationen für Kunden bereit, die noch nicht von Startkonfigurationen zu Startvorlagen migriert sind. Informationen zum Migrieren Ihrer Auto-Scaling-Gruppen zu Startvorlagen finden Sie unter Migrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten.

In diesem Thema wird beschrieben, wie Sie Ihrer Auto Scaling Scaling-Gruppe eine andere Startkonfiguration zuordnen.

Nachdem Sie die Startkonfiguration geändert haben, werden alle neuen Instances mit den neuen Konfigurationsoptionen gestartet, bestehende Instances sind davon jedoch nicht betroffen. Weitere Informationen finden Sie unter Aktualisieren von Auto-Scaling-Instances.

So ändern Sie die Startkonfiguration einer Auto-Scaling-Gruppe (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im linken Navigationsbereich unter Auto Scaling Auto-Scaling-Gruppen aus.
3. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
4. Wählen Sie auf der Registerkarte Details die Option Konfiguration starten, Bearbeiten aus.
5. Wählen Sie unter Startkonfiguration die Startkonfiguration aus.
6. Wählen Sie Aktualisieren aus, wenn Sie fertig sind.

So ändern Sie die Startkonfiguration für eine Auto Scaling Scaling-Gruppe über die Befehlszeile

Verwenden Sie einen der folgenden Befehle:

- update-auto-scaling-group (AWS CLI)
- Als AutoScaling Gruppe aktualisieren ()AWS Tools for Windows PowerShell

Auto-Scaling-Gruppen

> **Note**
>
> Wenn Sie mit Auto Scaling-Gruppen noch nicht vertraut sind, führen Sie zunächst die Schritte im Tutorial Erstellen Sie Ihre erste Auto Scaling Scaling-Gruppe durch und sehen Sie, wie eine Auto Scaling Scaling-Gruppe reagiert, wenn eine Instance in der Gruppe beendet wird.

Eine Auto-Scaling-Gruppe enthält eine Sammlung von EC2-Instances, die zur automatischen Skalierung und Verwaltung als logische Gruppierung behandelt werden. Eine Auto-Scaling-Gruppe ermöglicht Ihnen außerdem die Verwendung von Amazon EC2 Auto Scaling-Funktionen wie Ersetzungen im Zuge von Zustandsprüfungen und Skalierungsrichtlinien. Sowohl die Aufrechterhaltung der Anzahl von Instances in einer Auto-Scaling-Gruppe und die automatische Skalierung sind die wichtigsten Funktionen des Amazon EC2 Auto Scaling-Services.

Die Größe einer Auto-Scaling-Gruppe richtet sich nach der Anzahl der Instances, die Sie als die gewünschte Kapazität einstellen. Sie können seine Größe an den Bedarf anpassen, entweder manuell oder durch automatische Skalierung.

Eine Auto-Scaling-Gruppe beginnt mit dem Start einer ausreichenden Anzahl von EC2-Instances, um die gewünschte Kapazität zu erfüllen. Sie erhält diese Anzahl der Instances aufrecht, indem die Instances der Gruppe regelmäßigen Zustandsprüfungen unterzogen werden. Die Auto-Scaling-Gruppe verwendet weiterhin eine feste Anzahl von Instances, selbst wenn eine Instance fehlerhaft wird. Wird eine Instance fehlerhaft, beendet die Gruppe die fehlerhafte Instance und startet eine andere Instance, um sie zu ersetzen. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Sie können Skalierungsrichtlinien zur dynamischen Erhöhung bzw. Verringerung der Anzahl an Instances in der Gruppe verwenden, um wechselnden Bedingungen entgegenzukommen. Wenn die Skalierungsrichtlinie verwendet wird, passt die Auto-Scaling-Gruppe die gewünschte Kapazität der Gruppe zwischen den von Ihnen angegebenen minimalen und maximalen Kapazitätswerten an und startet bzw. beendet die Instances je nach Bedarf. Sie können auch nach einem Zeitplan skalieren. Weitere Informationen finden Sie unter Wählen Sie Ihre Skalierungsmethode aus.

Beim Erstellen einer Auto-Scaling-Gruppe können Sie entweder On-Demand-Instances, Spot-Instances oder beides starten. Sie können nur dann mehrere Kaufoptionen für Ihre Auto-Scaling-

Gruppe angeben, wenn Sie die Gruppe so konfigurieren, dass sie eine Startvorlage verwendet. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.

Spot Instances ermöglichen den Zugriff auf ungenutzte EC2-Kapazitäten zu deutlich günstigeren Preisen (im Vergleich zu den On-Demand-Preisen). Weitere Informationen finden Sie unter Amazon-EC2-Spot-Instances. Es gibt wesentliche Unterschiede zwischen Spot-Instances und On-Demand-Instances:

- Der Preis für Spot-Instances variiert je nach Bedarf
- Amazon EC2 kann eine einzelne Spot-Instance beenden, wenn sich die Verfügbarkeit oder der Preis für Spot-Instances ändert.

Wird eine Spot-Instance beendet, versucht die Auto-Scaling-Gruppe, eine Ersatz-Instance zu starten, um die gewünschte Kapazität für die Gruppe aufrechtzuerhalten.

Wenn Instances gestartet werden, wird, wenn Sie mehrere Availability Zones angegeben haben, die gewünschte Kapazität über all diese Availability Zones verteilt. Wenn eine Skalierungsaktion erfolgt, behält Amazon EC2 Auto Scaling automatisch das Gleichgewicht über alle von Ihnen angegebenen Availability Zones bei.

Inhalt

Erstellen Sie Auto-Scaling-Gruppen mit Startvorlagen

Wenn Sie eine Startvorlage erstellt haben, können Sie eine Auto-Scaling-Gruppe erstellen, die eine Startvorlage als Konfigurationsvorlage für ihre EC2-Instances verwendet. Die Startvorlage gibt Informationen wie die AMI-ID, den Instance-Typ, das Schlüsselpaar, Sicherheitsgruppen und die Blockgerät-Zuweisung für Ihre Instances an. Weitere Informationen zum Erstellen von Startvorlagen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.

Sie müssen über IAM-Berechtigungen verfügen, um eine Auto-Scaling-Gruppe zu erstellen. Sie müssen auch über ausreichende Berechtigungen verfügen, um die serviceverknüpfte Rolle zu erstellen, die Amazon EC2 Auto Scaling verwendet, um Aktionen in Ihrem Namen durchzuführen, falls sie noch nicht existiert. Beispiele für IAM-Richtlinien, die ein Administrator als Referenz für die Erteilung von Berechtigungen verwenden kann, finden Sie unter Beispiele für identitätsbasierte Richtlinien und Support für Startvorlagen.

Inhalt

Erstellen einer Auto-Scaling-Gruppe mithilfe einer Startvorlage

Wenn Sie eine Auto-Scaling-Gruppe erstellen, müssen Sie die notwendigen Informationen zur Konfiguration der Amazon EC2-Instances, die Availability Zones und VPC-Subnetze für die Instances, die gewünschte Kapazität sowie die minimalen und maximalen Kapazitätsgrenzen angeben.

Um Amazon EC2-Instances zu konfigurieren, die von Ihrer Auto-Scaling-Gruppe gestartet werden, können Sie eine Startvorlage oder eine Startkonfiguration angeben. Das folgende Verfahren veranschaulicht das Erstellen einer Auto-Scaling-Gruppe mithilfe einer Startvorlage.

Voraussetzungen

- Sie müssen eine Startvorlage erstellt haben. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.

Erstellen einer Auto-Scaling-Gruppe mithilfe einer Startvorlage (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm dieselbe aus, AWS-Region die Sie bei der Erstellung der Startvorlage verwendet haben.
3. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
4. Gehen Sie auf der Seite Choose launch template or configuration (Startvorlage oder Konfiguration auswählen) folgendermaßen vor:

   a. Für Auto-Scaling-Gruppenname geben Sie einen Namen für Ihre Auto-Scaling-Gruppe ein.

   b. Wählen Sie für Launch template (Startvorlage) eine vorhandene Startvorlage aus.

   c. Wählen Sie unter Launch template version (Version der Startvorlage) aus, ob die Auto-Scaling-Gruppe beim horizontalen Skalieren nach oben die standardmäßige, die neueste oder eine bestimmte Version der Startvorlage verwenden soll.

   d. Stellen Sie sicher, dass Ihre Startvorlage alle Optionen unterstützt, die Sie verwenden möchten, und wählen Sie dann Next (Weiter) aus.
5. Wenn Sie auf der Seite Instance-Startoptionen auswählen nicht mehrere Instance-Typen verwenden, können Sie den Abschnitt Anforderungen an den Instance-Typ überspringen, um den EC2-Instance-Typ zu verwenden, der in der Startvorlage angegeben ist.

   Um mehrere Instance-Typen zu verwenden, siehe Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.
6. Wählen Sie unter Netzwerk für VPC eine VPC. Die Auto-Scaling-Gruppe muss in derselben VPC erstellt werden wie die Sicherheitsgruppe, die Sie in Ihrer Startvorlage angegeben haben.
7. Für Availability Zones und Subnets (Subnetze) wählen Sie ein oder mehrere Subnetze in der angegebenen VPC aus. Verwenden Sie Subnetze in mehreren Availability Zones, um eine hohe Verfügbarkeit zu erzielen. Weitere Informationen finden Sie unter Überlegungen bei der Auswahl von VPC-Subnetzen.
8. Wenn Sie eine Startvorlage mit einem bestimmten Instance-Typ erstellt haben, können Sie mit dem nächsten Schritt fortfahren, um eine Auto-Scaling-Gruppe zu erstellen, die den Instance-Typ in der Startvorlage verwendet.

   Alternativ können Sie auch die Option Startvorlage überschreiben wählen, wenn in Ihrer Startvorlage kein Instance-Typ angegeben ist oder wenn Sie mehrere Instance-Typen für die

automatische Skalierung verwenden möchten. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.

9. Wählen Sie Next (Weiter) aus, um mit dem nächsten Schritt fortzufahren.

   Oder akzeptieren Sie die weiteren Standardwerte, und klicken Sie dann auf Skip to review (Mit Prüfen fortfahren).

10. (Optional) Konfigurieren Sie auf der Seite Konfigurieren von erweiterten Optionen die folgenden Optionen und wählen Sie Weiter:

    a. Wählen Sie unter Zusätzliche Einstellungen, Überwachung, aus, ob die Erfassung von CloudWatch Gruppenmetriken aktiviert werden soll. Diese Metriken liefern Messwerte, die Indikatoren für ein potenzielles Problem sein können, wie z.B. die Anzahl der abgebrochenen Instances oder die Anzahl der ausstehenden Instances. Weitere Informationen finden Sie unter Überwachen Sie CloudWatch Metriken für Ihre Auto Scaling Scaling-Gruppen und -Instances.

    b. Wählen Sie unter Standardinstanzaufwärmen aktivieren diese Option und wählen Sie die Aufwärmzeit für Ihre Anwendung aus. Wenn Sie eine Auto Scaling-Gruppe mit einer Skalierungsrichtlinie erstellen, verbessert die Standard-Instance-Aufwärmfunktion die CloudWatch Amazon-Metriken, die für die dynamische Skalierung verwendet werden. Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

11. Konfigurieren Sie auf der Seite Configure group size and scaling policies (Gruppengröße und Skalierungsrichtlinien konfigurieren) die folgenden Optionen, und wählen Sie dann Next (Weiter):

    a. Geben Sie unter Gruppengröße für Gewünschte Kapazität die anfängliche Anzahl von Instances ein, die gestartet werden sollen.

    b. Wenn im Abschnitt Skalierung unter Skalierungslimits Ihr neuer Wert für die gewünschte Kapazität größer als die gewünschte Mindestkapazität und die gewünschte Höchstkapazität ist, wird die gewünschte Höchstkapazität automatisch auf den neuen Wert für die gewünschte Kapazität erhöht. Sie können die Limits bei Bedarf ändern. Weitere Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.

    c. Wählen Sie für Automatische Skalierung aus, ob Sie eine Skalierungsrichtlinie für die Zielverfolgung erstellen möchten. Sie können diese Richtlinie auch erstellen, nachdem Sie Ihre Auto-Scaling-Gruppe erstellt haben.

Wenn Sie sich für die Skalierungsrichtlinie für die Zielverfolgung entscheiden, befolgen Sie die Anweisungen unter Erstellen einer Zielverfolgungs-Skalierungsrichtlinie, um die Richtlinie zu erstellen.

d. Wählen Sie unter Instance-Wartungsrichtlinie aus, ob Sie eine Instance-Wartungsrichtlinie erstellen möchten. Sie können diese Richtlinie auch erstellen, nachdem Sie Ihre Auto-Scaling-Gruppe erstellt haben. Befolgen Sie zum Erstellen der Richtlinie die Anweisungen unter Festlegen einer Instance-Wartungsrichtlinie.

e. Wählen Sie unter Instance scale-in protection (Instance-Skalierungsschutz), ob der Instance-Skalierungsschutz aktiviert werden soll. Weitere Informationen finden Sie unter Instance-Abskalierungsschutz verwenden.

12. (Optional) Um Benachrichtigungen zu erhalten, konfigurieren Sie für Add notification (Benachrichtigungen hinzufügen) die Benachrichtigung und wählen Sie anschließend Next (Weiter) aus. Weitere Informationen finden Sie unter Amazon SNS-Benachrichtigungsoptionen für Amazon EC2 Auto Scaling.
13. (Optional) Um Tags hinzuzufügen, wählen Sie Add tag (Tag hinzufügen) aus, geben Sie für jedes Tag einen Tag-Schlüssel und einen Wert an und wählen Sie anschließend Next (Weiter) aus. Weitere Informationen finden Sie unter Tagging von Auto-Scaling-Gruppen und Instances.
14. Wählen Sie auf der Seite Review (Prüfen) Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Erstellen Sie wie folgt eine Auto-Scaling-Gruppe über die Befehlszeile:

Verwenden Sie einen der folgenden Befehle:

- create-auto-scaling-group (AWS CLI)
- AutoScalingGroupNeu-AS ()AWS Tools for Windows PowerShell

Erstellen einer Auto-Scaling-Gruppe mithilfe des Amazon EC2-Startassistenten

Das folgende Verfahren zeigt, wie Sie mit dem Assistenten zum Starten von Instances in der Amazon EC2-Konsole eine Auto-Scaling-Gruppe erstellen. Mit dieser Option wird eine Startvorlage automatisch mit bestimmten Konfigurationsdetails aus dem Assistenten zum Starten von Instances gefüllt.

> **Note**
>
> Der Assistent füllt die Auto-Scaling-Gruppe nicht mit der von Ihnen angegebenen Anzahl von Instances, sondern nur die Startvorlage mit der Amazon Machine Image (AMI)-ID und dem Instance-Typ. Verwenden Sie den Assistenten zum Create Auto Scaling group (Auto-Scaling-Gruppe erstellen), um die Anzahl der zu startenden Instances anzugeben.
> Ein AMI enthält die für die Konfiguration einer Instance erforderlichen Informationen. Sie können mehrere Instances aus einem einzigen AMI starten, wenn Sie mehrere Instances mit derselben Konfiguration benötigen. Wir empfehlen die Verwendung eines benutzerdefinierten AMI, auf dem Ihre Anwendung bereits installiert ist, um zu vermeiden, dass Ihre Instances beendet werden, wenn Sie eine Instance neu starten, die zu einer Auto-Scaling-Gruppe gehört. Um ein benutzerdefiniertes AMI mit Amazon EC2 Auto Scaling zu verwenden, müssen Sie zunächst Ihr AMI aus einer benutzerdefinierten Instance erstellen und dann das AMI verwenden, um eine Startvorlage für Ihre Auto-Scaling-Gruppe zu erstellen.

Voraussetzungen

- Sie müssen dort, AWS-Region wo Sie die Auto Scaling Scaling-Gruppe erstellen möchten, ein benutzerdefiniertes AMI erstellt haben. Weitere Informationen finden Sie unter Create an AMI im Amazon EC2 EC2-Benutzerhandbuch.

Verwenden Sie ein benutzerdefiniertes AMI als Vorlage

In diesem Abschnitt verwenden Sie den Launch Wizard von Amazon EC2, um eine Startvorlage automatisch mit Ihrem benutzerdefinierten AMI auszufüllen. Wenn Sie die Startvorlage von Grund auf neu einrichten möchten oder weitere Informationen zu den Parametern benötigen, die Sie für Ihre Startvorlage konfigurieren können, lesen Sie So erstellen Sie eine Startvorlage (Konsole).

So verwenden Sie ein benutzerdefiniertes AMI als Vorlage

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. In der Navigationsleiste oben auf dem Bildschirm AWS-Region wird der aktuelle Wert angezeigt. Wählen Sie eine Region aus, in der Sie Ihre Auto-Scaling-Gruppe starten möchten.
3. Wählen Sie im Navigationsbereich Instances aus.
4. Wählen Sie Launch instance (Instance starten) aus und gehen Sie folgendermaßen vor:

a. Lassen Sie unter Name and tags (Name und Tags) Name (Name) leer. Der Name ist nicht Teil der Daten, die zum Erstellen einer Startvorlage verwendet werden.

b. Wählen Sie unter Application and OS Images (Amazon Machine Image) (Anwendungs- und Betriebssystem-Images (Amazon Machine Image)) Browse more AMIs (Weitere AMIs durchsuchen), um den vollständigen AMI-Katalog zu durchsuchen.

c. Wählen My AMIs (Meine AMIs), suchen Sie das AMI, das Sie zuvor erstellt haben, und wählen Sie anschließend Select (Auswählen) aus.

d. Wählen Sie unter EC2 Instance Type (EC2-Instance-Typ) einen Instance-Typ aus.

> **Note**
>
> Wählen Sie denselben Instance-Typ, den Sie bei der Erstellung des AMI verwendet haben, oder einen leistungsfähigeren.

e. Geben Sie auf der rechten Seite des Bildschirms unter Summary (Übersicht), für Number of instances (Anzahl der Instances) eine beliebige Zahl ein. Die Zahl, die Sie hier eingeben, ist nicht wichtig. Sie geben die Anzahl der Instances an, die gestartet werden sollen, wenn Sie die Auto-Scaling-Gruppe erstellen.

   Unter dem Feld Number of instances (Anzahl der Instances) wird eine Meldung angezeigt, die besagt When launching more than 1 instance, consider EC2 Auto Scaling (Beim Starten von mehr als einer Instance EC2 Auto Scaling berücksichtigen).

f. Wählen Sie den Hyperlinktext consider EC Auto Scaling (EC2 Auto Scaling berücksichtigen).

g. Wählen Sie im Bestätigungsdialog Launch into Auto Scaling Group (In Auto-Scaling-Gruppe starten) Continue (Weiter), um zur Seite Create launch template (Startvorlage erstellen) zu gelangen, auf der das AMI und der Instance-Typ, die Sie im Assistenten zum Starten der Instance ausgewählt haben, bereits eingetragen sind.

Nachdem Sie Continue (Weiter) gewählt haben, öffnet sich die Seite Create launch template (Startvorlage erstellen). Gehen Sie folgendermaßen vor, um die Erstellung einer Startvorlage abzuschließen.

Eine Startvorlage erstellen

1. Geben Sie unter Launch template name and description (Name und Beschreibung der Startvorlage) einen Namen und eine Beschreibung für die neue Startvorlage ein.

2. (Optional) Wählen Sie unter Key pair (login) (Schlüsselpaar (Login)) für Key pair name (Schlüsselpaar-Name) den Namen des zuvor erstellten Schlüsselpaars, das Sie für die Verbindung zu Instances verwenden möchten, z. B. über SSH.
3. (Optional) Wählen Sie unter Network settings (Netzwerkeinstellungen) für Security groups (Sicherheitsgruppen) eine oder mehrere zuvor erstellte security groups (Sicherheitsgruppen).
4. (Optional) Aktualisieren Sie unter Configure storage (Speicher konfigurieren) die Speicherkonfiguration. Die Standardspeicherkonfiguration wird vom AMI und dem Instance-Typ bestimmt.
5. Wenn Sie mit der Konfiguration der Startvorlage fertig sind, wählen Sie Startvorlage erstellen.
6. Wählen Sie auf der Bestätigungsseite Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Erstellen einer Auto-Scaling-Gruppe

> **Note**
>
> Der Rest dieses Themas beschreibt das grundlegende Verfahren zur Erstellung einer Auto-Scaling-Gruppe. Eine Beschreibung der Parameter, die Sie für Ihre Auto-Scaling-Gruppe konfigurieren können, finden Sie unter Erstellen einer Auto-Scaling-Gruppe mithilfe einer Startvorlage.

Nachdem Sie Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) gewählt haben, öffnet sich der Assistent Create Auto Scaling group (Auto-Scaling-Gruppe erstellen). Gehen Sie folgendermaßen vor, um eine Auto-Scaling-Gruppe zu erstellen.

So erstellen Sie eine Auto Scaling-Gruppe

1. Geben Sie auf der Seite Startvorlage oder Konfiguration auswählen einen Namen für die Auto-Scaling-Gruppe ein.
2. Die Startvorlage, die Sie erstellt haben, ist bereits für Sie ausgewählt.

   Wählen Sie unter Launch template version (Version der Startvorlage) aus, ob die Auto-Scaling-Gruppe beim horizontalen Skalieren nach oben die standardmäßige, die neueste oder eine bestimmte Version der Startvorlage verwenden soll.
3. Wählen Sie Next (Weiter) aus, um mit dem nächsten Schritt fortzufahren.

4. Wenn Sie auf der Seite Instance-Startoptionen auswählen nicht mehrere Instance-Typen verwenden, können Sie den Abschnitt Anforderungen an den Instance-Typ überspringen, um den EC2-Instance-Typ zu verwenden, der in der Startvorlage angegeben ist.

   Um mehrere Instance-Typen zu verwenden, siehe Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.
5. Wählen Sie unter Netzwerk für VPC eine VPC. Die Auto-Scaling-Gruppe muss in derselben VPC erstellt werden wie die Sicherheitsgruppe, die Sie in Ihrer Startvorlage angegeben haben.

   > **Tip**
   >
   > Wenn Sie in Ihrer Startvorlage keine Sicherheitsgruppe angegeben haben, werden Ihre Instances mit einer Standardsicherheitsgruppe aus der von Ihnen angegebenen VPC gestartet. Standardmäßig lässt diese Sicherheitsgruppe eingehenden Datenverkehr von externen Netzwerken nicht zu.
6. Für Availability Zones und Subnets (Subnetze) wählen Sie ein oder mehrere Subnetze in der angegebenen VPC aus.
7. Wählen Sie zweimal Next (Weiter), um zur Seite Configure group size and scaling policies (Gruppengröße und Skalierungsrichtlinien konfigurieren) zu gelangen.
8. Legen Sie unter Gruppengröße die gewünschte Kapazität fest (anfängliche Anzahl von Instances, die sofort nach Erstellen der Auto-Scaling-Gruppe gestartet werden sollen).
9. Wenn im Abschnitt Skalierung unter Skalierungslimits Ihr neuer Wert für die gewünschte Kapazität größer als die gewünschte Mindestkapazität und die gewünschte Höchstkapazität ist, wird die gewünschte Höchstkapazität automatisch auf den neuen Wert für die gewünschte Kapazität erhöht. Sie können die Limits bei Bedarf ändern. Weitere Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.
10. Wählen Sie Skip to review (Mit Prüfen fortfahren) aus.
11. Wählen Sie auf der Seite Review (Prüfen) Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Nächste Schritte

Sie können überprüfen, ob die Auto-Scaling-Gruppe korrekt erstellt wurde, indem Sie sich den Aktivitätsverlauf ansehen. Auf der Registerkarte Activity (Aktivität) wird unter Activity history (Aktivitätsverlauf) in der Spalte Status angezeigt, ob Ihre Auto-Scaling-Gruppe-Instances erfolgreich

gestartet hat. Wenn die Instances nicht starten oder zwar starten, dann aber sofort abbrechen, lesen Sie die folgenden Themen zu möglichen Ursachen und Lösungen:

- Fehlersuche bei Amazon EC2 Auto Scaling: Startfehler von EC2-Instance
- Fehlersuche bei Amazon EC2 Auto Scaling: AMI-Probleme
- Fehlerbehebung bei fehlerhaften Instances in Amazon EC2 Auto Scaling

Sie können nun ein Load Balancer in derselben Region wie Ihre Auto-Scaling-Gruppe einrichten, falls gewünscht. Weitere Informationen finden Sie unter Um den Datenverkehr über die Instances in Ihrer Auto-Scaling-Gruppe zu verteilen, verwenden Sie Elastic-Load-Balancing..

Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen

Sie können eine Flotte von On-Demand-Instances und Spot-Instances innerhalb einer einzigen Auto-Scaling-Gruppe starten und automatisch skalieren. Zusätzlich zum Erhalt von Rabatten für die Verwendung von Spot-Instances können Sie mit Reserved Instances oder einem Savings Plan Rabatte auf die regulären On-Demand-Instance-Preise erhalten. Diese Faktoren helfen Ihnen, Ihre Kosteneinsparungen für EC2-Instances zu optimieren und die gewünschte Skalierung und Leistung für Ihre Anwendung zu erhalten.

Spot-Instances sind Kapazitätsreserven, die im Vergleich zum EC2-On-Demand-Preis stark reduziert erhältlich sind. Spot-Instances sind eine kostengünstige Wahl, sofern Sie bei der Ausführung Ihrer Anwendungen zeitlich flexibel sind und Unterbrechungen verschmerzen können. Sie können für verschiedene fehlertolerante und flexible Anwendungen verwendet werden. Beispiele hierfür sind statuslose Webserver, API-Endpunkte, Big Data- und Analyseanwendungen, containerisierte Workloads, CI/CD-Pipelines, Hochleistungsrechnen und Hochdurchsatzrechnen (HPC/HTC), Rendering-Workloads und andere flexible Workloads.

Weitere Informationen finden Sie unter Kaufoptionen für Instances im Amazon EC2 EC2-Benutzerhandbuch.

Themen

- Übersicht über die Einrichtung
- Zuweisungsstrategien
- Erstellen einer gemischten Instances-Gruppe mit attributbasierter Auswahl des Instance-Typs
- Erstellen Sie eine Gruppe mit gemischten Instances, indem Sie die Instance-Typen manuell auswählen

- Konfigurieren Sie eine Auto Scaling Scaling-Gruppe für die Verwendung von Instanzgewichten
- Verwenden Sie eine andere Startvorlage für einen Instance-Typ

Übersicht über die Einrichtung

Dieses Thema bietet einen Überblick und bewährte Verfahren für die Erstellung einer Gruppe gemischter Instanzen.

Inhalt

Übersicht

Es gibt zwei Möglichkeiten zum Erstellen einer Instances-Gruppe mit gemischten Instances:

- Attributbasierte Auswahl des Instance-Typs — Definieren Sie Ihre Rechenanforderungen, sodass Ihre Instance-Typen automatisch auf der Grundlage ihrer spezifischen Instance-Attribute ausgewählt werden.
- Manuelle Auswahl des Instance-Typs — Wählen Sie manuell die Instance-Typen aus, die zu Ihrem Workload passen.

Manual selection

In den folgenden Schritten wird beschrieben, wie Sie eine Instances-Gruppe erstellen, indem Sie die Instances-Gruppe manuell auswählen:

1. Wählen Sie eine Startvorlage, die die Parameter zum Starten einer EC2-Instance enthält. Parameter in Startvorlagen sind optional, aber Amazon EC2 Auto Scaling kann keine Instance starten, wenn die Amazon Machine Image-(AMI)-ID in der Startvorlage fehlt.
2. Wählen Sie die Option zum Überschreiben der Startvorlage.
3. Wählen Sie manuell die Instance-Typen aus, die zu Ihrem Workload passen.
4. Geben Sie die Prozentsätze der On-Demand-Instances und Spot Instances an, die gestartet werden sollen.
5. Die folgenden Zuweisungsstrategien bestimmen, wie die Amazon EC2 Auto Scaling-Gruppe Ihre gewünschte Kapazität für On-Demand- und Spot-Kapazität von den möglichen Instance-Typen erfüllt.
6. Wählen Sie die Availability Zones und VPC-Subnetze aus, in denen Sie Ihre Instances starten möchten.
7. Geben Sie die Anfangsgröße der Gruppe (die gewünschte Kapazität) sowie die Mindest- und Maximalgröße der Gruppe an.

Überschreibungen sind erforderlich, um den in der Startvorlage deklarierten Instance-Typ zu überschreiben und mehrere Instance-Typen zu verwenden, die in die eigene Ressourcendefinition der Auto-Scaling-Gruppe eingebettet sind. Weitere Informationen zu den verfügbaren Instance-Typen finden Sie unter Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch.

Sie können auch die folgenden optionalen Parameter für jeden Instance-Typ konfigurieren:

- `LaunchTemplateSpecification`— Sie können einem Instance-Typ nach Bedarf eine andere Startvorlage zuweisen. Diese Option ist zur Zeit in der Konsole nicht verfügbar. Weitere Informationen finden Sie unter Verwenden Sie eine andere Startvorlage für einen Instance-Typ.
- `WeightedCapacity`— Sie entscheiden, wie viel die Instance im Vergleich zu den übrigen Instances in Ihrer Gruppe auf die gewünschte Kapazität angerechnet wird. Wenn Sie einen `WeightedCapacity`-Wert für einen Instance-Typ angeben, müssen Sie einen `WeightedCapacity`-Wert für alle Instance-Typen angeben. Standardmäßig wird jede Instance als eine Instance auf Ihre gewünschte Kapazität angerechnet. Weitere Informationen finden Sie unter Konfigurieren Sie eine Auto Scaling Scaling-Gruppe für die Verwendung von Instanzgewichten.

Attribute-based selection

Damit Amazon EC2 Auto Scaling Ihre Instance-Typen automatisch auf der Grundlage ihrer spezifischen Instance-Attribute auswählen kann, erstellen Sie mithilfe der folgenden Schritte eine gemischte Instance-Gruppe, indem Sie Ihre Rechenanforderungen angeben:

1. Wählen Sie eine Startvorlage, die die Parameter zum Starten einer EC2-Instance enthält. Parameter in Startvorlagen sind optional, aber Amazon EC2 Auto Scaling kann keine Instance starten, wenn die Amazon Machine Image-(AMI)-ID in der Startvorlage fehlt.
2. Wählen Sie die Option zum Überschreiben der Startvorlage.
3. Geben Sie Instance-Attribute an, die Ihren Rechenanforderungen entsprechen, z. B. vCPUs und Speicheranforderungen.
4. Geben Sie die Prozentsätze der On-Demand-Instances und Spot Instances an, die gestartet werden sollen.
5. Die folgenden Zuweisungsstrategien bestimmen, wie die Amazon EC2 Auto Scaling-Gruppe Ihre gewünschte Kapazität für On-Demand- und Spot-Kapazität von den möglichen Instance-Typen erfüllt.
6. Wählen Sie die Availability Zones und VPC-Subnetze aus, in denen Sie Ihre Instances starten möchten.
7. Geben Sie die Anfangsgröße der Gruppe (die gewünschte Kapazität) sowie die Mindest- und Maximalgröße der Gruppe an.

Überschreibungen sind erforderlich, um den in der Startvorlage deklarierten Instance-Typ außer Kraft zu setzen und eine Reihe von Instance-Attributen zu verwenden, die Ihre Rechenanforderungen beschreiben. Informationen zu den unterstützten Attributen finden Sie InstanceRequirementsin der Amazon EC2 Auto Scaling API-Referenz. Alternativ können Sie eine Startvorlage verwenden, die bereits die Definition der Instance-Attribute enthält.

Sie können den `LaunchTemplateSpecification`-Parameter auch innerhalb der Overrides-Struktur konfigurieren, um einer Reihe von Instance-Anforderungen nach Bedarf eine andere Startvorlage zuzuweisen. Diese Option ist zur Zeit in der Konsole nicht verfügbar. Weitere Informationen finden Sie unter LaunchTemplateOverrides in der Amazon EC2 Auto Scaling API-Referenz.

Standardmäßig legen Sie die Anzahl der Instances als die gewünschte Kapazität Ihrer Auto Scaling-Gruppe fest.

Alternativ können Sie den Wert für die gewünschte Kapazität auf die Anzahl der vCPUs oder die Menge des Speichers setzen. Verwenden Sie dazu die `DesiredCapacityType`-Eigenschaft im `CreateAutoScalingGroup` API-Vorgang oder das Dropdown-Feld Gewünschter Kapazitätstyp im AWS Management Console. Dies ist eine nützliche Alternative zu Instance-Gewichten.

Flexibilität bezüglich der Instance-Größe

Um die Verfügbarkeit zu erhöhen, stellen Sie Ihre Anwendung für mehrere Instance-Typen bereit. Es hat sich bewährt, mehrere Instance-Typen zu verwenden, um die Kapazitätsanforderungen zu erfüllen. Dadurch kann Amazon EC2 Auto Scaling einen weiteren Instance-Typ starten, wenn in den ausgewählten Availability Zones nicht genügend Instance-Kapazität zur Verfügung steht.

Falls die Instance-Kapazität bei Spot Instances nicht ausreicht, versucht Amazon EC2 Auto Scaling immer wieder, Instances aus anderen Spot-Instance-Pools zu starten. (Die verwendeten Pools hängen von den von Ihnen ausgewählten Instance-Typen und der Zuweisungsstrategie ab.) Amazon EC2 Auto Scaling hilft Ihnen dabei, die Kosteneinsparungen von Spot Instances zu nutzen, indem Sie sie anstelle von On-Demand-Instances starten.

Wir empfehlen, für jeden Workload über mindestens 10 Instance-Typen hinweg flexibel zu sein. Beschränken Sie sich bei der Auswahl Ihrer Instance-Typen nicht auf die beliebtesten neuen Instance-Typen. Die Wahl von Instance-Typen der früheren Generation führt in der Regel zu weniger Spot-Unterbrechungen, da sie von On-Demand-Kunden weniger nachgefragt werden.

Flexibilität bezüglich der Availability Zone

Wir empfehlen dringend, dass Sie Ihre Auto Scaling-Gruppe auf mehrere Availability Zones verteilen. Mit mehreren Availability Zones können Sie Anwendungen entwerfen, die automatisch zwischen den Zonen umschalten, um die Ausfallsicherheit zu erhöhen.

Ein zusätzlicher Vorteil ist, dass Sie im Vergleich zu Gruppen in einer einzelnen Availability Zone auf einen größeren Amazon EC2-Kapazitätspool zugreifen können. Da die Kapazität für jeden Instance-Typ in jeder Availability Zone unabhängig schwankt, können Sie oft mehr Rechenkapazität mit Flexibilität sowohl für den Instance-Typ als auch für die Availability Zone erhalten.

Weitere Informationen zur Verwendung mehrerer Availability Zones finden Sie unter Beispiel: Aufteilen von Instances in mehrere Availability Zones.

Maximaler Spotpreis

Wenn Sie Ihre Auto Scaling Scaling-Gruppe mit dem AWS CLI oder einem SDK erstellen, können Sie den `SpotMaxPrice` Parameter angeben. Der `SpotMaxPrice`-Parameter bestimmt den Höchstpreis, den Sie für eine Spot-Instance-Stunde zu zahlen bereit sind.

Wenn Sie den `WeightedCapacity`-Parameter in Ihren Overrides (oder `"DesiredCapacityType": "vcpu"` oder `"DesiredCapacityType": "memory-mib"` auf Gruppenebene) angeben, stellt der Höchstpreis den maximalen Einzelpreis dar, nicht den Höchstpreis für eine ganze Instance.

Wir empfehlen ausdrücklich, keinen Höchstpreis anzugeben. Ihre Anwendung läuft möglicherweise nicht, wenn Sie keine Spot-Instances erhalten, z. B. wenn Ihr Höchstpreis zu niedrig ist. Wenn Sie keinen Höchstpreis angeben, entspricht der Standardhöchstpreis dem On-Demand-Preis. Sie zahlen nur den Spot-Preis für Spot-Instances, die Sie starten. Sie erhalten weiterhin die hohen Rabatte von Spot Instances. Diese Rabatte sind dank der stabilen Spot-Preise des Spot-Preismodells möglich. Weitere Informationen finden Sie unter Preise und Einsparungen im Amazon EC2 EC2-Benutzerhandbuch.

Proaktiver Kapazitätsausgleich

Wenn Ihr Anwendungsfall dies zulässt, empfehlen wir Capacity Rebalancing (Kapazitätsausgleich). Der Kapazitätsausgleich hilft Ihnen, die Verfügbarkeit von Workloads aufrechtzuerhalten, indem Sie Ihre Flotte proaktiv um eine neue Spot-Instance erweitern, bevor eine laufende Spot-Instance eine zweiminütige Spot-Instance-Unterbrechungsbenachrichtigung erhält.

Wenn der Kapazitätsausgleich aktiviert ist, versucht Amazon EC2 Auto Scaling proaktiv Spot-Instances zu ersetzen, für die eine Ausgleichsempfehlung vorliegt. Dies bietet Ihnen die Möglichkeit, Ihre Arbeitslast auf neue Spot-Instances zu verlagern, bei denen kein erhöhtes Risiko einer Unterbrechung besteht.

Weitere Informationen finden Sie unter Verwenden des Kapazitätsausgleichs, um Amazon-EC2-Spot-Unterbrechungen zu behandeln.

Skalierungsverhalten

Wenn Sie eine gemischte Instance-Gruppe erstellen, werden standardmäßig On-Demand-Instances verwendet. Um Spot-Instances verwenden zu können, müssen Sie den Prozentsatz der Gruppe ändern, die als On-Demand-Instances gestartet werden soll. Sie können eine beliebige Zahl zwischen 0 und 100 als On-Demand-Prozentsatz angeben.

Optional können Sie auch eine Basisanzahl von On-Demand-Instances festlegen, mit der begonnen werden soll. Wenn Sie dies tun, wartet Amazon EC2 Auto Scaling mit dem Start von Spot-Instances, bis die Basiskapazität der On-Demand-Instances erreicht ist, sobald die Gruppe aufskaliert. Für alles außerhalb der Basiskapazität werden die On-Demand-Prozentsätze verwendet, um zu bestimmen, wie viele On-Demand-Instances und Spot-Instances gestartet werden sollen.

Amazon EC2 Auto Scaling konvertiert den Prozentsatz in die entsprechende Anzahl von Instances. Wenn das Ergebnis eine Bruchzahl ergibt, wird zugunsten der On-Demand-Instances auf die nächste Ganzzahl aufgerundet.

Die folgende Tabelle veranschaulicht das Verhalten der Auto-Scaling-Gruppe, wenn sie sich vergrößert oder verkleinert.

Beispiel: Skalierungsverhalten

Kaufoptionen	Gruppengröße und Anzahl der laufenden Instances bei allen Kaufoptionen			
	10	20	30	40
Beispiel 1: Basis von 10, 50/50% On-Demand/ Spot				
On-Demand -Instances (Grundmenge)	10	10	10	10
On-Demand Instances	0	5	10	15
Spot-Instances	0	5	10	15
Beispiel 2: Basis von 0, 0/100% On-Demand/ Spot				

Kaufoptionen	Gruppengröße und Anzahl der laufenden Instances bei allen Kaufoptionen			
On-Demand -Instances (Grundmenge)	0	0	0	0
On-Demand Instances	0	0	0	0
Spot-Instances	10	20	30	40
Beispiel 3: Basis von 0, 60/40% On-Demand/ Spot				
On-Demand -Instances (Grundmenge)	0	0	0	0
On-Demand Instances	6	12	18	24
Spot-Instances	4	8	12	16
Beispiel 4: Basis von 0, 100/0% On-Demand/ Spot				
On-Demand -Instances (Grundmenge)	0	0	0	0
On-Demand Instances	10	20	30	40
Spot-Instances	0	0	0	0

Kaufoptionen	Gruppengröße und Anzahl der laufenden Instances bei allen Kaufoptionen			
Beispiel 5: Basis von 12, 0/100% On-Demand/ Spot				
On-Demand -Instances (Grundmenge)	10	12	12	12
On-Demand Instances	0	0	0	0
Spot-Instances	0	8	18	28

Wenn die Gruppengröße zunimmt, versucht Amazon EC2 Auto Scaling, Ihre Kapazität gleichmäßig über die angegebenen Availability Zones zu verteilen. Anschließend startet es Instance-Typen entsprechend der angegebenen Zuweisungsstrategie.

Wenn die Gruppengröße abnimmt, identifiziert Amazon EC2 Auto Scaling zunächst, welcher der beiden Typen (Spot oder On-Demand) beendet werden soll. Anschließend wird versucht, Instances auf ausgewogene Weise über Ihre angegebenen Availability Zones hinweg zu beenden. Außerdem wird die Beendigung von Instances auf eine Weise begünstigt, die Ihren Allokationsstrategien näher kommt. Weitere Informationen zu den Richtlinien zum Beenden finden Sie unter Kündigungsrichtlinien für Amazon EC2 Auto Scaling konfigurieren.

Regionale Verfügbarkeit von Instance-Typen

Die Verfügbarkeit von EC2-Instance-Typen hängt von Ihrem AWS-Region ab. So kann es beispielsweise sein, dass die neueste Generation von Instance-Typen in einer bestimmten Region noch nicht verfügbar ist. Aufgrund der regionalen Unterschiede bei der Instance-Verfügbarkeit können Probleme auftreten, sobald Sie programmatische Anfragen stellen, wenn mehrere Instance-Typen in Ihren Overrides in Ihrer Region nicht verfügbar sind. Die Verwendung mehrerer Instance-Typen, die in Ihrer Region nicht verfügbar sind, kann dazu führen, dass die Anfrage vollständig fehlschlägt. Um das Problem zu lösen, wiederholen Sie die Anfrage mit verschiedenen Instance-Typen und stellen Sie sicher, dass jeder Instance-Typ in der Region verfügbar ist. Um nach Instance-Typen zu suchen, die nach Standort angeboten werden, verwenden Sie den Befehl describe-instance-type-offerings

(Instance-Typ-Angebote beschreiben). Weitere Informationen finden Sie unter Suchen nach einem Amazon EC2 EC2-Instance-Typ im Amazon EC2 EC2-Benutzerhandbuch.

Zugehörige Ressourcen

Weitere bewährte Methoden für Spot-Instances finden Sie unter Bewährte Methoden für EC2 Spot im Amazon EC2 EC2-Benutzerhandbuch.

Einschränkungen

Nachdem Sie einer Auto Scaling Scaling-Gruppe mithilfe einer Richtlinie für gemischte Instanzen Overrides hinzugefügt haben, können Sie die Overrides mit dem `UpdateAutoScalingGroup` API-Aufruf aktualisieren, aber nicht löschen. Um die Überschreibungen vollständig zu entfernen, müssen Sie zunächst die Auto Scaling Scaling-Gruppe so ändern, dass sie eine Startvorlage oder eine Startkonfiguration anstelle einer Richtlinie für gemischte Instanzen verwendet. Anschließend können Sie erneut eine Richtlinie für gemischte Instanzen ohne Überschreibungen hinzufügen.

Zuweisungsstrategien

Wenn Sie mehrere Instance-Typen verwenden, verwalten Sie, wie Amazon EC2 Auto Scaling Ihre On-Demand- und Spot-Kapazität mithilfe der möglichen Instance-Typen erfüllt. Zu diesem Zweck spezifizieren Sie Zuweisungsstrategien und .

Informationen zu den bewährten Methoden für eine Gruppe mit gemischten Instanzen finden Sie unterÜbersicht über die Einrichtung.

Inhalt

Spot-Instances

Amazon EC2 Auto Scaling bietet die folgenden Allokationsstrategien, die für Spot-Instances verwendet werden können:

`price-capacity-optimized` (empfohlen)

Die preis- und kapazitätsoptimierte Zuweisungsstrategie betrachtet sowohl den Preis als auch die Kapazität, um die Spot-Instance-Pools auszuwählen, die am unwahrscheinlichsten unterbrochen werden und den niedrigstmöglichen Preis haben.

Wir empfehlen diese Strategie für den Einstieg. Weitere Informationen finden Sie im AWS Blog unter Einführung in die price-capacity-optimized Zuweisungsstrategie für EC2-Spot-Instances.

`capacity-optimized`

Amazon EC2 Auto Scaling fordert Ihre Spot Instance aus dem Pool mit optimaler Kapazität für die Anzahl der zu startenden Instances an.

Bei Spot-Instances ändert sich die Preisgestaltung im Laufe der Zeit basierend auf langfristigen Trends bei Angebot und Nachfrage langsam. Die Kapazität schwankt jedoch in Echtzeit. Bei Anwendung der Strategie `capacity-optimized` wird Spot-Instances automatisch zu den am besten verfügbaren Pools gestartet, indem Echtzeitdaten zur Kapazität analysiert werden und prognostiziert wird, welche Pools am besten verfügbar sind. Dies trägt dazu bei, mögliche Unterbrechungen für Workloads zu minimieren, bei denen Unterbrechungen ggf. aufgrund des Neustarts von Aufgaben sowie aufgrund von Checkpointing zu höheren Kosten führen. Um bestimmten Instance-Typen eine höhere Chance zu geben, zuerst zu starten, verwenden Sie `capacity-optimized-prioritized`.

`capacity-optimized-prioritized`

Sie legen die Reihenfolge der Instance-Typen für die Überschreibungen der Startvorlagen von der höchsten bis zur niedrigsten Priorität (vom ersten bis zum letzten in der Liste) fest. Amazon EC2 Auto Scaling erfüllt die Prioritäten des Instance-Typen auf Best Effort-Basis, optimiert jedoch zuerst die Kapazität. Dies ist eine gute Option für Workloads, bei denen die Möglichkeit von Unterbrechungen minimiert werden muss, aber auch die Präferenz für bestimmte Instance-Typen von Bedeutung ist. Wenn die On-Demand-Zuweisungsstrategie auf `prioritized` festgelegt ist, wird bei der Abdeckung von On-Demand-Kapazität die gleiche Priorität angewendet.

`lowest-price`

Amazon EC2 Auto Scaling fordert Ihre Spot Instances unter Verwendung der preisgünstigsten Pools innerhalb einer Availability Zone für die Anzahl N von Spot-Pools an, die Sie für die Einstellung Pools mit dem niedrigsten Preis angegeben haben. Wenn Sie also beispielsweise vier Instance-Typen und vier Availability Zones angeben, kann Ihre Auto-Scaling-Gruppe auf bis zu 16 Spot-Pools zugreifen. (Vier in jeder Availability Zone.) Wenn Sie für die Zuweisungsstrategie

zwei Spot-Pools (N=2) angeben, kann Ihre Auto-Scaling-Gruppe die beiden preisgünstigsten Pools pro Availability Zone nutzen, um Ihre Spot-Kapazität abzudecken.

Da bei dieser Strategie nur der Instance-Preis und nicht die Kapazitätsverfügbarkeit berücksichtigt wird, kann es zu hohen Unterbrechungsraten kommen.

Amazon EC2 Auto Scaling versucht, Spot Instances aus der Anzahl (N) der Pools zu ziehen, die Sie angeben. Wenn einem Pool die Spot-Kapazität ausgeht, bevor Ihre gewünschte Kapazität erreicht ist, wird Amazon EC2 Auto Scaling Ihre Anfrage weiterhin erfüllen, indem sie sie aus dem nächstgünstigsten Pool zieht. Um Ihre gewünschte Kapazität abzudecken, erhalten Sie möglicherweise Spot Instances aus mehr Pools als der von Ihnen angegebenen Anzahl (N). Wenn die meisten Pools keine Spot-Kapazität haben, erhalten Sie Ihre volle gewünschte Kapazität möglicherweise von weniger als der von Ihnen angegebenen Anzahl (N) von Pools.

> **Note**
>
> Wenn Sie eine Spot Instance mit aktiviertem AMD SEV-SNP starten, wird Ihnen eine zusätzliche stündliche Nutzungsgebühr in Höhe von 10 % des On-Demand-Stundensatzes des ausgewählten Instance-Typs berechnet. Wenn die Zuweisungsstrategie den Preis als Eingabe verwendet, berücksichtigt Amazon EC2 Auto Scaling diese zusätzliche Gebühr nicht; es wird nur der Spot-Preis verwendet.

On-Demand Instances

Amazon EC2 Auto Scaling bietet die folgenden Zuweisungsstrategien, die für On–Demand-Instances verwendet werden können:

`lowest-price`

Amazon EC2 Auto Scaling stellt automatisch den günstigsten Instance-Typ in jeder Availability Zone bereit, basierend auf dem aktuellen On-Demand-Preis.

Um Ihre gewünschte Kapazität zu erreichen, erhalten Sie in den einzelnen Availability Zones möglicherweise On-Demand-Instances von mehreren Instance-Typen. Dies hängt davon ab, wie viel Kapazität Sie anfordern.

`prioritized`

Bei der Abdeckung der On-Demand-Kapazität bestimmt Amazon EC2 Auto Scaling, welcher Instance-Typ zuerst verwendet wird (basierend auf der Reihenfolge der Instance-Typen in der Liste der Startvorlagen-Überschreibungen). Ein Beispiel: Angenommen, Sie geben drei Startvorlagen-Überschreibungen in der folgenden Reihenfolge an: `c5.large`, `c4.large` und `c3.large`. Beim Start Ihrer On-Demand-Instances verwendet die Auto-Scaling-Gruppe erst `c5.large`, dann `c4.large` und schließlich `c3.large`, um die On-Demand-Kapazität abzudecken.

Berücksichtigen Sie beim Verwalten der Prioritätsreihenfolge Ihrer On-Demand-Instances Folgendes:

- Sie können für die Nutzung im Voraus bezahlen, um erhebliche Rabatte für On-Demand-Instances zu erhalten, indem Sie entweder Savings Plans oder Reserved Instances verwenden. Weitere Informationen finden Sie auf der Preisseite für Amazon EC2.
- Bei Reserved Instances gilt die ermäßigte Rate der regulären On-Demand-Instance-Preise, wenn Amazon EC2 Auto Scaling passende Instance-Typen startet. Das bedeutet: Wenn Sie über ungenutzte Reserved Instances für `c4.large` verfügen, können Sie die Priorität der Instance-Typen so festlegen, dass dem Instance-Typ `c4.large` die höchste Priorität für Ihre Reserved Instances eingeräumt wird. Wenn eine `c4.large`-Instance gestartet wird, erhalten Sie die Preise für die Reserved Instance.
- Bei Savings Plans gelten bei Verwendung von Amazon EC2 Instance Savings Plans oder Compute Savings Plans die regulären On-Demand-Instance-Preise. Savings Plans bieten mehr Flexibilität bei der Priorisierung Ihrer Instance-Typen. Solange Sie Instance-Typen verwenden, die durch Ihren Savings Plan abgedeckt sind, können Sie sie in beliebiger Prioritätsreihenfolge festlegen. Sie können auch gelegentlich die gesamte Reihenfolge Ihrer Instance-Typen ändern und trotzdem weiterhin den Savings-Plan-Rabatt erhalten. Weitere Informationen zu Savings Plans finden Sie im Savings Plans User Guide.

Wie funktionieren die Allokationsstrategien mit Gewichten

Wenn Sie den `WeightedCapacity` Parameter in Ihren Überschreibungen (`"DesiredCapacityType": "vcpu"`oder `"DesiredCapacityType": "memory-mib"` auf Gruppenebene) angeben, funktionieren die Zuweisungsstrategien genauso wie bei anderen Auto Scaling Scaling-Gruppen.

Der einzige Unterschied besteht darin, dass, wenn Sie sich für die `price-capacity-optimized` Strategie `lowest-price` oder entscheiden, Ihre Instances aus den Instance-Pools mit dem niedrigsten Preis pro Einheit in jeder Availability Zone stammen. Weitere Informationen finden Sie unter Konfigurieren Sie eine Auto Scaling Scaling-Gruppe für die Verwendung von Instanzgewichten.

Stellen Sie sich zum Beispiel vor, Sie haben eine Auto-Scaling-Gruppe mit mehreren Instance-Typen, die unterschiedliche Mengen an vCPUs haben. Sie verwenden `lowest-price` für Ihre Spot- und On-Demand-Allokationsstrategien. Wenn Sie sich dafür entscheiden, Gewichtungen auf der Grundlage der vCPU-Anzahl jedes Instance-Typs zuzuweisen, startet Amazon EC2 Auto Scaling die Instance-Typen, die zum Zeitpunkt der Erfüllung den niedrigsten Preis für die von Ihnen zugewiesenen Gewichtungswerte haben (z. B. pro vCPU). Wenn es sich um eine Spot-Instance handelt, dann ist dies der niedrigste Spot-Preis pro vCPU. Wenn es sich um eine On-Demand-Instance handelt, dann ist dies der niedrigste On-Demand-Preis pro vCPU.

Erstellen einer gemischten Instances-Gruppe mit attributbasierter Auswahl des Instance-Typs

Anstatt Instance-Typen manuell für Ihre gemischte Instances-Gruppe auszuwählen, können Sie eine Reihe von Instance-Attributen angeben, die Ihre Rechenanforderungen beschreiben. Da Amazon EC2 Auto Scaling Instances startet, müssen alle Instance-Typen, die von der Auto-Scaling-Gruppe verwendet werden, Ihren gewünschten Instance-Attributen entsprechen. Dies ist bekannt als attributbasierte Instance-Typauswahl.

Dieser Ansatz ist ideal für Workloads und Frameworks, die bei der Wahl der Instance-Typen flexibel sind, wie z.B. Container, Big Data und CI/CD.

Im Folgenden finden Sie die Vorteile der attributbasierten Auswahl von Instance-Typen:

- Optimale Flexibilität für Spot-Instances — Amazon EC2 Auto Scaling kann aus einer Vielzahl von Instance-Typen für den Start von Spot-Instances wählen. Dies entspricht der bewährten Spot-Praxis, bei den Instance-Typen flexibel zu sein, wodurch der Amazon-EC2-Spot-Service eine bessere Chance hat, die von Ihnen benötigte Menge an Rechenkapazität zu finden und zuzuweisen.
- Einfache Verwendung der richtigen Instance-Typen — Bei so vielen verfügbaren Instance-Typen kann es zeitaufwändig sein, die richtigen Instance-Typen für Ihren Workload zu finden. Wenn Sie Instance-Attribute angeben, haben die Instance-Typen automatisch die erforderlichen Attribute für Ihre Workload.

- Automatische Verwendung neuer Instance-Typen — Ihre Auto Scaling Scaling-Gruppen können Instance-Typen der neueren Generation verwenden, sobald sie veröffentlicht werden. Instance-Typen der neueren Generation werden automatisch verwendet, wenn sie Ihren Anforderungen entsprechen und mit den Zuweisungsstrategien übereinstimmen, die Sie für Ihre Auto-Scaling-Gruppe gewählt haben.

Themen

Attributbasierte Auswahl von Instance-Typen

Bei der attributbasierten Auswahl von Instance-Typen geben Sie statt einer Liste bestimmter Instance-Typen eine Liste von Instance-Attributen an, die Ihre Instances benötigen, wie z. B.:

- vCPU-Anzahl — Die minimale und maximale Anzahl von vCPUs pro Instanz.
- Arbeitsspeicher — Das Minimum und das Maximum an Arbeitsspeicher GiBs pro Instanz.
- Lokaler Speicher — Ob EBS- oder Instance-Speicher-Volumes für den lokalen Speicher verwendet werden sollen.
- Spitzenleistung — Ob die T-Instance-Familie verwendet werden soll, einschließlich der Typen T4g, T3a, T3 und T2.

Es stehen viele Optionen zur Definition Ihrer Instance-Anforderungen zur Verfügung. Eine Beschreibung der einzelnen Optionen und der Standardwerte finden Sie InstanceRequirementsin der Amazon EC2 Auto Scaling API-Referenz.

Wenn Ihre Auto Scaling Scaling-Gruppe eine Instance starten muss, sucht sie nach Instance-Typen, die Ihren angegebenen Attributen entsprechen und in dieser Availability Zone verfügbar

sind. Die Zuweisungsstrategie bestimmt dann, welcher der passenden Instance-Typen gestartet werden soll. Standardmäßig ist für die attributbasierte Instance-Typauswahl eine Preisschutzfunktion aktiviert, um zu verhindern, dass Ihre Auto Scaling Scaling-Gruppe Instance-Typen startet, die Ihre Budgetschwellenwerte überschreiten.

Standardmäßig verwenden Sie die Anzahl der Instances als Maßeinheit, wenn Sie die gewünschte Kapazität Ihrer Auto Scaling Scaling-Gruppe festlegen, was bedeutet, dass jede Instanz als eine Einheit zählt.

Alternativ können Sie den Wert für die gewünschte Kapazität auf die Anzahl der vCPUs oder die Menge des Speichers setzen. Verwenden Sie dazu das Dropdown-Feld Gewünschter Kapazitätstyp in der AWS Management Console oder der `DesiredCapacityType` Eigenschaft in der `UpdateAutoScalingGroup` API-Operation `CreateAutoScalingGroup` oder. Amazon EC2 Auto Scaling startet dann die Anzahl der Instances, die erforderlich sind, um die gewünschte vCPU- oder Speicherkapazität zu erreichen. Wenn Sie beispielsweise vCPUs als gewünschten Kapazitätstyp verwenden und Instances mit jeweils 2 vCPUs verwenden, würde eine gewünschte Kapazität von 10 vCPUs 5 Instances starten. Dies ist eine nützliche Alternative zu Instance-Gewichten.

Preisschutz

Mit Preisschutz können Sie den Höchstpreis angeben, den Sie bereit sind, für EC2-Instances zu zahlen, die von Ihrer Auto Scaling Scaling-Gruppe gestartet wurden. Der Preisschutz ist eine Funktion, die verhindert, dass Ihre Auto Scaling Scaling-Gruppe Instance-Typen verwendet, die Sie für zu teuer halten würden, selbst wenn sie zufällig den von Ihnen angegebenen Attributen entsprechen.

Der Preisschutz ist standardmäßig aktiviert und hat separate Preisschwellen für On-Demand-Instances und Spot-Instances. Wenn Amazon EC2 Auto Scaling neue Instances starten muss, werden Instance-Typen, deren Preis über dem entsprechenden Schwellenwert liegt, nicht gestartet.

Themen

- Preisschutz auf Abruf
- Schutz vor Spot-Preisen
- Passen Sie den Preisschutz individuell an

Preisschutz auf Abruf

Für On-Demand-Instances definieren Sie den maximalen On-Demand-Preis, den Sie zu zahlen bereit sind, als Prozentsatz, der über dem angegebenen On-Demand-Preis liegt. Der identifizierte On-

Demand-Preis ist der Preis des günstigsten Instance-Typs C, M oder R der aktuellen Generation mit den von Ihnen angegebenen Attributen.

Wenn ein On-Demand-Preisschutzwert nicht explizit definiert ist, wird ein standardmäßiger maximaler On-Demand-Preis verwendet, der 20 Prozent über dem angegebenen On-Demand-Preis liegt.

Schutz vor Spot-Preisen

Standardmäßig wendet Amazon EC2 Auto Scaling automatisch den optimalen Spot-Instance-Preisschutz an, um konsistent aus einer Vielzahl von Instance-Typen auszuwählen. Sie können den Preisschutz auch manuell selbst festlegen. Wenn Sie dies jedoch Amazon EC2 Auto Scaling für Sie erledigen lassen, können Sie die Wahrscheinlichkeit erhöhen, dass Ihre Spot-Kapazität ausgeschöpft ist.

Sie können den Preisschutz mithilfe einer der folgenden Optionen manuell angeben. Wenn Sie den Preisschutz manuell festlegen, empfehlen wir, die erste Option zu verwenden.

- Ein Prozentsatz eines identifizierten On-Demand-Preises — Der identifizierte On-Demand-Preis ist der Preis des günstigsten Instance-Typs C, M oder R der aktuellen Generation mit Ihren angegebenen Attributen.
- Ein Prozentsatz höher als ein identifizierter Spot-Preis — Der identifizierte Spot-Preis ist der Preis des günstigsten Instance-Typs C, M oder R der aktuellen Generation mit den von Ihnen angegebenen Attributen. Wir empfehlen, diese Option nicht zu verwenden, da die Spot-Preise schwanken können und daher auch Ihr Preisschutzschwellenwert schwanken kann.

Passen Sie den Preisschutz individuell an

Sie können die Schwellenwerte für den Preisschutz in der Amazon EC2 Auto Scaling Scaling-Konsole oder mithilfe der SDKs AWS CLI oder anpassen.

- Verwenden Sie in der Konsole die Einstellungen On-Demand-Preisschutz und Spot-Preisschutz unter Zusätzliche Instance-Attribute.
- Verwenden Sie in der InstanceRequirementsStruktur die `OnDemandMaxPricePercentageOverLowestPrice` Eigenschaft, um den Schwellenwert für den Preisschutz bei On-Demand-Instances anzugeben. Um den Schwellenwert für den Preisschutz für Spot-Instances anzugeben, verwenden Sie entweder die `SpotMaxPricePercentageOverLowestPrice` Eigenschaft `MaxSpotPriceAsPercentageOfOptimalOnDemandPrice` oder.

Wenn Sie den gewünschten Kapazitätstyp (`DesiredCapacityType`) auf vCPUs oder Memory GiB festlegen, gilt der Preisschutz auf der Grundlage des Preises pro vCPU oder pro Speicher und nicht auf dem Preis pro Instance.

Sie können den Preisschutz auch deaktivieren. Wenn Sie angeben möchten, dass es keinen Schwellenwert für den Preisschutz gibt, geben Sie einen hohen Prozentwert an, z. 999999 B.

> **Note**
>
> Wenn keine Instance-Typen der aktuellen Generation C, M oder R Ihren angegebenen Attributen entsprechen, gilt der Preisschutz trotzdem. Wenn keine Übereinstimmung gefunden wird, stammt der identifizierte Preis von den günstigsten Instance-Typen der aktuellen Generation oder, falls dies nicht der Fall ist, von den günstigsten Instance-Typen der vorherigen Generation, die Ihren Attributen entsprechen.

Voraussetzungen

- Erstellen Sie eine Startvorlage. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.
- Stellen Sie sicher, dass die Startvorlage nicht bereits Spot-Instances anfordert.

Erstellen Sie eine gemischte Instanzgruppe mit attributbasierter Instanztypauswahl (Konsole)

Gehen Sie wie folgt vor, um eine gemischte Instances-Gruppe zu erstellen, indem Sie die attributbasierte Auswahl des Instance-Typs verwenden. Um die Schritte effizient ausführen zu können, wurden einige optionale Abschnitte übersprungen.

Für die meisten allgemeinen Arbeitslasten reicht es aus, die Anzahl der vCPUs und des Speichers anzugeben, die Sie benötigen. Für fortgeschrittene Anwendungsfälle können Sie Attribute wie Speichertyp, Netzwerkschnittstellen, CPU-Hersteller und Beschleunigertyp angeben.

Informationen zu den bewährten Methoden für eine Gruppe mit gemischten Instanzen finden Sie unter. Übersicht über die Einrichtung

So erstellen Sie eine gemischte Instances-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.

2. Wählen Sie auf der Navigationsleiste oben auf dem Bildschirm dieselbe AWS-Region , die Sie bei der Erstellung der Startvorlage angegeben haben.
3. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
4. Geben Sie auf der Seite Startvorlage oder -konfiguration auswählen für Auto-Scaling-Gruppenname einen Namen für Ihre Auto-Scaling-Gruppe ein.
5. Gehen Sie folgendermaßen vor, um Ihre Startungsvorlage auszuwählen:

   a. Wählen Sie für Launch template (Startvorlage) eine vorhandene Startvorlage aus.

   b. Wählen Sie unter Launch template version (Version der Startvorlage) aus, ob die Auto-Scaling-Gruppe beim horizontalen Skalieren nach oben die standardmäßige, die neueste oder eine bestimmte Version der Startvorlage verwenden soll.

   c. Stellen Sie sicher, dass Ihre Startvorlage alle Optionen unterstützt, die Sie verwenden möchten, und wählen Sie dann Next (Weiter) aus.
6. Wählen Sie auf der Seite Instance-Startoptionen auswählen die folgenden Einstellungen aus.

   a. Wählen Sie für Instance-Typanforderungen die Option Startvorlage überschreiben.

   > **Note**
   >
   > Bei Auswahl einer Startvorlage, die bereits eine Reihe von Instance-Attributen wie etwa vCPUs und Arbeitsspeicher enthält, werden die Instance-Attribute angezeigt. Diese Attribute werden zu den Eigenschaften der Auto-Scaling-Gruppe hinzugefügt, wo Sie sie jederzeit über die Amazon EC2 Auto Scaling-Konsole aktualisieren können.

   b. Beginnen Sie unter Instance-Attribute angeben mit der Eingabe Ihrer vCPU- und Speicheranforderungen.

      - Geben Sie für vCPUs die gewünschte minimale und maximale Anzahl der vCPUs ein. Um kein Limit anzugeben, wählen Sie Kein Minimum, Kein Maximum oder beides.
      - Geben Sie für Arbeitsspeicher (GiB) den gewünschten Mindest- und Höchstwert ein. Um kein Limit anzugeben, wählen Sie Kein Minimum, Kein Maximum oder beide Optionen aus.

   c. (Optional) Für Zusätzliche Instance-Attribute können Sie optional ein oder mehrere Attribute angeben, um Ihre Computinganforderungen genauer auszudrücken. Jedes zusätzliche Attribut fügt Ihrer Anfrage weitere Einschränkungen hinzu.

d. Erweitern Sie Vorschau der passenden Instance-Typen, um die Instance-Typen mit Ihren angegebenen Attributen anzuzeigen.

e. Geben Sie unter Optionen für den Kauf von Instances unter Instances Distribution die Prozentsätze der Gruppe an, die als On-Demand-Instances bzw. Spot Instances gestartet werden sollen. Wenn Ihre Anwendung zustandslos und fehlertolerant ist und damit umgehen kann, dass eine Instance unterbrochen wird, können Sie einen höheren Prozentsatz an Spot-Instances angeben.

f. (Optional) Wenn Sie sich für einen Prozentsatz an Spot Instances entschieden haben, können Sie On-Demand-Basiskapazität einbeziehen auswählen und dann die Mindestmenge der Anfangskapazität der Auto-Scaling-Gruppe angeben, die von On-Demand-Instances erfüllt werden muss. Für alles, was über die Basiskapazität hinausgeht, werden die Einstellungen für die Instance-Verteilung verwendet, um zu bestimmen, wie viele On-Demand-Instances und Spot-Instances gestartet werden sollen.

g. Unter Zuteilungsstrategien wird für die On-Demand-Zuteilungsstrategie automatisch der niedrigste Preis ausgewählt und kann nicht geändert werden.

h. Wählen Sie für die Spot-Zuweisungsstrategie eine Zuweisungsstrategie. Price capacity optimized (Preiskapazität optimiert) ist standardmäßig ausgewählt. Lowest price (Niedrigster Preis) ist standardmäßig ausgeblendet und wird nur angezeigt, wenn Sie Show all strategies (Alle Strategien anzeigen) auswählen. Wenn Sie Niedrigster Preis ausgewählt haben, geben Sie zur übergreifenden Verteilung für Pools mit dem niedrigsten Preis die Anzahl der Pools mit dem niedrigsten Preis an.

i. Für Kapazitätsausgleich wählen Sie aus, ob Sie den Kapazitätsausgleich aktivieren oder deaktivieren möchten. Verwenden Sie Capacity Rebalancing, um automatisch zu reagieren, wenn Ihre Spot Instances aufgrund einer Spot-Unterbrechung bald beendet werden. Weitere Informationen finden Sie unter Verwenden des Kapazitätsausgleichs, um Amazon-EC2-Spot-Unterbrechungen zu behandeln.

j. Wählen Sie unter Netzwerk für VPC eine VPC. Die Auto-Scaling-Gruppe muss in derselben VPC erstellt werden wie die Sicherheitsgruppe, die Sie in Ihrer Startvorlage angegeben haben.

k. Wählen Sie für Availability Zones and subnets (Subnetz) eines der öffentlichen Subnetze in der festgelegten VPC aus. Verwenden Sie Subnetze in mehreren Availability Zones, um eine hohe Verfügbarkeit zu erzielen. Weitere Informationen finden Sie unter Überlegungen bei der Auswahl von VPC-Subnetzen.

l. Wählen Sie Weiter, Weiter aus.

7. Gehen Sie für den Schritt Gruppengröße und Skalierungsrichtlinien konfigurieren wie folgt vor:

    a. Um Ihre gewünschte Kapazität in anderen Einheiten als Instances zu messen, wählen Sie die entsprechende Option für Gruppengröße, Gewünschter Kapazitätstyp aus. Einheiten, vCPUs und Arbeitsspeicher GiB werden unterstützt. Standardmäßig gibt Amazon EC2 Auto Scaling Einheiten an, was sich in einer Anzahl von Instances niederschlägt.

    b. Für Gewünschte Kapazität, die Anfangsgröße Ihrer Auto-Scaling-Gruppe.

    c. Wenn im Abschnitt Skalierung unter Skalierungslimits Ihr neuer Wert für die gewünschte Kapazität größer als die gewünschte Mindestkapazität und die gewünschte Höchstkapazität ist, wird die gewünschte Höchstkapazität automatisch auf den neuen Wert für die gewünschte Kapazität erhöht. Sie können die Limits bei Bedarf ändern. Weitere Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.

8. Wählen Sie Skip to review (Mit Prüfen fortfahren) aus.
9. Wählen Sie auf der Seite Review (Prüfen) Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Erstellen Sie eine gemischte Instanzgruppe mit einer attributbasierten Instanztypauswahl ()AWS CLI

Erstellen Sie wie folgt eine gemischte Instances-Gruppe über die Befehlszeile:

Verwenden Sie einen der folgenden Befehle:

- create-auto-scaling-group (AWS CLI)
- Neu-AS-Gruppe AutoScaling ()AWS Tools for Windows PowerShell

Beispielkonfiguration

Um eine Auto-Scaling-Gruppe mit attributbasierter Auswahl des Instance-Typs über die AWS CLI zu erstellen, verwenden Sie den folgenden Befehl create-auto-scaling-group.

Die folgenden Instance-Attribute werden angegeben:

- `VCpuCount` – Die Instance-Typen müssen mindestens vier vCPUs und maximal acht vCPUs haben.
- `MemoryMiB` – Die Instance-Typen müssen über mindestens 16.384 MiB Arbeitsspeicher verfügen.
- `CpuManufacturers` – Die Instance-Typen müssen eine von Intel hergestellte CPU haben.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Im Folgenden sehen Sie ein Beispiel für eine `config.json`-Datei.

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredCapacityType": "units",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Default"
            },
            "Overrides": [{
                "InstanceRequirements": {
                    "VCpuCount": {"Min": 4, "Max": 8},
                    "MemoryMiB": {"Min": 16384},
                    "CpuManufacturers": ["intel"]
                }
            }]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 50,
            "SpotAllocationStrategy": "price-capacity-optimized"
        }
    },
    "MinSize": 0,
    "MaxSize": 100,
    "DesiredCapacity": 4,
    "DesiredCapacityType": "units",
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

Um den Wert für die gewünschte Kapazität als Anzahl der vCPUs oder die Größe des Arbeitsspeichers festzulegen, geben Sie `"DesiredCapacityType": "vcpu"` oder `"DesiredCapacityType": "memory-mib"` in der Datei an. Der Standardtyp für die gewünschte Kapazität ist `units`, wodurch der Wert für die gewünschte Kapazität als Anzahl der Instanzen festgelegt wird.

YAML

Alternativ können Sie den Befehl create-auto-scaling-group verwenden, um die Auto-Scaling-Gruppe zu erstellen. Dadurch wird auf eine YAML-Datei als einziger Parameter für Ihre Auto-Scaling-Gruppe verwiesen.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

Im Folgenden sehen Sie ein Beispiel für eine `config.yaml`-Datei.

```
---
AutoScalingGroupName: my-asg
DesiredCapacityType: units
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceRequirements:
        VCpuCount:
          Min: 2
          Max: 4
        MemoryMiB:
          Min: 2048
        CpuManufacturers:
        - intel
  InstancesDistribution:
    OnDemandPercentageAboveBaseCapacity: 50
    SpotAllocationStrategy: price-capacity-optimized
MinSize: 0
MaxSize: 100
DesiredCapacity: 4
DesiredCapacityType: units
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Um den Wert für die gewünschte Kapazität als Anzahl der vCPUs oder die Größe des Arbeitsspeichers festzulegen, geben Sie `DesiredCapacityType: vcpu` oder `DesiredCapacityType: memory-mib` in der Datei an. Der Standardtyp für die gewünschte Kapazität ist `units`, wodurch der Wert für die gewünschte Kapazität als Anzahl der Instanzen festgelegt wird.

Eine Vorschau Ihrer Instance-Typen anzeigen

Sie können die Instance-Typen, die Ihren Rechenanforderungen entsprechen, in der Vorschau anzeigen, ohne sie zu starten, und Ihre Anforderungen bei Bedarf anpassen. Wenn Sie Ihre Auto-Scaling-Gruppe in der Amazon EC2 Auto Scaling-Konsole erstellen, erscheint eine Vorschau der Instance-Typen im Abschnitt Preview matching instance types (Vorschau passender Instance-Typen) auf der Seite Choose instance launch options (Startoptionen für Instances auswählen).

Alternativ können Sie eine Vorschau der Instance-Typen anzeigen, indem Sie einen Amazon EC2 GetInstanceTypesFromInstanceRequirementsEC2-API-Aufruf mit dem AWS CLI oder einem SDK durchführen. Übergeben Sie die `InstanceRequirements`-Parameter in der Anfrage genau in dem Format, das Sie zum Erstellen oder Aktualisieren einer Auto-Scaling-Gruppe verwenden würden. Weitere Informationen finden Sie unter Preview-Instance-Typen mit bestimmten Attributen im Amazon EC2 EC2-Benutzerhandbuch.

Zugehörige Ressourcen

Weitere Informationen zur attributbasierten Instanztypauswahl finden Sie unter Attributbasierte Instanztypauswahl für EC2 Auto Scaling und EC2 Fleet im Blog. AWS

Sie können eine attributbasierte Auswahl des Instance-Typs erklären, wenn Sie eine Auto-Scaling-Gruppe mit AWS CloudFormation erstellen. Weitere Informationen finden Sie unter Auto Scaling-Vorlagenbeispiele im Abschnitt des AWS CloudFormation -Benutzerhandbuchs.

Erstellen Sie eine Gruppe mit gemischten Instances, indem Sie die Instance-Typen manuell auswählen

In diesem Thema erfahren Sie, wie Sie mehrere Instance-Typen in einer einzelnen Auto-Scaling-Gruppe starten, indem Sie die Instance-Typen manuell auswählen.

Wenn Sie Instance-Attribute lieber als Kriterien für die Auswahl von Instance-Typen verwenden möchten, finden Sie weitere Informationen unter Erstellen einer gemischten Instances-Gruppe mit attributbasierter Auswahl des Instance-Typs.

Inhalt

Voraussetzungen

- Erstellen Sie eine Startvorlage. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.
- Stellen Sie sicher, dass die Startvorlage nicht bereits Spot-Instances anfordert.

Eine gemischte Instances-Gruppe (Konsole) erstellen

Gehen Sie wie folgt vor, um eine Instances-Gruppe zu erstellen, indem Sie manuell auswählen, welche Instance-Typen Ihre Gruppe starten kann. Um die Schritte effizient ausführen zu können, wurden einige optionale Abschnitte übersprungen.

Informationen zu den bewährten Methoden für eine Gruppe mit gemischten Instanzen finden Sie unterÜbersicht über die Einrichtung.

So erstellen Sie eine gemischte Instances-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie auf der Navigationsleiste oben auf dem Bildschirm dieselbe AWS-Region , die Sie bei der Erstellung der Startvorlage angegeben haben.
3. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
4. Geben Sie auf der Seite Startvorlage oder -konfiguration auswählen für Auto-Scaling-Gruppenname einen Namen für Ihre Auto-Scaling-Gruppe ein.
5. Gehen Sie folgendermaßen vor, um Ihre Startungsvorlage auszuwählen:

   a. Wählen Sie für Launch template (Startvorlage) eine vorhandene Startvorlage aus.

   b. Wählen Sie unter Launch template version (Version der Startvorlage) aus, ob die Auto-Scaling-Gruppe beim horizontalen Skalieren nach oben die standardmäßige, die neueste oder eine bestimmte Version der Startvorlage verwenden soll.

   c. Stellen Sie sicher, dass Ihre Startvorlage alle Optionen unterstützt, die Sie verwenden möchten, und wählen Sie dann Next (Weiter) aus.
6. Wählen Sie auf der Seite Instance-Startoptionen auswählen die folgenden Einstellungen aus.

   a. Wählen Sie für Instance type requirements (Anforderungen an Instance-Typen) Override launch template (Startvorlage überschreiben), Manually add instance types (Startvorlage überschreiben) aus.

b. Wählen Sie Ihre Instance-Typen aus. Sie können unsere Empfehlungen als Ausgangspunkt verwenden. Die Option Family and generation flexible (Familie und Generation flexibel) ist standardmäßig ausgewählt.

   - Um die Reihenfolge der Instance-Typen zu ändern, verwenden Sie die Pfeile. Wenn Sie eine Zuweisungsstrategie auswählen, die Priorisierung unterstützt, legt die Reihenfolge der Instance-Typen deren Startpriorität fest.
   - Um einen Instance-Typ zu entfernen, wählen Sie X aus.
   - (Optional) Für die Felder in der Spalte Gewichtung können Sie jedem Instance-Typ eine relative Gewichtung zuweisen. Geben Sie dazu die Anzahl der Einheiten ein, die eine Instance dieses Typs zur gewünschten Kapazität der Gruppe beiträgt. Dies kann nützlich sein, wenn sich bei den Instance-Typen die vCPU, der Arbeitsspeicher, der Speicherplatz oder die Netzwerkbandbreitenfunktionen unterscheiden. Weitere Informationen finden Sie unter Konfigurieren Sie eine Auto Scaling Scaling-Gruppe für die Verwendung von Instanzgewichten.

     Wenn Sie sich für Empfehlungen vom Typ Größe flexibel entschieden haben, haben alle Instance-Typen, die Teil dieses Abschnitts sind, automatisch einen Gewichtungswert. Wenn Sie keine Gewichtungen angeben möchten, leeren Sie die Felder in der Spalte Weight (Gewichtung) für alle Instance-Typen.

c. Geben Sie unter Optionen für den Kauf von Instances unter Instances Distribution die Prozentsätze der Gruppe an, die als On-Demand-Instances bzw. Spot-Instances gestartet werden sollen. Wenn Ihre Anwendung zustandslos und fehlertolerant ist und damit umgehen kann, dass eine Instance unterbrochen wird, können Sie einen höheren Prozentsatz an Spot-Instances angeben.

d. (Optional) Wenn Sie sich für einen Prozentsatz an Spot Instances entschieden haben, können Sie On-Demand-Basiskapazität einbeziehen auswählen und dann die Mindestmenge der Anfangskapazität der Auto-Scaling-Gruppe angeben, die von On-Demand-Instances erfüllt werden muss. Für alles, was über die Basiskapazität hinausgeht, werden die Einstellungen für die Instance-Verteilung verwendet, um zu bestimmen, wie viele On-Demand-Instances und Spot-Instances gestartet werden sollen.

e. Wählen Sie unter Zuteilungsstrategien für On-Demand-Zuteilungsstrategie eine Zuteilungsstrategie. Wenn Sie Ihre Instance-Typen manuell auswählen, ist Prioritized (Priorisiert) standardmäßig ausgewählt.

f. Wählen Sie für die Spot-Zuweisungsstrategie eine Zuweisungsstrategie. Price capacity optimized (Preiskapazität optimiert) ist standardmäßig ausgewählt. Lowest price (Niedrigster

Preis) ist standardmäßig ausgeblendet und wird nur angezeigt, wenn Sie Show all strategies (Alle Strategien anzeigen) auswählen.

- Wenn Sie Niedrigster Preis ausgewählt haben, geben Sie zur übergreifenden Verteilung für Pools mit dem niedrigsten Preis die Anzahl der Pools mit dem niedrigsten Preis an.
- Wenn Sie Kapazitätsoptimiert ausgewählt haben, können Sie optional das Feld Instance-Typen priorisieren aktivieren, um Amazon EC2 Auto Scaling auf der Grundlage der Reihenfolge, in der Ihre Instance-Typen aufgeführt sind, auswählen zu lassen, welcher Instance-Typ zuerst gestartet wird.

g. Für Kapazitätsausgleich wählen Sie aus, ob Sie den Kapazitätsausgleich aktivieren oder deaktivieren möchten. Verwenden Sie Capacity Rebalancing, um automatisch zu reagieren, wenn Ihre Spot Instances aufgrund einer Spot-Unterbrechung bald beendet werden. Weitere Informationen finden Sie unter Verwenden des Kapazitätsausgleichs, um Amazon-EC2-Spot-Unterbrechungen zu behandeln.

h. Wählen Sie unter Netzwerk für VPC eine VPC. Die Auto-Scaling-Gruppe muss in derselben VPC erstellt werden wie die Sicherheitsgruppe, die Sie in Ihrer Startvorlage angegeben haben.

i. Wählen Sie für Availability Zones and subnets (Subnetz) eines der öffentlichen Subnetze in der festgelegten VPC aus. Verwenden Sie Subnetze in mehreren Availability Zones, um eine hohe Verfügbarkeit zu erzielen. Weitere Informationen finden Sie unter Überlegungen bei der Auswahl von VPC-Subnetzen.

j. Wählen Sie Weiter, Weiter aus.

7. Gehen Sie für den Schritt Gruppengröße und Skalierungsrichtlinien konfigurieren wie folgt vor:

a. Geben Sie unter Gruppengröße für Gewünschte Kapazität die anfängliche Anzahl von Instances ein, die gestartet werden sollen.

Standardmäßig wird die gewünschte Kapazität als Anzahl von Instances ausgedrückt. Wenn Sie Ihren Instance-Typen Gewichtungen zugewiesen haben, müssen diese Werte in die Maßeinheit umgerechnet werden, die Sie für die Zuweisung der Gewichtungen verwendet haben (beispielsweise die Anzahl von vCPUs).

b. Wenn im Abschnitt Skalierung unter Skalierungslimits Ihr neuer Wert für die gewünschte Kapazität größer als die gewünschte Mindestkapazität und die gewünschte Höchstkapazität ist, wird die gewünschte Höchstkapazität automatisch auf den neuen Wert für die gewünschte Kapazität erhöht. Sie können die Limits bei Bedarf ändern. Weitere

Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.

8. Wählen Sie Skip to review (Mit Prüfen fortfahren) aus.
9. Wählen Sie auf der Seite Review (Prüfen) Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Eine gemischte Instances-Gruppe (AWS CLI) erstellen

Erstellen Sie wie folgt eine gemischte Instances-Gruppe über die Befehlszeile:

Verwenden Sie einen der folgenden Befehle:

- create-auto-scaling-group (AWS CLI)
- AutoScalingNeu-AS-Gruppe ()AWS Tools for Windows PowerShell

Beispielkonfigurationen

Die folgenden Beispielkonfigurationen zeigen, wie gemischte Instance-Gruppen mit verschiedenen Spot-Zuweisungsstrategien erstellt werden können.

> **Note**
>
> Diese Beispiele zeigen, wie Sie eine Konfigurationsdatei verwenden, die in JSON oder YAML formatiert ist. Wenn Sie AWS CLI Version 1 verwenden, müssen Sie eine Konfigurationsdatei im JSON-Format angeben. Wenn Sie AWS CLI Version 2 verwenden, können Sie eine Konfigurationsdatei angeben, die entweder in YAML oder JSON formatiert ist.

Beispiele

- Beispiel 1: Starten von Spot-Instances mit der capacity-optimized- Zuweisungsstrategie
- Beispiel 2: Starten von Spot-Instances mit der capacity-optimized-prioritized- Zuweisungsstrategie
- Beispiel 3: Starten von Spot-Instances mit der über zwei Pools diversifizierten lowest-price-Zuweisungsstrategie
- Beispiel 4: Starten von Spot-Instances mit der price-capacity-optimized- Zuweisungsstrategie

Beispiel 1: Starten von Spot-Instances mit der **`capacity-optimized`**- Zuweisungsstrategie

Der folgende create-auto-scaling-group-Befehl erstellt eine Auto-Scaling-Gruppe, die Folgendes angibt:

- Der Prozentsatz der Gruppe, die als On-Demand-Instances gestartet werden soll (`0`) und eine Basisanzahl von On-Demand-Instances, mit denen begonnen werden soll (`1`).
- Die in der Prioritätsreihenfolge zu startenden Instance-Typen (`c5.large`, `c5a.large`, `m5.large`, `m5a.large`, `c4.large`, `m4.large`, `c3.large`, `m3.large`).
- Die Subnetze, in denen die Instances gestartet werden sollen (`subnet-5ea0c127`, `subnet-6194ea3b`, `subnet-c934b782`). Diese entsprechen jeweils einer anderen Availability Zone.
- Beschreibt eine Startvorlage (`my-launch-template`) und die Version der Startvorlage (`$Default`).

Wenn Amazon EC2 Auto Scaling versucht, Ihre On-Demand-Kapazität zu erfüllen, wird zuerst der `c5.large`-Instance-Typ gestartet. Die Spot-Instances stammen aus dem optimalen Spot-Pool in jeder Availability Zone basierend auf der Spot-Instance-Kapazität.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Die Datei `config.json` enthält den folgenden Inhalt.

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Default"
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
```

```
                {
                    "InstanceType": "m5.large"
                },
                {
                    "InstanceType": "m5a.large"
                },
                {
                    "InstanceType": "c4.large"
                },
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandBaseCapacity": 1,
            "OnDemandPercentageAboveBaseCapacity": 0,
            "SpotAllocationStrategy": "capacity-optimized"
        }
    },
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 3,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

Alternativ können Sie den Befehl create-auto-scaling-group verwenden, um die Auto-Scaling-Gruppe zu erstellen. Dadurch wird auf eine YAML-Datei als einziger Parameter für Ihre Auto-Scaling-Gruppe verwiesen.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

Die Datei `config.yaml` enthält den folgenden Inhalt.

```
---
```

```
AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
    - InstanceType: c3.large
    - InstanceType: m3.large
  InstancesDistribution:
    OnDemandBaseCapacity: 1
    OnDemandPercentageAboveBaseCapacity: 0
    SpotAllocationStrategy: capacity-optimized
MinSize: 1
MaxSize: 5
DesiredCapacity: 3
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Beispiel 2: Starten von Spot-Instances mit der **`capacity-optimized-prioritized`**-Zuweisungsstrategie

Der folgende create-auto-scaling-group-Befehl erstellt eine Auto-Scaling-Gruppe, die Folgendes angibt:

- Der Prozentsatz der Gruppe, die als On-Demand-Instances gestartet werden soll (`0`) und eine Basisanzahl von On-Demand-Instances, mit denen begonnen werden soll (1).
- Die in der Prioritätsreihenfolge zu startenden Instance-Typen (`c5.large`, `c5a.large`, `m5.large`, `m5a.large`, `c4.large`, `m4.large`, `c3.large`, `m3.large`).
- Die Subnetze, in denen die Instances gestartet werden sollen (`subnet-5ea0c127`, `subnet-6194ea3b`, `subnet-c934b782`). Diese entsprechen jeweils einer anderen Availability Zone.
- Beschreibt eine Startvorlage (`my-launch-template`) und die Version der Startvorlage (`$Latest`).

Wenn Amazon EC2 Auto Scaling versucht, Ihre On-Demand-Kapazität zu erfüllen, wird zuerst der `c5.large`-Instance-Typ gestartet. Wenn Amazon EC2 Auto Scaling versucht, Ihre Spot-Kapazität zu erfüllen, erfüllt es die Prioritäten des Instance-Typen auf Best-Effort-Basis. An erster Stelle steht jedoch immer die Kapazitätsoptimierung.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Die Datei `config.json` enthält den folgenden Inhalt.

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
                {
                    "InstanceType": "m5.large"
                },
                {
                    "InstanceType": "m5a.large"
                },
                {
                    "InstanceType": "c4.large"
                },
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
```

```
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandBaseCapacity": 1,
            "OnDemandPercentageAboveBaseCapacity": 0,
            "SpotAllocationStrategy": "capacity-optimized-prioritized"
        }
    },
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 3,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

Alternativ können Sie den Befehl create-auto-scaling-group verwenden, um die Auto-Scaling-Gruppe zu erstellen. Dadurch wird auf eine YAML-Datei als einziger Parameter für Ihre Auto-Scaling-Gruppe verwiesen.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

Die Datei `config.yaml` enthält den folgenden Inhalt.

```
---
AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
    - InstanceType: c3.large
    - InstanceType: m3.large
  InstancesDistribution:
    OnDemandBaseCapacity: 1
```

```
    OnDemandPercentageAboveBaseCapacity: 0
    SpotAllocationStrategy: capacity-optimized-prioritized
MinSize: 1
MaxSize: 5
DesiredCapacity: 3
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Beispiel 3: Starten von Spot-Instances mit der über zwei Pools diversifizierten **`lowest-price`**-Zuweisungsstrategie

Der folgende create-auto-scaling-group-Befehl erstellt eine Auto-Scaling-Gruppe, die Folgendes angibt:

- Der Prozentsatz der Gruppe, die als On-Demand-Instances gestartet werden soll (`50`). (Gibt keine Basisanzahl von On-Demand-Instances an, mit der gestartet werden soll.)
- Die in der Prioritätsreihenfolge zu startenden Instance-Typen (`c5.large`, `c5a.large`, `m5.large`, `m5a.large`, `c4.large`, `m4.large`, `c3.large`, `m3.large`).
- Die Subnetze, in denen die Instances gestartet werden sollen (`subnet-5ea0c127`, `subnet-6194ea3b`, `subnet-c934b782`). Diese entsprechen jeweils einer anderen Availability Zone.
- Beschreibt eine Startvorlage (`my-launch-template`) und die Version der Startvorlage (`$Latest`).

Wenn Amazon EC2 Auto Scaling versucht, Ihre On-Demand-Kapazität zu erfüllen, wird zuerst der `c5.large`-Instance-Typ gestartet. Für Ihre Spot-Kapazität versucht Amazon EC2 Auto Scaling, die Spot Instances gleichmäßig über die beiden kostengünstigsten Pools in jeder Availability Zone zu starten.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Die Datei `config.json` enthält den folgenden Inhalt.

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
```

```
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
                {
                    "InstanceType": "m5.large"
                },
                {
                    "InstanceType": "m5a.large"
                },
                {
                    "InstanceType": "c4.large"
                },
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 50,
            "SpotAllocationStrategy": "lowest-price",
            "SpotInstancePools": 2
        }
    },
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 3,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

Alternativ können Sie den Befehl create-auto-scaling-group verwenden, um die Auto-Scaling-Gruppe zu erstellen. Dadurch wird auf eine YAML-Datei als einziger Parameter für Ihre Auto-Scaling-Gruppe verwiesen.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

Die Datei `config.yaml` enthält den folgenden Inhalt.

```
---
AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
    - InstanceType: c3.large
    - InstanceType: m3.large
  InstancesDistribution:
    OnDemandPercentageAboveBaseCapacity: 50
    SpotAllocationStrategy: lowest-price
    SpotInstancePools: 2
MinSize: 1
MaxSize: 5
DesiredCapacity: 3
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Beispiel 4: Starten von Spot-Instances mit der **`price-capacity-optimized`**- Zuweisungsstrategie

Der folgende create-auto-scaling-group-Befehl erstellt eine Auto-Scaling-Gruppe, die Folgendes angibt:

- Der Prozentsatz der Gruppe, die als On-Demand-Instances gestartet werden soll (`30`). (Gibt keine Basisanzahl von On-Demand-Instances an, mit der gestartet werden soll.)

- Die in der Prioritätsreihenfolge zu startenden Instance-Typen (`c5.large`, `c5a.large`, `m5.large`, `m5a.large`, `c4.large`, `m4.large`, `c3.large`, `m3.large`).
- Die Subnetze, in denen die Instances gestartet werden sollen (`subnet-5ea0c127`, `subnet-6194ea3b`, `subnet-c934b782`). Diese entsprechen jeweils einer anderen Availability Zone.
- Beschreibt eine Startvorlage (`my-launch-template`) und die Version der Startvorlage (`$Latest`).

Wenn Amazon EC2 Auto Scaling versucht, Ihre On-Demand-Kapazität zu erfüllen, wird zuerst der `c5.large`-Instance-Typ gestartet. Für Ihre Spot-Kapazität versucht Amazon EC2 Auto Scaling, die Spot Instances gleichmäßig über Spot-Instance-Pools mit dem günstigsten Preis sowie mit optimaler Kapazität für die Anzahl der zu startenden Instances zu starten.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Die Datei `config.json` enthält den folgenden Inhalt.

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
                {
                    "InstanceType": "m5.large"
                },
                {
                    "InstanceType": "m5a.large"
                },
```

```
                {
                    "InstanceType": "c4.large"
                },
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 30,
            "SpotAllocationStrategy": "price-capacity-optimized"
        }
    },
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 3,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

Alternativ können Sie den Befehl create-auto-scaling-group verwenden, um die Auto-Scaling-Gruppe zu erstellen. Dadurch wird auf eine YAML-Datei als einziger Parameter für Ihre Auto-Scaling-Gruppe verwiesen.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

Die Datei `config.yaml` enthält den folgenden Inhalt.

```
---
AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
```

```
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
    - InstanceType: c3.large
    - InstanceType: m3.large
  InstancesDistribution:
    OnDemandPercentageAboveBaseCapacity: 30
    SpotAllocationStrategy: price-capacity-optimized
MinSize: 1
MaxSize: 5
DesiredCapacity: 3
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Konfigurieren Sie eine Auto Scaling Scaling-Gruppe für die Verwendung von Instanzgewichten

Wenn Sie mehrere Instance-Typen verwenden, können Sie angeben, wie viele Einheiten jedem Instance-Typ zugeordnet werden sollen, und dann die Kapazität Ihrer Gruppe mit derselben Maßeinheit angeben. Diese Option zur Kapazitätsspezifikation wird als Gewichte bezeichnet.

Nehmen wir einmal an, Sie führen eine rechenintensive Anwendung aus, die mit mindestens 8 vCPUs und 15 GiB RAM am besten funktioniert. Wenn Sie `c5.2xlarge` als Basiseinheit verwenden, würde jeder der folgenden EC2-Instance-Typen Ihre Anwendungsanforderungen erfüllen.

Beispiel für Instance-Typen

Instance-Typ	vCPU	Arbeitsspeicher (GiB)
`c5.2xlarge`	8	16
`c5.4xlarge`	16	32
`c5.12xlarge`	48	96
`c5.18xlarge`	72	144
`c5.24xlarge`	96	192

Standardmäßig haben alle Instance-Typen unabhängig von ihrer Größe das gleiche Gewicht. Mit anderen Worten: Unabhängig davon, ob Amazon EC2 Auto Scaling einen großen oder kleinen Instance-Typ startet, zählt jede Instance gleich viel für die gewünschte Kapazität der Auto-Scaling-Gruppe.

Bei Gewichtungen weisen Sie jedoch einen Zahlenwert zu, der angibt, wie viele Einheiten jedem Instance-Typ zugeordnet werden sollen. Wenn die Instances beispielsweise unterschiedliche Größen aufweisen, kann eine `c5.2xlarge`-Instance eine Gewichtung von „2“ haben, und eine (doppelt so große) `c5.4xlarge` könnte eine Gewichtung von „4“ haben usw. Wenn Amazon EC2 Auto Scaling die Gruppe skaliert, geben die Gewichtungen die Anzahl der Einheiten an, die jede Instance auf die gewünschte Kapazität angerechnet wird.

Die Gewichtungen ändern nicht, welche Instance-Typen Amazon EC2 Auto Scaling startet. Stattdessen tun dies die Zuweisungsstrategien. Weitere Informationen finden Sie unter Zuweisungsstrategien.

> **Important**
>
> Um eine Auto-Scaling-Gruppe so zu konfigurieren, dass sie die gewünschte Kapazität mithilfe der Anzahl der vCPUs oder der Menge des Speichers jedes Instance-Typs erfüllt, empfehlen wir die attributbasierte Auswahl des Instance-Typs. Durch die Einstellung des `DesiredCapacityType` Parameters wird automatisch die Anzahl der Einheiten angegeben, die jedem Instance-Typ zugeordnet werden sollen, basierend auf dem Wert, den Sie für diesen Parameter festlegen. Weitere Informationen finden Sie unter Erstellen einer gemischten Instances-Gruppe mit attributbasierter Auswahl des Instance-Typs.

Inhalt

Überlegungen

In diesem Abschnitt werden die wichtigsten Überlegungen zur effektiven Implementierung von Gewichtungen erörtert.

- Wählen Sie einige Instance-Typen aus, die den Leistungsanforderungen Ihrer Anwendung entsprechen. Entscheiden Sie anhand ihrer Fähigkeiten, welches Gewicht jeder Instance-Typ auf die gewünschte Kapazität Ihrer Auto Scaling Scaling-Gruppe angerechnet werden soll. Diese Gewichte gelten für aktuelle und future Fälle.
- Vermeiden Sie große Gewichtungsunterschiede. Geben Sie beispielsweise nicht die Gewichtung 1 für einen Instance-Typ an, wenn der nächstgrößere Instance-Typ eine Gewichtung von 200 hat. Der Unterschied zwischen der kleinsten und der größten Gewichtung sollte auch nicht extrem sein. Extreme Gewichtsunterschiede können sich negativ auf die Optimierung von Kosten und Leistung auswirken.
- Geben Sie die gewünschte Kapazität der Gruppe in Einheiten und nicht in Instanzen an. Wenn Sie beispielsweise vCPU-basierte Gewichtungen verwenden, legen Sie die gewünschte Anzahl von Kernen sowie die Mindest- und Höchstzahl fest.
- Legen Sie die Gewichtungen und die gewünschte Kapazität so fest, dass die gewünschte Kapazität mindestens zwei- bis dreimal größer ist als Ihr größtes Gewicht.

Beachten Sie bei der Aktualisierung vorhandener Gruppen Folgendes:

- Wenn Sie einer vorhandenen Gruppe Gewichtungen hinzufügen, schließen Sie Gewichtungen für alle derzeit verwendeten Instance-Typen mit ein.
- Wenn Sie Gewichtungen hinzufügen oder ändern, startet oder beendet Amazon EC2 Auto Scaling Instances, um die gewünschte Kapazität auf der Grundlage der neuen Gewichtungswerte zu erreichen.
- Wenn Sie einen Instance-Typ entfernen, behalten laufende Instances dieses Typs ihre letzte Gewichtung, auch wenn sie nicht mehr definiert sind.

Verhalten beim Gewichten von Instanzen

Wenn Sie Instance-Gewichtungen verwenden, verhält sich Amazon EC2 Auto Scaling folgendermaßen:

- Die aktuelle Kapazität wird entweder bei der gewünschten Kapazität oder darüber liegen. Die aktuelle Kapazität kann die gewünschte Kapazität überschreiten, wenn Instances gestartet werden, die die verbleibenden gewünschten Kapazitätseinheiten überschreiten. Angenommen, Sie geben die zwei Instance-Typen `c5.2xlarge` und `c5.12xlarge` an und weisen für `c5.2xlarge` eine Instance-Gewichtung von „2“ und für `c5.12xlarge` eine von „12“ zu. Wenn 5 Einheiten übrig

sind, um die gewünschte Kapazität zu erfüllen, und Amazon EC2 Auto Scaling eine `c5.12xlarge` bereitstellt, wird die gewünschte Kapazität um sieben Einheiten überschritten.

- Beim Starten von Instances priorisiert Amazon EC2 Auto Scaling die Verteilung der Kapazität auf die Availability Zones und die Einhaltung der Zuweisungsstrategien gegenüber der Überschreitung der gewünschten Kapazität.
- Amazon EC2 Auto Scaling kann die maximale Kapazitätsgrenze überschreiten, um das Gleichgewicht zwischen den Availability Zones aufrechtzuerhalten. Dabei werden Ihre bevorzugten Zuweisungsstrategien verwendet. Das von Amazon EC2 Auto Scaling erzwungene feste Limit ist Ihre gewünschte Kapazität zuzüglich Ihres größten Gewichts.

Konfigurieren einer Auto-Scaling-Gruppe zur Verwendung von Gewichtungen

Sie können eine Auto-Scaling-Gruppe für die Verwendung von Gewichtungen konfigurieren, wie in den folgenden AWS CLI Beispielen gezeigt. Weitere Informationen zur Verwendung der Konsole finden Sie unter Erstellen Sie eine Gruppe mit gemischten Instances, indem Sie die Instance-Typen manuell auswählen.

So konfigurieren Sie eine Auto-Scaling-Gruppe zur Verwendung von Gewichtungen (AWS CLI)

Verwenden Sie den create-auto-scaling-groups-Befehl. Der folgende Befehl erstellt zum Beispiel eine neue Auto-Scaling-Gruppe und weist Gewichtungen zu, indem er Folgendes angibt:

- Der Prozentsatz der Gruppe, die als On-Demand-Instances gestartet werden soll (`0`)
- Die Zuordnungsstrategie für Spot-Instances in jeder Availability Zone (`capacity-optimized`)
- Die in der Prioritätsreihenfolge zu startenden Instance-Typen (`m4.16xlarge`, `m5.24xlarge`)
- Die Instance-Gewichtungen, die dem relativen Größenunterschied (vCPUs) zwischen Instance-Typen (16, 24) entsprechen
- Die Subnetze, in denen die Instances gestartet werden sollen (`subnet-5ea0c127`, `subnet-6194ea3b`, `subnet-c934b782`), die jeweils einer anderen Availability Zone entsprechen
- Beschreibt eine Startvorlage (`my-launch-template`) und die Version der Startvorlage (`$Latest`).

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Die Datei `config.json` enthält den folgenden Inhalt.

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "m4.16xlarge",
                    "WeightedCapacity": "16"
                },
                {
                    "InstanceType": "m5.24xlarge",
                    "WeightedCapacity": "24"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 0,
            "SpotAllocationStrategy": "capacity-optimized"
        }
    },
    "MinSize": 160,
    "MaxSize": 720,
    "DesiredCapacity": 480,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
    "Tags": []
}
```

So konfigurieren Sie eine vorhandene Auto-Scaling-Gruppe für die Verwendung von Gewichtungen (AWS CLI)

Verwenden Sie den Befehl update-auto-scaling-group. Der folgende Befehl weist beispielsweise den Instance-Typen in einer bestehenden Auto-Scaling-Gruppe Gewichtungen zu, indem er Folgendes angibt:

- Die in der Prioritätsreihenfolge zu startenden Instance-Typen (`c5.18xlarge`, `c5.24xlarge`, `c5.2xlarge`, `c5.4xlarge`)
- Die Instance-Gewichtungen, die dem relativen Größenunterschied (vCPUs) zwischen Instance-Typen (18, 24, 2, 4) entsprechen

- Die neue, erhöhte gewünschte Kapazität, die größer als das größte Gewicht ist

```
aws autoscaling update-auto-scaling-group --cli-input-json file://~/config.json
```

Die Datei `config.json` enthält den folgenden Inhalt.

```
{
    "AutoScalingGroupName": "my-existing-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "Overrides": [
                {
                    "InstanceType": "c5.18xlarge",
                    "WeightedCapacity": "18"
                },
                {
                    "InstanceType": "c5.24xlarge",
                    "WeightedCapacity": "24"
                },
                {
                    "InstanceType": "c5.2xlarge",
                    "WeightedCapacity": "2"
                },
                {
                    "InstanceType": "c5.4xlarge",
                    "WeightedCapacity": "4"
                }
            ]
        }
    },
    "MinSize": 0,
    "MaxSize": 100,
    "DesiredCapacity": 100
}
```

So überprüfen Sie die Gewichtungen mithilfe der Befehlszeile

Verwenden Sie einen der folgenden Befehle:

- describe-auto-scaling-groups (AWS CLI)
- Get-AS-Gruppe AutoScaling ()AWS Tools for Windows PowerShell

Beispiel: Spot-Preis pro Einheitsstunde

Die folgende Tabelle vergleicht den stündlichen Preis für Spot-Instances in verschiedenen Availability Zones in USA Ost (Nord-Virginia) mit dem Preis für On-Demand-Instances in derselben Region. Bei den angezeigten Preisen handelt es sich um Beispielpreise und nicht um aktuelle Preise. Dies sind Ihre Kosten pro Instance-Stunde.

Beispiel: Spot-Preise pro Instance-Stunde

Instance-Typ	us-ost-1a	us-ost-1b	us-ost-1c	On-Demand-Preise
`c5.2xlarge`	0,180 US-Dollar	0,191 US-Dollar	0,170 US-Dollar	0,34 US-Dollar
`c5.4xlarge`	0,341 US-Dollar	0,361 US-Dollar	0,318 US-Dollar	0,68 US-Dollar
`c5.12xlarge`	0,779 US-Dollar	0,777 US-Dollar	0,777 US-Dollar	2,04 US-Dollar
`c5.18xlarge`	1,207 US-Dollar	1,475 US-Dollar	1,357 US-Dollar	3,06 US-Dollar
`c5.24xlarge`	1,555 US-Dollar	1,555 US-Dollar	1,555 US-Dollar	4,08 US-Dollar

Mit der Instance-Gewichtung können Sie Ihre Kosten auf Grundlage Ihrer Verwendung pro Einheitsstunde bewerten. Der Preis pro Einheitsstunde lässt sich bestimmen, indem der Preis für einen Instance-Typ durch die Anzahl an Einheiten geteilt wird, den er darstellt. Bei On-Demand-Instances entspricht der Preis pro Einheitsstunde bei der Bereitstellung eines Instance-Typs dem Preis der Bereitstellung desselben Instance-Typs einer anderen Größe. Im Gegensatz dazu variiert der Spot-Preis pro Einheitsstunde nach Spot-Pool.

Das folgende Beispiel zeigt, wie die Berechnung des Spot-Preises pro Stunde mit Instance-Gewichtungen funktioniert. Angenommen, Sie möchten Spot-Instances nur in `us-east-1a` starten. Der Preis pro Stunde wird in der folgenden Tabelle erfasst.

Beispiel: Spot-Preis pro Einheitsstunde

Instance-Typ	us-ost-1a	Instance-Gewichtung	Preis pro Einheitsstunde
`c5.2xlarge`	0,180 US-Dollar	2	0,090 US-Dollar
`c5.4xlarge`	0,341 US-Dollar	4	0,085 US-Dollar
`c5.12xlarge`	0,779 US-Dollar	12	0,065 US-Dollar
`c5.18xlarge`	1,207 US-Dollar	18	0,067 US-Dollar
`c5.24xlarge`	1,555 US-Dollar	24	0,065 US-Dollar

Verwenden Sie eine andere Startvorlage für einen Instance-Typ

Sie können nicht nur mehrere Instance-Typen verwenden, sondern auch mehrere Startvorlagen.

Nehmen wir an, Sie konfigurieren eine Auto-Scaling-Gruppe für rechenintensive Anwendungen und möchten eine Mischung aus C5-, C5a- und C6g-Instance-Typen einbeziehen. C6g-Instances verfügen jedoch über einen AWS Graviton-Prozessor, der auf der 64-Bit-ARM-Architektur basiert, während die C5- und C5a-Instances auf 64-Bit-Intel x86-Prozessoren ausgeführt werden. Die AMIs für C5- und C5a-Instances funktionieren beide auf diesen Instances, aber nicht auf C6g-Instances. Verwenden Sie eine andere Startvorlage für C6g-Instances, um dieses Problem zu lösen. Sie können immer noch dieselbe Startvorlage für C5- und C5a-Instances verwenden.

Dieser Abschnitt enthält Verfahren zur Verwendung von, um Aufgaben im AWS CLI Zusammenhang mit der Verwendung mehrerer Startvorlagen auszuführen. Derzeit ist diese Funktion nur verfügbar, wenn Sie die AWS CLI oder ein SDK verwenden, und ist nicht von der Konsole aus verfügbar.

Inhalt

Konfigurieren einer Auto-Scaling-Gruppe zum Verwenden mehrerer Startvorlagen

Sie können eine Auto-Scaling-Gruppe so konfigurieren, dass sie mehrere Startvorlagen verwendet, wie in den folgenden Beispielen gezeigt.

So konfigurieren Sie eine neue Auto-Scaling-Gruppe für die Verwendung mehrerer Startvorlagen (AWS CLI)

Verwenden Sie den create-auto-scaling-groups-Befehl. Mit dem folgenden Befehl wird zum Beispiel eine neue Auto-Scaling-Gruppe erstellt. Es gibt die Instance-Typen `c5.large`, `c5a.large` und `c6g.large` an und definiert eine neue Startvorlage für den Instance-Typ `c6g.large`, um sicherzustellen, dass ein geeignetes AMI zum Starten von Arm-Instances verwendet wird. Amazon EC2 Auto Scaling verwendet die Reihenfolge der Instance-Typen, um festzulegen, welcher Instance-Typ beim Erfüllen der On-Demand-Kapazität zuerst verwendet werden soll.

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Die Datei `config.json` enthält den folgenden Inhalt.

```
{
  "AutoScalingGroupName":"my-asg",
  "MixedInstancesPolicy":{
    "LaunchTemplate":{
      "LaunchTemplateSpecification":{
        "LaunchTemplateName":"my-launch-template-for-x86",
        "Version":"$Latest"
      },
      "Overrides":[
        {
          "InstanceType":"c6g.large",
          "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template-for-arm",
            "Version": "$Latest"
          }
        },
        {
          "InstanceType":"c5.large"
        },
        {
          "InstanceType":"c5a.large"
        }
      ]
    },
    "InstancesDistribution":{
      "OnDemandBaseCapacity": 1,
      "OnDemandPercentageAboveBaseCapacity": 50,
      "SpotAllocationStrategy": "capacity-optimized"
```

```
    }
  },
  "MinSize":1,
  "MaxSize":5,
  "DesiredCapacity":3,
  "VPCZoneIdentifier":"subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
  "Tags":[ ]
}
```

So konfigurieren Sie eine bestehende Auto-Scaling-Gruppe für die Verwendung mehrerer Startvorlagen (AWS CLI)

Verwenden Sie den Befehl update-auto-scaling-group. Der folgende Befehl weist beispielsweise die Startvorlage namens *my-launch-template-for-arm* dem *c6g.large*-Instance-Typ für die Auto-Scaling-Gruppe namens *my-asg* zu.

```
aws autoscaling update-auto-scaling-group --cli-input-json file://~/config.json
```

Die Datei `config.json` enthält den folgenden Inhalt.

```
{
  "AutoScalingGroupName":"my-asg",
  "MixedInstancesPolicy":{
    "LaunchTemplate":{
      "Overrides":[
        {
          "InstanceType":"c6g.large",
          "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template-for-arm",
            "Version": "$Latest"
          }
        },
        {
          "InstanceType":"c5.large"
        },
        {
          "InstanceType":"c5a.large"
        }
      ]
    }
  }
}
```

So überprüfen Sie die Startvorlagen für eine Auto-Scaling-Gruppe

Verwenden Sie einen der folgenden Befehle:

- describe-auto-scaling-groups (AWS CLI)
- AutoScalingGet-AS-Gruppe ()AWS Tools for Windows PowerShell

Zugehörige Ressourcen

Ein Beispiel für die Angabe mehrerer Startvorlagen mithilfe der attributbasierten Instanztypauswahl finden Sie in einer AWS CloudFormation Vorlage auf re:POST.AWS

Erstellen Sie Auto-Scaling-Gruppen mit Startkonfigurationen

> **Important**
>
> Sie können `CreateLaunchConfiguration` nicht mit neuen Typen von Amazon-EC2-Instances aufrufen, die nach dem 31. Dezember 2022 veröffentlicht wurden. Darüber hinaus besteht für alle neuen Konten, die nach dem 1. Juni 2023 erstellt werden, nicht die Möglichkeit, neue Startkonfigurationen über die Konsole zu erstellen. In future werden neue Konten nicht mehr in der Lage sein, neue Startkonfigurationen mithilfe der Konsole, API, CLI und zu erstellen CloudFormation. Migrieren Sie zu Startvorlagen, um sicherzustellen, dass Sie weder jetzt noch in future neue Startkonfigurationen erstellen müssen. Informationen zum Migrieren Ihrer Auto-Scaling-Gruppen zu Startvorlagen finden Sie unter Migrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten.

Wenn Sie eine Startkonfiguration oder eine EC2-Instance erstellt haben, können Sie eine Auto-Scaling-Gruppe erstellen, die eine Startkonfiguration als Konfigurationsvorlage für ihre EC2-Instances verwendet. Die Startkonfiguration gibt Informationen wie die AMI-ID, den Instance-Typ, das Schlüsselpaar, Sicherheitsgruppen und die Blockgerät-Zuweisung für Ihre Instances an. Weitere Informationen zum Erstellen von Startkonfigurationen finden Sie unter Erstellen einer Startkonfiguration.

Sie müssen über IAM-Berechtigungen verfügen, um eine Auto-Scaling-Gruppe zu erstellen. Sie müssen auch über ausreichende Berechtigungen verfügen, um die serviceverknüpfte Rolle zu erstellen, die Amazon EC2 Auto Scaling verwendet, um Aktionen in Ihrem Namen durchzuführen, falls sie noch nicht existiert. Beispiele für IAM-Richtlinien, die ein Administrator als Referenz für

die Erteilung von Berechtigungen verwenden kann, finden Sie unter Beispiele für identitätsbasierte Richtlinien.

Inhalt

Eine Auto-Scaling-Gruppe mithilfe einer Startkonfiguration erstellen

> **Important**
>
> Wir stellen Informationen zu Startkonfigurationen für Kunden bereit, die noch nicht von Startkonfigurationen zu Startvorlagen migriert sind. Informationen zum Migrieren Ihrer Auto-Scaling-Gruppen zu Startvorlagen finden Sie unter Migrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten.

Wenn Sie eine Auto-Scaling-Gruppe erstellen, müssen Sie die notwendigen Informationen zur Konfiguration der Amazon EC2-Instances, die Availability Zones und VPC-Subnetze für die Instances, die gewünschte Kapazität sowie die minimalen und maximalen Kapazitätsgrenzen angeben.

Das folgende Verfahren veranschaulicht das Erstellen einer Auto-Scaling-Gruppe mithilfe einer Startkonfiguration. Sie können eine Startkonfiguration nicht ändern, nachdem sie erstellt wurde. Sie können jedoch die Startkonfiguration für eine Auto-Scaling-Gruppe ersetzen. Weitere Informationen finden Sie unter Ändern der Startkonfiguration für eine Auto-Scaling-Gruppe.

Voraussetzungen

- Sie müssen eine Startkonfiguration erstellt haben. Weitere Informationen finden Sie unter Erstellen einer Startkonfiguration.

Erstellen einer Auto-Scaling-Gruppe mithilfe einer Startkonfiguration (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.

2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm dieselbe aus, AWS-Region die Sie bei der Erstellung der Startkonfiguration verwendet haben.
3. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
4. Geben Sie auf der Seite Startvorlage oder -konfiguration auswählen für Auto-Scaling-Gruppenname einen Namen für Ihre Auto-Scaling-Gruppe ein.
5. Gehen Sie folgendermaßen vor, um eine Startkonfiguration auszuwählen:

   a. Wählen Sie unter Launch Template (Startvorlage) die Option Switch to launch configuration (Wechsel zur Startkonfiguration) aus.

   b. Wählen Sie unter Launch configuration (Startkonfiguration) eine vorhandene Startkonfiguration aus.

   c. Stellen Sie sicher, dass Ihre Startkonfiguration alle Optionen unterstützt, die Sie verwenden möchten, und wählen Sie dann Next (Weiter) aus.
6. Wählen Sie auf der Seite Instance-Startoptionen konfigurieren unter Netzwerk für VPC eine VPC aus. Die Auto-Scaling-Gruppe muss in derselben VPC erstellt werden wie die Sicherheitsgruppe, die Sie in Ihrer Startkonfiguration angegeben haben.
7. Für Availability Zones und Subnets (Subnetze) wählen Sie ein oder mehrere Subnetze in der angegebenen VPC aus. Verwenden Sie Subnetze in mehreren Availability Zones, um eine hohe Verfügbarkeit zu erzielen. Weitere Informationen finden Sie unter Überlegungen bei der Auswahl von VPC-Subnetzen.
8. Wählen Sie Weiter.

   Oder akzeptieren Sie die weiteren Standardwerte, und klicken Sie dann auf Skip to review (Mit Prüfen fortfahren).
9. (Optional) Konfigurieren Sie auf der Seite Konfigurieren von erweiterten Optionen die folgenden Optionen und wählen Sie Weiter:

   a. Wählen Sie unter Zusätzliche Einstellungen, Überwachung, aus, ob die Erfassung von CloudWatch Gruppenmetriken aktiviert werden soll. Diese Metriken liefern Messwerte, die Indikatoren für ein potenzielles Problem sein können, wie z.B. die Anzahl der abgebrochenen Instances oder die Anzahl der ausstehenden Instances. Weitere Informationen finden Sie unter Überwachen Sie CloudWatch Metriken für Ihre Auto Scaling Scaling-Gruppen und -Instances.

   b. Wählen Sie unter Standardinstanzaufwärmen aktivieren diese Option und wählen Sie die Aufwärmzeit für Ihre Anwendung aus. Wenn Sie eine Auto Scaling-Gruppe mit einer

Skalierungsrichtlinie erstellen, verbessert die Standard-Instance-Aufwärmfunktion die CloudWatch Amazon-Metriken, die für die dynamische Skalierung verwendet werden. Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

10. Konfigurieren Sie auf der Seite Configure group size and scaling policies (Gruppengröße und Skalierungsrichtlinien konfigurieren) die folgenden Optionen, und wählen Sie dann Next (Weiter):

    a. Geben Sie unter Gruppengröße für Gewünschte Kapazität die anfängliche Anzahl von Instances ein, die gestartet werden sollen.

    b. Wenn im Abschnitt Skalierung unter Skalierungslimits Ihr neuer Wert für die gewünschte Kapazität größer als die gewünschte Mindestkapazität und die gewünschte Höchstkapazität ist, wird die gewünschte Höchstkapazität automatisch auf den neuen Wert für die gewünschte Kapazität erhöht. Sie können die Limits bei Bedarf ändern. Weitere Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.

    c. Wählen Sie für Automatische Skalierung aus, ob Sie eine Skalierungsrichtlinie für die Zielverfolgung erstellen möchten. Sie können diese Richtlinie auch erstellen, nachdem Sie Ihre Auto-Scaling-Gruppe erstellt haben.

       Wenn Sie sich für die Skalierungsrichtlinie für die Zielverfolgung entscheiden, befolgen Sie die Anweisungen unter Erstellen einer Zielverfolgungs-Skalierungsrichtlinie, um die Richtlinie zu erstellen.

    d. Wählen Sie unter Instance-Wartungsrichtlinie aus, ob Sie eine Instance-Wartungsrichtlinie erstellen möchten. Sie können diese Richtlinie auch erstellen, nachdem Sie Ihre Auto-Scaling-Gruppe erstellt haben. Befolgen Sie zum Erstellen der Richtlinie die Anweisungen unter Festlegen einer Instance-Wartungsrichtlinie.

    e. Wählen Sie unter Instance scale-in protection (Instance-Skalierungsschutz), ob der Instance-Skalierungsschutz aktiviert werden soll. Weitere Informationen finden Sie unter Instance-Abskalierungsschutz verwenden.

11. (Optional) Um Benachrichtigungen zu erhalten, konfigurieren Sie für Add notification (Benachrichtigungen hinzufügen) die Benachrichtigung und wählen Sie anschließend Next (Weiter) aus. Weitere Informationen finden Sie unter Amazon SNS-Benachrichtigungsoptionen für Amazon EC2 Auto Scaling.

12. (Optional) Um Tags hinzuzufügen, wählen Sie Add tag (Tag hinzufügen) aus, geben Sie für jedes Tag einen Tag-Schlüssel und einen Wert an und wählen Sie anschließend Next (Weiter) aus. Weitere Informationen finden Sie unter Tagging von Auto-Scaling-Gruppen und Instances.

13. Wählen Sie auf der Seite Review (Prüfen) Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

Erstellen Sie wie folgt eine Auto-Scaling-Gruppe über die Befehlszeile:

Verwenden Sie einen der folgenden Befehle:

- create-auto-scaling-group (AWS CLI)
- AutoScalingGroupNeu-AS ()AWS Tools for Windows PowerShell

Eine Auto-Scaling-Gruppe unter Verwendung von Parametern einer bestehenden Instance erstellen

> **Important**
>
> Wir stellen Informationen zu Startkonfigurationen für Kunden bereit, die noch nicht von Startkonfigurationen zu Startvorlagen migriert sind. Informationen zum Migrieren Ihrer Auto-Scaling-Gruppen zu Startvorlagen finden Sie unter Migrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten.

Wenn Sie zum ersten Mal eine Auto-Scaling-Gruppe erstellen, empfehlen wir Ihnen, die Konsole zu verwenden, um eine Startvorlage aus einer bestehenden EC2-Instance zu erstellen. Verwenden Sie dann die Startvorlage, um eine neue Auto-Scaling-Gruppe zu erstellen. Informationen zu diesen Verfahren finden Sie unter Erstellen einer Auto-Scaling-Gruppe mithilfe des Amazon EC2-Startassistenten.

Das folgende Verfahren zeigt, wie Sie eine Auto-Scaling-Gruppe erstellen, indem Sie eine vorhandene Instance angeben, die als Basis zum Starten anderer Instances verwendet werden soll. Für die Erstellung einer EC2-Instance sind mehrere Parameter erforderlich, z. B. die Amazon Machine Image (AMI) ID, der Instanztyp, das Schlüsselpaar und die Sicherheitsgruppe. Alle diese Informationen werden auch von Amazon EC2 Auto Scaling benutzt, um Instances in Ihrem Namen zu starten, wenn eine Skalierung erforderlich ist. Diese Informationen werden entweder in einer Startvorlage oder in einer Startkonfiguration gespeichert.

Wenn Sie eine vorhandene Instance verwenden, erstellt Amazon EC2 Auto Scaling eine Auto-Scaling-Gruppe, die Instances auf der Grundlage einer gleichzeitig erstellten Startkonfiguration

startet. Die neue Startkonfiguration hat denselben Namen wie die Auto-Scaling-Gruppe und enthält bestimmte Konfigurationsdetails der identifizierten Instance.

Die folgenden Konfigurationsdetails werden von der identifizierten Instance in die Startkonfiguration kopiert:

- AMI-ID
- Instance-Typ
- Schlüsselpaar
- Sicherheitsgruppen
- Typ der IP-Adresse (öffentlich oder privat)
- IAM-Instance-Profil, falls zutreffend
- Überwachung (richtig oder falsch)
- EBS optimiert (richtig oder falsch)
- Tenancy-Einstellung beim Start in einer VPC (geteilt oder dediziert)
- Kernel-ID und RAM-Datenträger-ID, falls zutreffend
- Benutzerdaten, falls angegeben
- Spotpreis (maximal)

Das VPC-Subnetz und die Availability Zone werden von der identifizierten Instance in die eigene Ressourcendefinition der Auto-Scaling-Gruppe kopiert.

Wenn sich die identifizierte Instance in einer Platzierungsgruppe befindet, startet die neue Auto-Scaling-Gruppe Instanzen in dieselbe Platzierungsgruppe wie die identifizierte Instance. Da die Startkonfigurationseinstellungen die Angabe einer Platzierungsgruppe nicht zulassen, wird die Platzierungsgruppe in das `PlacementGroup`-Attribut der neuen Auto-Scaling-Gruppe kopiert.

Die folgenden Konfigurationsdetails werden nicht von Ihrer identifizierten Instance übernommen:

- Speicher: Die Blockgeräte (EBS-Volumen und Instance-Speichervolumen) werden nicht von der identifizierten Instance kopiert. Stattdessen bestimmt die bei der Erstellung des AMI erstellte Blockgerät-Zuweisung, welche Geräte verwendet werden.
- Anzahl der Netzwerkschnittstellen: Die Netzwerkschnittstellen werden nicht von Ihrer identifizierten Instance kopiert. Stattdessen verwendet Amazon EC2 Auto Scaling die Standardeinstellungen, um eine Netzwerkschnittstelle zu erstellen, nämlich die primäre Netzwerkschnittstelle (eth0).

- Optionen für Instance-Metadaten: Die Einstellungen für die zugänglichen Metadaten, die Metadatenversion und das Sprunglimit für Token-Antworten werden nicht von der identifizierten Instance übernommen. Stattdessen verwendet Amazon EC2 Auto Scaling seine Standardeinstellungen. Weitere Informationen finden Sie unter Konfigurieren der Instance-Metadaten-Optionen.
- Lastenverteilung: Ist die identifizierte Instance bei mindestens einem Load Balancer angemeldet, werden die Informationen über den Load Balancer nicht automatisch in den Load Balancer oder in das Zielgruppenattribut der neuen Auto-Scaling-Gruppe kopiert.
- Tags: Verfügt die identifizierte Instance über Tags, werden diese nicht in das Attribut `Tags` der neuen Auto-Scaling-Gruppe kopiert.

Voraussetzungen

Die EC2-Instance muss die folgenden Kriterien erfüllen:

- Die Instance gehört keiner anderen Auto-Scaling-Gruppe an.
- Der Status der Instance lautet `running`.
- Das AMI zum Starten der Instance muss noch vorhanden sein.

Erstellen einer Auto-Scaling-Gruppe aus einer EC2-Instance (Konsole)

So erstellen Sie eine Auto-Scaling-Gruppe aus einer EC2-Instance

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Instances und dann eine Instance aus.
3. Wählen Sie Actions, Instance Settings und Attach to Auto Scaling Group aus.
4. Geben Sie auf der Seite Attach to Auto Scaling Group (An Auto-Scaling-Gruppe anhängen) für Auto Scaling group einen Namen für die Gruppe ein und wählen Sie anschließend Attach (Anhängen) aus.

   Nachdem die Instance angefügt wurde, wird sie als Teil der Auto-Scaling-Gruppe angesehen. Die neue Auto-Scaling-Gruppe wird mithilfe einer neuen Startkonfiguration mit demselben Namen erstellt, den Sie für die Auto-Scaling-Gruppe angegeben haben. Die Auto-Scaling-Gruppe hat eine gewünschte Kapazität und eine maximale Größe von 1.

5. (Optional) Klicken Sie zum Bearbeiten der Einstellungen der Auto-Scaling Gruppe im Navigationsbereich unter Auto Scaling (automatische Skalierung) auf Auto Scaling Groups (Auto-Scaling-Gruppen). Aktivieren Sie das Kontrollkästchen neben der neuen Auto-Scaling-Gruppe, wählen Sie die Schaltfläche Bearbeiten, die sich über der Liste der Gruppen befindet, ändern Sie die Einstellungen nach Bedarf und wählen Sie dann Aktualisieren aus.

Erstellen einer Auto-Scaling-Gruppe aus einer EC2-Instance (AWS CLI)

Das folgende Verfahren demonstriert die Verwendung eines CLI-Befehls zum Erstellen einer Auto-Scaling-Gruppe aus einer EC2-Instance.

Bei diesem Verfahren wird die Instance nicht zur Auto-Scaling-Gruppe hinzugefügt. Damit die Instance angefügt werden kann, müssen Sie den Befehl attach-instances ausführen, nachdem die Auto-Scaling-Gruppe erstellt wurde.

Bevor Sie beginnen, suchen Sie die ID der EC2-Instance mithilfe der Amazon-EC2-Konsole oder dem Befehl describe-instances.

So verwenden Sie die aktuelle Instance als Vorlage

- Verwenden Sie den folgenden create-auto-scaling-group-Befehl, um die Auto-Scaling-Gruppe `my-asg-from-instance` aus der EC2-Instance `i-0e69cc3f05f825f4f` zu erstellen.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg-from-instance \
  --instance-id i-0e69cc3f05f825f4f --min-size 1 --max-size 2 --desired-capacity 2
```

So prüfen Sie, dass Ihre Auto-Scaling-Gruppe eine neue Instance gestartet hat

- Verwenden Sie den folgenden describe-auto-scaling-groups-Befehl, um zu überprüfen, ob die Auto-Scaling-Gruppe erfolgreich erstellt wurde.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg-from-instance
```

  Die folgende Beispielantwort zeigt, dass die gewünschte Kapazität der Gruppe 2 beträgt, die Gruppe über zwei laufende Instances verfügt und die Startkonfiguration den Namen `my-asg-from-instance` hat.

```
{
  "AutoScalingGroups":[
    {
      "AutoScalingGroupName":"my-asg-from-instance",
      "AutoScalingGroupARN":"arn",
      "LaunchConfigurationName":"my-asg-from-instance",
      "MinSize":1,
      "MaxSize":2,
      "DesiredCapacity":2,
      "DefaultCooldown":300,
      "AvailabilityZones":[
        "us-west-2a"
      ],
      "LoadBalancerNames":[],
      "TargetGroupARNs":[],
      "HealthCheckType":"EC2",
      "HealthCheckGracePeriod":0,
      "Instances":[
        {
          "InstanceId":"i-06905f55584de02da",
          "InstanceType":"t2.micro",
          "AvailabilityZone":"us-west-2a",
          "LifecycleState":"InService",
          "HealthStatus":"Healthy",
          "LaunchConfigurationName":"my-asg-from-instance",
          "ProtectedFromScaleIn":false
        },
        {
          "InstanceId":"i-087b42219468eacde",
          "InstanceType":"t2.micro",
          "AvailabilityZone":"us-west-2a",
          "LifecycleState":"InService",
          "HealthStatus":"Healthy",
          "LaunchConfigurationName":"my-asg-from-instance",
          "ProtectedFromScaleIn":false
        }
      ],
      "CreatedTime":"2020-10-28T02:39:22.152Z",
      "SuspendedProcesses":[ ],
      "VPCZoneIdentifier":"subnet-6bea5f06",
      "EnabledMetrics":[ ],
      "Tags":[ ],
      "TerminationPolicies":[
```

```
        "Default"
      ],
      "NewInstancesProtectedFromScaleIn":false,
      "ServiceLinkedRoleARN":"arn",
      "TrafficSources":[]
    }
  ]
}
```

So zeigen Sie die Startkonfiguration an

- Verwenden Sie den folgenden describe-launch-configuration-Befehl, um die Details der Startkonfiguration anzuzeigen.

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-asg-from-instance
```

Das Folgende ist Ausgabebeispiel:

```
{
  "LaunchConfigurations":[
    {
      "LaunchConfigurationName":"my-asg-from-instance",
      "LaunchConfigurationARN":"arn",
      "ImageId":"ami-0528a5175983e7f28",
      "KeyName":"my-key-pair-uswest2",
      "SecurityGroups":[
        "sg-05eaec502fcdadc2e"
      ],
      "ClassicLinkVPCSecurityGroups":[ ],
      "UserData":"",
      "InstanceType":"t2.micro",
      "KernelId":"",
      "RamdiskId":"",
      "BlockDeviceMappings":[ ],
      "InstanceMonitoring":{
        "Enabled":true
      },
      "CreatedTime":"2020-10-28T02:39:22.321Z",
      "EbsOptimized":false,
      "AssociatePublicIpAddress":true
```

```
    }
  ]
}
```

Beenden der Instances

- Sie können die Instance beenden, wenn Sie sie nicht mehr benötigen. Der folgende terminate-instances-Befehl beendet die Instance `i-0e69cc3f05f825f4f`.

```
aws ec2 terminate-instances --instance-ids i-0e69cc3f05f825f4f
```

  Wenn Sie eine Amazon EC2-Instance beenden, kann diese nicht neu gestartet werden. Nach dem Beenden sind die Daten nicht mehr vorhanden und das Volume kann nicht an eine Instance angefügt werden. Weitere Informationen zum Beenden von Instances finden Sie unter Eine Instance beenden im Amazon EC2 EC2-Benutzerhandbuch.

Aktualisieren einer Auto-Scaling-Gruppe

Sie können die meisten Details Ihrer Auto-Scaling-Gruppe aktualisieren. Sie können den Namen einer Auto Scaling Scaling-Gruppe nicht aktualisieren oder ändern AWS-Region.

So aktualisieren Sie eine Auto-Scaling-Gruppe (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie Ihre Auto-Scaling-Gruppe aus, um Informationen über die Gruppe anzuzeigen, mit Registerkarten für Details, Aktivität, Automatische Skalierung, Instance-Verwaltung, Überwachung und Instance-Aktualisierung.
3. Wählen Sie die Registerkarten für die gewünschten Konfigurationsbereiche aus und aktualisieren Sie die Einstellungen nach Bedarf. Wählen Sie für jede Einstellung, die Sie bearbeiten, Aktualisieren aus, um Ihre Änderungen an der Konfiguration der Auto-Scaling-Gruppe zu speichern.

   - Registerkarte Details

     Dies sind die allgemeinen Einstellungen für Ihre Auto-Scaling-Gruppe. Sie können diese auf dieselbe Weise bearbeiten und verwalten wie bei der Erstellung der Auto-Scaling-Gruppe.

Der Abschnitt Erweiterte Konfigurationen enthält einige Optionen, die bei der Erstellung der Gruppe nicht verfügbar sind, z. B. Beendigungsrichtlinien, Abkühlungsphase, Ausgesetzte Prozesse und Maximale Instance-Lebensdauer. Sie können auch die Platzierungsgruppe und die serviceverknüpfte Rolle der Auto-Scaling-Gruppe anzeigen, aber nicht bearbeiten.

Wenn die Gruppe mit Elastic Load Balancing-Ressourcen verbunden ist, lesen Sie vor dem Ändern der Availability Zones bitte Hinzufügen oder Entfernen von Availability Zones. Einige Einschränkungen auf dem Load Balancer können Sie daran hindern, Änderungen an den Availability Zones Ihrer Gruppe auf die Availability Zones Ihres Load Balancers anzuwenden.

- Registerkarte Aktivität
  - Aktivitätsbenachrichtigungen — Amazon SNS SNS-Benachrichtigungen
- Registerkarte Automatische Skalierung
  - Dynamische Skalierungsrichtlinien — Dynamische Skalierungsrichtlinien
  - Richtlinien für vorausschauende Skalierung — Richtlinien für prädiktive Skalierung
  - Geplante Aktionen — Geplante Aktionen
- Registerkarte Instance-Verwaltung
  - Lebenszyklus-Hooks — Lebenszyklus-Hooks
  - Warmer Pool — Warme Pools
- Registerkarte Überwachung
  - Auf dieser Registerkarte gibt es nur eine einzige Option, mit der Sie die Erfassung von CloudWatch Gruppenmetriken aktivieren oder deaktivieren können.

Aktualisieren Sie wie folgt eine Auto-Scaling-Gruppe über die Befehlszeile:

Verwenden Sie einen der folgenden Befehle:

- update-auto-scaling-group (AWS CLI)
- Als AutoScaling Gruppe aktualisieren ()AWS Tools for Windows PowerShell

Aktualisieren von Auto-Scaling-Instances

Wenn Sie eine neue Startvorlage oder Startkonfiguration mit einer Auto-Scaling-Gruppe verknüpfen, erhalten alle neuen Instances die aktualisierte Konfiguration. Vorhandene Instances werden weiterhin

mit der Konfiguration ausgeführt, mit der sie ursprünglich gestartet wurden. Um Ihre Änderungen auf vorhandene Instances anzuwenden, haben Sie die folgenden Möglichkeiten:

- Starten Sie eine Instance-Aktualisierung, um die älteren Instances zu ersetzen. Weitere Informationen finden Sie unter Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren.
- Warten Sie auf Skalierungsaktivitäten, um ältere Instances allmählich durch neuere Instances auf der Grundlage Ihrer Beendigungsrichtlinien zu ersetzen.
- Beenden Sie diese manuell, damit sie durch Ihre Auto-Scaling-Gruppe ersetzt werden.

> **Note**
>
> Sie können die folgenden Instance-Attribute ändern, indem Sie sie als Teil der Startvorlage oder Startkonfiguration angeben:
>
> - Amazon Machine Image (AMI)
> - Blockgeräte
> - Schlüsselpaar
> - Instance-Typ
> - Sicherheitsgruppen
> - Benutzerdaten
> - Überwachung
> - IAM-Instance-Profil
> - Placement Tenancy
> - kernel
> - Ramdisk
> - gibt an, ob Sie eine öffentliche IP-Adresse haben.

Tagging von Auto-Scaling-Gruppen und Instances

Ein Tag ist eine benutzerdefinierte Attributbezeichnung, die Sie einer Ressource zuweisen oder die einer AWS Ressource zugewiesen wird. AWS Jedes Tag besteht aus zwei Teilen:

- einem Tag-Schlüssel (z. B. `costcenter`, `environment` oder `project`)

- einem optionalen Feld, dem sogenannten Tag-Wert (z. B. `111122223333` oder `production`)

Tags sind für folgende Aktivitäten nützlich:

- Verfolgen Sie Ihre AWS Kosten. Sie aktivieren diese Tags auf dem AWS Billing and Cost Management Dashboard. AWS verwendet die Tags, um Ihre Kosten zu kategorisieren und Ihnen einen monatlichen Kostenverteilungsbericht zu senden. Weitere Informationen finden Sie unter Verwendung von Tags zur Kostenzuordnung im Benutzerhandbuch zu AWS Billing .
- Steuern Sie den Zugriff auf Auto Scaling-Ressourcen basierend auf Tags. Sie können Bedingungen in Ihren IAM-Richtlinien zum Steuern des Zugriffs auf Auto-Scaling-Gruppen auf Basis der Tags für diese Gruppe verwenden. Weitere Informationen finden Sie unter Tags für Sicherheit.
- Filtern und suchen Sie nach Auto-Scaling-Gruppen anhand der von Ihnen hinzugefügten Tags. Weitere Informationen finden Sie unter Verwenden Sie Tags, um Auto-Scaling-Gruppen zu filtern.
- Identifizieren und organisieren Sie Ihre AWS Ressourcen. Viele AWS-Services unterstützen Tagging, sodass Sie Ressourcen aus verschiedenen Diensten dasselbe Tag zuweisen können, um anzuzeigen, dass die Ressourcen miteinander verknüpft sind.

Sie können neue oder vorhandene Auto-Scaling-Gruppen markieren. Sie können auch Tags aus einer Auto-Scaling-Gruppe an die von ihr gestarteten EC2-Instances weitergeben.

Tags werden nicht an Amazon EBS-Volumes verbreitet. Um Tags zu Amazon EBS-Volumes hinzuzufügen, geben Sie die Tags in einer Startvorlage an. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.

Sie können Tags über die SDKs AWS Management Console AWS CLI, oder erstellen und verwalten.

Inhalt

Einschränkungen für die Tag-Benennung und -Nutzung

Die folgenden grundlegenden Einschränkungen gelten für Tags (Markierungen):

- Die maximale Anzahl an Tags pro Ressource beträgt 50.
- Die maximale Anzahl an Tags, die Sie mit einem einzigen Aufruf hinzufügen oder entfernen können, beträgt 25.
- Die maximale Schlüssellänge beträgt 128 Unicode-Zeichen.
- Die maximale Wertlänge beträgt 256 Unicode-Zeichen.
- Bei Tag-Schlüsseln und -Werten muss die Groß-/Kleinschreibung beachtet werden. Eine bewährte Methode besteht darin, sich für eine einheitliche Schreibweise der Tag-Benennungen zu entscheiden und diese Strategie für alle Ressourcentypen umzusetzen.
- Verwenden Sie das `aws:` Präfix nicht in Ihren Tagnamen oder -Werten, da es für die AWS Verwendung reserviert ist. Sie können Tag-Namen oder -Werte mit diesem Präfix nicht bearbeiten oder löschen und sie werden nicht zu Ihren Tags pro Ressourcenkontingent gezählt.

Tagging-Lebenszyklus von EC2-Instances

Wenn Sie sich für die Weitergabe von Tags an Ihre EC2-Instances entschieden haben, werden die Tags wie folgt gehandhabt:

- Wenn eine Auto-Scaling-Gruppe Instances startet, fügt sie diesen während der Ressourcenerstellung Tags hinzu, nicht nach der Erstellung der Ressource.
- Die Auto-Scaling-Gruppe fügt den Instances automatisch einen Tag mit dem Schlüssel `aws:autoscaling:groupName` und einen Wert des Namens der Auto-Scaling-Gruppe hinzu.
- Wenn Sie Instance-Tags in Ihrer Startvorlage angeben und sich dafür entschieden haben, die Tags Ihrer Gruppe an ihre Instances zu übertragen, werden alle Tags zusammengeführt. Wenn derselbe Tag-Schlüssel für einen Tag in Ihrer Startvorlage und einen Tag in Ihrer Auto-Scaling-Gruppe angegeben wird, hat der Tag-Wert aus der Gruppe Vorrang.
- Beim Anfügen vorhandener Instances fügt die Auto-Scaling-Gruppe die Tags den Instances hinzu. Hierbei werden alle vorhandenen Tags mit demselben Tag-Schlüssel überschrieben. Zusätzlich fügt sie einen Tag mit `aws:autoscaling:groupName` als Schlüssel und mit dem Namen der Auto-Scaling-Gruppe als Wert hinzu.
- Beim Trennen einer Instance von einer Auto-Scaling-Gruppe entfernt sie nur das `aws:autoscaling:groupName`-Tag.

Markieren Ihrer Auto-Scaling-Gruppen

Beim Hinzufügen eines Tags zu einer Auto-Scaling-Gruppe können Sie angeben, ob es zu gestarteten Instances in der Auto-Scaling-Gruppe hinzugefügt werden soll. Nach der Änderung eines Tags wird neuen Instances in der Auto-Scaling-Gruppe die aktualisierte Version des Tags hinzugefügt. Bei der Erstellung oder Änderung eines Tags einer Auto-Scaling-Gruppe werden diese Änderungen an den bereits gestarteten Instances der Auto-Scaling-Gruppe nicht vorgenommen.

Inhalt

Hinzufügen oder Ändern von Tags (Konsole)

So markieren Sie eine Auto-Scaling-Gruppe bei der Erstellung

Wenn Sie eine Auto-Scaling-Gruppe mit der Amazon EC2-Konsole erstellen, können Sie auf der Seite Add Tags (Tags hinzufügen) im Assistenten zum Erstellen von Auto-Scaling-Gruppen Tag-Schlüssel und -Werte angeben. Zum Übertragen eines Tags an die in der Auto-Scaling-Gruppe gestarteten Instances achten Sie darauf, dass die Option Tag New Instances (Neue Instances markieren) für das Tag ausgewählt bleibt. Andernfalls können Sie sie deaktivieren.

Hinzufügen oder Ändern von Tags für eine vorhandene Auto-Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option Tags, Bearbeiten.
4. Bearbeiten Sie zum Ändern bestehender Tags die Werte Key und Value.
5. Um ein neues Tag hinzuzufügen, wählen Sie Add tag aus und bearbeiten Sie Key und Value. Lassen Sie Tag new instances (Neue Instances markieren) aktiviert, damit das Tag zu in der Auto-Scaling-Gruppe gestarteten Instances automatisch hinzugefügt wird, oder deaktivieren Sie die Option, falls dies nicht erwünscht ist.
6. Wenn Sie mit dem Hinzufügen der Tags fertig sind, wählen Sie Update (Aktualisieren).

Hinzufügen oder Ändern von Tags (AWS CLI)

Die folgenden Beispiele zeigen, wie Sie Tags hinzufügen, wenn Sie Auto Scaling Scaling-Gruppen erstellen, und wie Sie Tags für bestehende Auto Scaling Scaling-Gruppen hinzufügen oder ändern können. AWS CLI

So markieren Sie eine Auto-Scaling-Gruppe bei der Erstellung

Verwenden Sie den create-auto-scaling-group-Befehl, um eine neue Auto-Scaling-Gruppe zu erstellen und der Auto-Scaling-Gruppe ein Tag hinzuzufügen, z. B. **`environment=production`**. Das Tag wird auch jeder gestarteten Instance in der Auto-Scaling-Gruppe hinzugefügt.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-configuration-name my-launch-config --min-size 1 --max-size 3 \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
  --tags Key=environment,Value=production,PropagateAtLaunch=true
```

Erstellen oder Ändern von Tags für eine vorhandene Auto-Scaling-Gruppe

Verwenden Sie den create-or-update-tags-Befehl, um ein Tag zu erstellen oder zu ändern. Der folgende Befehl fügt z. B. die Tags **`Name=my-asg`** und **`costcenter=cc123`** hinzu. Die Tags werden nach dieser Änderung auch jeder gestarteten Instance in der Auto-Scaling-Gruppe hinzugefügt. Ist ein Tag mit einem dieser Schlüssel bereits vorhanden, wird das vorhandene Tag ersetzt. Die Amazon EC2-Konsole ordnet den Anzeigenamen für jede Instance dem Namen zu, der für den `Name`-Schlüssel angegeben ist (unter Beachtung der Groß-/Kleinschreibung).

```
aws autoscaling create-or-update-tags \
  --tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Name,Value=my-
asg,PropagateAtLaunch=true \
  ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=costcenter,Value=cc123,PropagateAtLaunch=true
```

Beschreiben der Tags für eine Auto-Scaling-Gruppe (AWS CLI)

Wenn Sie die Tags anzeigen möchten, die auf eine bestimmte Auto-Scaling-Gruppe angewendet werden, können Sie entweder einen der folgenden Befehle verwenden:

- describe-tags — Sie geben Ihren Auto Scaling Scaling-Gruppennamen ein, um eine Liste der Tags für die angegebene Gruppe anzuzeigen.

```
aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "Tags": [
        {
            "ResourceType": "auto-scaling-group",
            "ResourceId": "my-asg",
            "PropagateAtLaunch": true,
            "Value": "production",
            "Key": "environment"
        }
    ]
}
```

- describe-auto-scaling-groups — Sie geben Ihren Auto Scaling Scaling-Gruppennamen an, um die Attribute der angegebenen Gruppe, einschließlich aller Tags, anzuzeigen.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-0b97f1e282EXAMPLE",
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "MinSize": 1,
            "MaxSize": 5,
            "DesiredCapacity": 1,
            ...
            "Tags": [
                {
```

```
                    "ResourceType": "auto-scaling-group",
                    "ResourceId": "my-asg",
                    "PropagateAtLaunch": true,
                    "Value": "production",
                    "Key": "environment"
                }
            ],
            ...
        }
    ]
}
```

Löschen von Tags

Sie können Tags, die einer Auto-Scaling-Gruppe zugeordnet sind, jederzeit löschen.

Inhalt

- Löschen von Tags (Konsole)
- Löschen von Tags (AWS CLI)

Löschen von Tags (Konsole)

So löschen Sie ein Tag

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben einer vorhandenen Gruppe.

   Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option Tags, Bearbeiten.
4. Wählen Sie Remove (Entfernen) neben dem Tag.
5. Wählen Sie Aktualisieren.

Löschen von Tags (AWS CLI)

Verwenden Sie den Befehl delete-tags, um ein Tag zu löschen. Mit dem folgenden Befehl wird beispielsweise ein Tag mit dem Schlüssel **environment** gelöscht.

```
aws autoscaling delete-tags --tags "ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=environment"
```

Sie müssen den Tag-Schlüssel angeben, nicht aber den Wert. Wenn Sie einen Wert angeben und der Wert nicht korrekt ist, wird das Tag nicht gelöscht.

Tags für Sicherheit

Verwenden Sie Tags, um zu überprüfen, ob der Anforderer (z. B. ein IAM-Benutzer oder eine IAM-Rolle) über Berechtigungen zum Erstellen, Ändern oder Löschen bestimmter Auto-Scaling-Gruppen verfügt. Geben Sie Tag-Informationen im Bedingungselement einer IAM-Richtlinie mithilfe eines oder mehrerer der folgenden Bedingungsschlüssel an:

- Verwenden Sie `autoscaling:ResourceTag/`*`tag-key`*`: `*`tag-value`*, um Benutzeraktionen für Auto Scaling-Gruppen 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.

Sie könnten beispielsweise den Zugriff auf alle Auto-Scaling-Gruppen, die ein Tag mit dem Schlüssel **environment** und dem Wert **production** enthalten, wie im folgenden Beispiel verweigern.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:DeleteAutoScalingGroup"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {"autoscaling:ResourceTag/environment": "production"}
            }
        }
    ]
}
```

Weitere Informationen über die Verwendung von Bedingungsschlüsseln zur Kontrolle des Zugriffs auf Auto-Scaling-Gruppen finden Sie unter Funktionsweise von Amazon EC2 Auto Scaling mit IAM.

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 Auto-Scaling-Gruppen verfügt.

Die folgende Beispiel-IAM-Richtlinie gibt dem Prinzipal die Berechtigung, nur das Tag mit dem **`temporary`**-Schlüssel aus Auto-Scaling-Gruppen zu entfernen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "autoscaling:DeleteTags",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": { "aws:TagKeys": ["temporary"] }
            }
        }
    ]
}
```

Weitere Beispiele für IAM-Richtlinien, die Einschränkungen für die für Auto-Scaling-Gruppen angegebenen Tags erzwingen, finden Sie unter Steuern, welche Tag-Schlüssel und Tag-Werte verwendet werden können.

> **Note**
>
> Selbst wenn Sie eine Richtlinie haben, die Ihre Benutzer daran hindert, einen Tagging-Vorgang für eine Auto-Scaling-Gruppe durchzuführen (oder rückgängig zu machen), bedeutet

dies nicht, dass sie die Tags der Instances nach dem Start manuell ändern können. Beispiele zur Steuerung des Zugriffs auf Tags auf EC2-Instances finden Sie unter Beispiel: Tagging resources im Amazon EC2 EC2-Benutzerhandbuch.

Verwenden Sie Tags, um Auto-Scaling-Gruppen zu filtern

Die folgenden Beispiele zeigen Ihnen, wie Sie Filter mit dem Befehl describe-auto-scaling-groups verwenden können, um Auto-Scaling-Gruppen mit bestimmten Tags zu beschreiben. Das Filtern nach Tags ist auf das AWS CLI oder ein SDK beschränkt und nicht über die Konsole verfügbar.

Überlegungen zum Filtern

- Sie können mehrere Filter und mehrere Filterwerte in einer einzelnen Anforderung angeben.
- Sie können Platzhalter in den Filterwerten nicht verwenden.
- Bei Filterwerten muss die Groß- und Kleinschreibung beachtet werden.

Beispiel: Beschreiben Sie Auto-Scaling-Gruppen mit einem bestimmten Tag-Schlüssel und Wertepaar

Der folgende Befehl zeigt, wie Sie die Ergebnisse so filtern, dass nur Auto-Scaling-Gruppen mit dem Tag-Schlüssel und dem Wertepaar **`environment=production`** angezeigt werden.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-key,Values=environment Name=tag-value,Values=production
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-0b97f1e282EXAMPLE",
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "MinSize": 1,
```

```
            "MaxSize": 5,
            "DesiredCapacity": 1,
            ...
            "Tags": [
                {
                    "ResourceType": "auto-scaling-group",
                    "ResourceId": "my-asg",
                    "PropagateAtLaunch": true,
                    "Value": "production",
                    "Key": "environment"
                }
            ],
            ...
        },

    ... additional groups ...

    ]
}
```

Alternativ können Sie auch Tags mit einem `tag:<key>`-Filter angeben. Der folgende Befehl zeigt zum Beispiel, wie Sie die Ergebnisse filtern können, um nur Auto-Scaling-Gruppen mit dem Tag-Schlüssel und dem Wertepaar **`environment=production`** anzuzeigen. Dieser Filter ist wie folgt formatiert: `Name=tag:<key>,Values=<value>`, wobei **`<key>`** und **`<value>`** ein Tag-Schlüssel- und Wertepaar darstellen.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag:environment,Values=production
```

Sie können die AWS CLI Ausgabe auch mithilfe der `--query` Option filtern. Das folgende Beispiel zeigt, wie die AWS CLI Ausgabe für den vorherigen Befehl nur auf den Gruppennamen, die Mindestgröße, die Maximalgröße und die gewünschten Kapazitätsattribute beschränkt werden kann.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag:environment,Values=production \
  --query "AutoScalingGroups[].{AutoScalingGroupName: AutoScalingGroupName, MinSize:
 MinSize, MaxSize: MaxSize, DesiredCapacity: DesiredCapacity}"
```

Nachfolgend finden Sie eine Beispielantwort.

```
[
```

```
    {
        "AutoScalingGroupName": "my-asg",
        "MinSize": 0,
        "MaxSize": 10,
        "DesiredCapacity": 1
    },

    ... additional groups ...

]
```

Weitere Informationen zum Filtern finden Sie im AWS Command Line Interface Benutzerhandbuch unter Filtern der AWS CLI Ausgabe.

Beispiel: Beschreiben Sie Auto-Scaling-Gruppen mit Tags, die mit dem angegebenen Tag-Schlüssel übereinstimmen

Der folgende Befehl zeigt, wie Sie die Ergebnisse so filtern, dass nur Auto-Scaling-Gruppen mit dem Tag angezeigt werden, unabhängig vom Wert des **`environment`**-Tags.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-key,Values=environment
```

Beispiel: Beschreiben Sie Auto-Scaling-Gruppen mit Tags, die dem angegebenen Satz von Tag-Schlüsseln entsprechen

Der folgende Befehl zeigt, wie Sie die Ergebnisse so filtern, dass nur Auto-Skalierungsgruppen mit Tags für **`environment`** und **`project`** angezeigt werden, unabhängig von den Tag-Werten.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-key,Values=environment Name=tag-key,Values=project
```

Beispiel: Beschreiben Sie Auto-Scaling-Gruppen mit Tags, die mindestens einem der angegebenen Tag-Schlüssel entsprechen

Der folgende Befehl zeigt, wie Sie die Ergebnisse so filtern, dass nur Auto-Scaling-Gruppen mit Tags für **`environment`** oder **`project`** angezeigt werden, unabhängig von den Tag-Werten.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-key,Values=environment,project
```

Beispiel: Beschreiben Sie Auto-Scaling-Gruppen mit dem angegebenen Tag-Wert

Der folgende Befehl zeigt, wie Sie die Ergebnisse so filtern, dass nur Auto-Scaling-Gruppen mit einem Tag-Wert von **`production`** angezeigt werden, unabhängig vom Tag-Schlüssel.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-value,Values=production
```

Beispiel: Beschreiben Sie Auto-Scaling-Gruppen mit den angegebenen Tag-Werten

Der folgende Befehl zeigt, wie Sie die Ergebnisse so filtern, dass nur Auto-Scaling-Gruppen mit den Tag-Werten **`production`** und **`development`** angezeigt werden, unabhängig vom Tagschlüssel.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-value,Values=production Name=tag-value,Values=development
```

Beispiel: Beschreiben Sie Auto-Scaling-Gruppen mit Tags, die mindestens einem der angegebenen Tag-Werte entsprechen

Der folgende Befehl zeigt, wie Sie die Ergebnisse filtern können, um nur Auto-Scaling-Gruppen mit einem Tag-Wert von **`production`** oder **`development`** anzuzeigen, unabhängig vom Tag-Schlüssel.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-value,Values=production,development
```

Beispiel: Beschreiben Sie Auto-Scaling-Gruppen mit Tags, die mehreren Tag-Schlüsseln und Werten entsprechen

Sie können auch Filter kombinieren, um benutzerdefinierte AND- und OR-Logik zu erstellen und so eine komplexere Filterung durchzuführen.

Der folgende Befehl zeigt, wie Sie die Ergebnisse filtern können, um nur Auto-Scaling-Gruppen mit einer bestimmten Gruppe von Tags anzuzeigen. Ein Tag-Schlüssel ist **`environment`** AND der Tag-Wert ist (**`production`** OR **`development`**) AND der andere Tag-Schlüssel ist **`costcenter`** AND der Tag-Wert ist **`cc123`**.

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag:environment,Values=production,development
 Name=tag:costcenter,Values=cc123
```

Wartungsrichtlinien für Instances

Sie können eine Instance-Wartungsrichtlinie für Ihre Auto-Scaling-Gruppe konfigurieren, um bestimmte Kapazitätsanforderungen bei Ereignissen zu erfüllen, die dazu führen, dass Instances ersetzt werden, z. B. bei einer Instance-Aktualisierung oder bei der Zustandsprüfung.

Nehmen wir an, Sie verfügen über eine Auto-Scaling-Gruppe, die eine geringe Anzahl von Instances aufweist. Sie möchten vermeiden, dass es zu möglichen Störungen kommt, wenn eine Instance beendet und anschließend ersetzt wird, sobal Zustandsprüfungen auf eine beeinträchtigte Instance hinweisen. Mit einer Instance-Wartungsrichtlinie können Sie sicherstellen, dass Amazon EC2 Auto Scaling zuerst eine neue Instance startet und dann wartet, bis sie vollständig bereit ist, bevor es die fehlerhafte Instance beendet.

Eine Instance-Wartungsrichtlinie hilft Ihnen auch dabei, mögliche Störungen zu minimieren, wenn mehrere Instances gleichzeitig ersetzt werden. Sie legen die minimalen und maximalen gesunden Prozentwerte für die Richtlinie fest, und Ihre Auto-Scaling-Gruppe kann die Kapazität nur innerhalb dieses minimalen und maximalen Bereichs erhöhen und verringern, wenn Instances ersetzt werden. Ein größerer Bereich erhöht die Anzahl der Instances, die gleichzeitig ausgetauscht werden können.

Inhalt

Überblick über die Instance-Wartungsrichtlinien

Dieses Thema bietet einen Überblick über die verfügbaren Optionen und beschreibt, was zu beachten ist, wenn Sie eine Instance-Wartungsrichtlinie erstellen.

Inhalt

Übersicht

Wenn Sie eine Instance-Wartungsrichtlinie für Ihre Auto-Scaling-Gruppe erstellen, wirkt sich die Richtlinie auf die Ergebnisse von Amazon EC2 Auto Scaling aus, die dazu führen, dass Instances ersetzt werden. Dies führt zu einem konsistenteren Austauschverhalten innerhalb derselben Auto-Scaling-Gruppe. Außerdem können Sie Verfügbarkeit oder Kosten für Ihre Gruppe je nach Bedarf optimieren.

In der Konsole stehen die folgenden Konfigurationsoptionen zur Verfügung:

- Vor dem Beenden starten – Eine neue Instance muss zuerst bereitgestellt werden, bevor eine bestehende Instance beendet werden kann. Dieser Ansatz ist eine gute Wahl für Anwendungen, bei denen Verfügbarkeit wichtiger ist als Kosteneinsparungen.
- Beenden und starten – Neue Instances werden zur gleichen Zeit bereitgestellt, wie Ihre bestehenden Instances beendet werden. Dieser Ansatz ist eine gute Wahl für Anwendungen, bei denen Kosteneinsparungen wichtiger sind als die Verfügbarkeit. Es ist auch eine gute Wahl für Anwendungen, die nicht mehr Kapazität benötigen, als derzeit verfügbar ist, selbst wenn Instances ersetzt werden.
- Benutzerdefinierte Richtlinie – Mit dieser Option können Sie für Ihre Richtlinie einen benutzerdefinierten Mindest- und Höchstbereich für die Kapazität einrichten, die beim Austausch von Instances verfügbar sein soll. Dieser Ansatz kann Ihnen helfen, das richtige Gleichgewicht zwischen Kosten und Verfügbarkeit zu finden.

Die Standardeinstellung für eine Auto-Scaling-Gruppe ist, dass sie keine Instance-Wartungsrichtlinie hat, was dazu führt, dass sie auf Instance-Wartungsereignisse mit dem Standardverhalten reagiert. Das Standardverhalten wird in der folgenden Tabelle beschrieben.

Standardverhalten bei Instance-Wartungsereignissen

Ereignis	Beschreibung	Standardverhalten
Fehlgeschlagene Zustandsprüfung	Passiert automatisch, wenn Instances ihre Zustandsprüfungen nicht bestehen. Amazon EC2 Auto Scaling ersetzt die Instances, die ihre Zustandsprüfungen nicht bestehen. Informationen zu	Beenden und starten.

Ereignis	Beschreibung	Standardverhalten
	den Ursachen für fehlgesch lagene Zustandsprüfungen finden Sie unter Zustandsp rüfungen für Instances in einer Auto-Scaling-Gruppe.	
Instance-Aktualisierung	Das geschieht, wenn Sie eine Instance-Aktualisierung starten. Abhängig von Ihrer Konfiguration kann eine Instance-Aktualisierung eine einzelne Instance, mehrere Instances auf einmal oder alle auf einmal ersetzen. Weitere Informationen finden Sie unter Verwenden Sie eine Instanzak tualisierung, um Instances in einer Auto Scaling Scaling-G ruppe zu aktualisieren.	Beenden und starten.
Maximale Lebensdauer von Instances	Das passiert automatis ch, wenn Instances die maximale Instance-Lebensdau er erreichen, die Sie für Ihre Auto-Scaling-Gruppe angegeben haben. Amazon EC2 Auto Scaling ersetzt solche Instances, die ihre maximale Instance-Lebensdau er erreichen. Weitere Informati onen finden Sie unter Auto-Scaling-Instances basierend auf der maximalen Instance-Lebensdauer ersetzen.	Beenden und starten.

Ereignis	Beschreibung	Standardverhalten
Neuausgleich	Dies erfolgt automatisch, wenn grundlegende Änderungen vorliegen, die dazu führen, dass die Gruppe aus dem Gleichgewicht gerät. Amazon EC2 Auto Scaling führt in den folgenden Situationen einen Neuausgleich für die Gruppe aus: • Eine Availability Zone, die zuvor zu wenig Kapazität hatte, wurde wiederher gestellt, oder Sie fügen der Gruppe eine Availability Zone hinzu oder entfernen sie aus ihr. In diesem Fall versucht Ihre Auto-Scaling-Gruppe, sich gleichmäß ig über die Availability Zones zu verteilen. Weitere Informationen finden Sie unter Wiederherstellen des Gleichgewichts von Aktivität en. • Sie aktivieren den Kapazität sausgleich in Ihrer Auto-Scaling-Gruppe, und sie versucht, neue Spot Instances zu starten, bevor vorhandene unterbroc hen werden, wenn sich die Verfügbarkeit von Spot Instances ändert. Weitere Informationen	Vor dem Beenden starten. Amazon EC2 Auto Scaling kann die Größenlimits Ihrer Gruppe um bis zu 10 Prozent der maximalen Kapazität überschreiten. Wenn Sie den Kapazitätsausgleich verwenden, können diese Grenzwerte jedoch nur um bis zu 10 Prozent der gewünscht en Kapazität überschritten werden.

Ereignis	Beschreibung	Standardverhalten
	finden Sie unter Verwenden des Kapazitätsausgleichs, um Amazon-EC2-Spot-Unterbrechungen zu behandeln. • Sie aktualisieren Ihre Auto-Scaling-Gruppe und sie ersetzt nach und nach Instances, um sie an die neuen Kaufoptionen anzupassen, die Sie bei der Aktualisierung einer Richtlinie für gemischte Instances ausgewählt haben. Weitere Informationen finden Sie unter Aktualisieren einer Auto-Scaling-Gruppe.	

Amazon EC2 Auto Scaling wird in den folgenden Situationen weiterhin standardmäßig beendet und gestartet. Wenn eine dieser Situationen eintritt, liegt die Kapazität Ihrer Gruppe daher u. U. unter dem unteren Schwellenwert Ihrer Instance-Wartungsrichtlinie.

- Wenn eine Instance unerwartet beendet wird, z. B. aufgrund menschlichen Eingreifens. Amazon EC2 Auto Scaling ersetzt sofort Instances, die nicht mehr ausgeführt werden. Weitere Informationen finden Sie unter Zustandsprüfungen von Amazon EC2.
- Wenn Amazon EC2 eine Instance im Rahmen eines geplanten Ereignisses neu startet, stoppt oder außer Betrieb setzt, bevor Amazon EC2 Auto Scaling die Ersatz-Instance starten kann. Weitere Informationen zu diesen Ereignissen finden Sie unter Geplante Ereignisse für Ihre Instances im Amazon EC2 EC2-Benutzerhandbuch.
- Wenn der Amazon EC2 Spot Service eine Spot-Instance-Unterbrechung einleitet und anschließend das Beenden einer Spot Instance erzwungen wird.

Wenn Sie bei Spot Instances den Kapazitätsausgleich in Ihrer Auto-Scaling-Gruppe aktiviert haben, kann es sein, dass die Instance bereits eine anhängige Instance aus einem anderen Spot-Pool hat, die gestartet wurde, bevor die Spot-Unterbrechung eingeleitet wurde. Weitere Informationen darüber, wie der Kapazitätsausgleich funktioniert, finden Sie unter Verwenden des Kapazitätsausgleichs, um Amazon-EC2-Spot-Unterbrechungen zu behandeln.

Da jedoch nicht garantiert werden kann, dass Spot Instances verfügbar bleiben, und sie mit einer zweiminütigen Benachrichtigung über die Unterbrechung der Spot Instance beendet werden können, kann der untere Schwellenwert Ihrer Instance-Wartungsrichtlinie überschritten werden, wenn Instances unterbrochen werden, bevor Ihre neuen Instances gestartet wurden.

Schlüsselkonzepte

Bevor Sie beginnen, sollten Sie sich mit folgenden Kernkonzepten und der Terminologie vertraut machen:

Gewünschte Kapazität

: Die gewünschte Kapazität stellt die Kapazität der Auto-Scaling-Gruppe zum Zeitpunkt der Erstellung dar. Dies ist auch die Kapazität, die die Gruppe aufrechtzuerhalten versucht, wenn keine Skalierungsbedingungen an die Gruppe angehängt wurden.

Instance-Wartungsrichtlinie

: Eine Instance-Wartungsrichtlinie steuert, ob eine Instance zuerst bereitgestellt wird, bevor eine vorhandene Instance wegen eines Instance-Wartungsereignisses beendet wird. Sie bestimmt auch, wie weit Ihre Auto-Scaling-Gruppe Ihre gewünschte Kapazität unterschreiten und überschreiten kann, um mehrere Instances gleichzeitig zu ersetzen.

Maximaler fehlerfreier Prozentsatz

: Der maximale fehlerfreie Prozentsatz ist der Prozentsatz der gewünschten Kapazität, auf den Ihre Auto-Scaling-Gruppe beim Austausch von Instances erhöhen kann. Dies stellt den maximalen Prozentsatz der Gruppe dar, der zur Unterstützung Ihrer Workload in Betrieb und fehlerfrei oder ausstehend sein kann. In der Konsole können Sie den maximalen fehlerfreien Prozentsatz festlegen, wenn Sie entweder die Option Vor dem Beenden starten oder Benutzerdefinierte Richtlinie verwenden. Die gültigen Werte lauten 100–200 Prozent.

Minimaler fehlerfreier Prozentsatz

: Der minimale fehlerfreie Prozentsatz ist der Prozentsatz der gewünschten Kapazität, die beim Austausch von Instances betriebsbereit, fehlerfrei und einsatzbereit zur Unterstützung Ihrer

Arbeitslast bleiben soll. Eine Instance gilt als fehlerfrei und einsatzbereit, wenn sie ihren ersten Integritätstest erfolgreich abgeschlossen hat und die angegebene Aufwärmzeit verstrichen ist. In der Konsole können Sie den minimalen fehlerfreien Prozentsatz festlegen, wenn Sie entweder die Option Beenden und starten oder Benutzerdefinierte Richtlinie verwenden. Die gültigen Werte lauten 0–100 Prozent.

> **Note**
>
> Um Instances schneller zu ersetzen, können Sie einen niedrigen Wert für den minimalen fehlerfreien Prozentsatz angeben. Wenn jedoch nicht genügend fehlerfreie Instanzen laufen, kann die Verfügbarkeit verringert werden. Wir empfehlen, einen angemessenen Wert auszuwählen, um die Verfügbarkeit in Situationen aufrechtzuerhalten, in denen mehrere Instances ersetzt werden.

Instance-Aufwärmphase

Wenn Ihre Instances nach dem Eintritt in den Status `InService` Zeit für die Initialisierung benötigen, aktivieren Sie die standardmäßige Instance-Aufwärmphase für Ihre Auto-Scaling-Gruppe. Mit der standardmäßigen Instance-Aufwärmphase können Sie verhindern, dass Instances auf den minimalen fehlerfreien Prozentsatz angerechnet werden, bevor sie bereit sind. Dadurch wird sichergestellt, dass Amazon EC2 Auto Scaling berücksichtigt, wie lange es dauert, bis genügend Kapazität zur Unterstützung der Workload vorhanden ist, bevor vorhandene Instances beendet werden.

Als zusätzlichen Vorteil können Sie die CloudWatch Amazon-Metriken, die für die dynamische Skalierung verwendet werden, verbessern, wenn Sie das Standard-Instance-Warmup aktivieren. Wenn Ihre Auto Scaling Scaling-Gruppe über Skalierungsrichtlinien verfügt, verwendet sie beim Skalieren der Gruppe dieselbe Standard-Aufwärmphase, um zu verhindern, dass Instances auf die CloudWatch Metriken angerechnet werden, bevor die Initialisierung abgeschlossen ist.

Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

Frist der Zustandsprüfung

Amazon EC2 Auto Scaling bestimmt anhand des Status der von Ihrer Auto-Scaling-Gruppe verwendeten Zustandsprüfungen, ob eine Instance fehlerfrei ist. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Um sicherzustellen, dass diese Zustandsprüfungen so schnell wie möglich beginnen, sollten Sie die Karenzzeit für die Zustandsprüfung der Gruppe nicht zu hoch ansetzen, nur hoch genug, damit Ihre Elastic Load Balancing-Zustandsprüfungen feststellen können, ob ein Ziel zur Bearbeitung von Anfragen verfügbar ist. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.

Skalieren Ihrer Auto-Scaling-Gruppe

Eine Instance-Wartungsrichtlinie gilt nur für Instance-Wartungsereignisse und verhindert nicht die manuelle oder automatische Skalierung der Gruppe.

Wenn Ihrer Auto-Scaling-Gruppe Skalierungsrichtlinien oder geplante Aktionen zugeordnet sind, können diese parallel ausgeführt werden, während Wartungsereignisse für Instances stattfinden. In einem solchen Fall könnten sie die gewünschte Kapazität der Gruppe erhöhen oder verringern, jedoch nur innerhalb der von Ihnen definierten Skalierungslimits. Weitere Informationen zu den Limits finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.

Beispielszenarien

In einem typischen Szenario könnten Ihre Instance-Wartungsrichtlinie und die gewünschte Kapazität ungefähr so aussehen:

- Minimaler fehlerfreier Prozentsatz = 90 Prozent
- Maximaler fehlerfreier Prozentsatz = 120 Prozent
- Gewünschte Kapazität = 100

Während eines Instance-Wartungsereignisses kann Ihre Auto-Scaling-Gruppe über 90 bis 120 Instances verfügen. Nach dem Ereignis verfügt die Gruppe wieder über 100 Instances.

Wenn Sie eine Instance-Wartungsrichtlinie mit einer Auto-Scaling-Gruppe verwenden, die über einen warmen Pool verfügt, werden die minimalen und maximalen fehlerfreien Prozentsätze getrennt auf die Auto-Scaling-Gruppe und den warmen Pool angewendet.

Nehmen wir die folgende Konfiguration als Beispiel:

- Minimaler fehlerfreier Prozentsatz = 90 Prozent
- Maximaler fehlerfreier Prozentsatz = 120 Prozent
- Gewünschte Kapazität = 100
- Größe des warmen Pools = 10

Wenn Sie eine Instance-Aktualisierung starten, um die Instances der Gruppe zu recyceln, ersetzt Amazon EC2 Auto Scaling zuerst die Instances in der Auto-Scaling-Gruppe und dann die Instances im warmen Pool. Amazon EC2 Auto Scaling arbeitet zwar immer noch daran, Instances in der Auto-Scaling-Gruppe zu ersetzen, aber die Gruppe könnte zwischen 90 und 120 Instances haben. Nach Fertigstellung der Gruppe kann Amazon EC2 Auto Scaling daran arbeiten, Instances im warmen Pool zu ersetzen. Währenddessen kann der warme Pool zwischen 9 und 12 Instances haben.

Festlegen einer Instance-Wartungsrichtlinie für Ihre Auto-Scaling-Gruppe

Sie können eine Instance-Wartungsrichtlinie erstellen, wenn Sie eine Auto-Scaling-Gruppe erstellen. Sie können sie auch für vorhandene Gruppen erstellen.

Durch Festlegen einer Instance-Wartungsrichtlinie für Ihre Auto-Scaling-Gruppe müssen Sie für die Instance-Aktualisierung keine Werte mehr angeben, es sei denn, Sie möchten die Instance-Wartungsrichtlinie überschreiben.

In der Konsole bietet Amazon EC2 Auto Scaling Optionen, die Ihnen die ersten Schritte erleichtern.

Inhalt

Festlegen einer Instance-Wartungsrichtlinie

Verwenden Sie eine der folgenden Methoden, um eine Instance-Wartungsrichtlinie für eine Auto-Scaling-Gruppe festzulegen:

Console

Festlegen einer Instance-Wartungsrichtlinie für eine Gruppe (Konsole)

1. Befolgen Sie die Anweisungen in Erstellen einer Auto-Scaling-Gruppe mithilfe einer Startvorlage und führen Sie jeden Schritt des Verfahrens bis zu Schritt 11 durch.
2. Geben Sie unter Konfigurieren von Gruppengröße und Skalierungsrichtlinien für Gewünschte Kapazität die anfängliche Anzahl von Instances ein, die gestartet werden sollen.
3. Wenn im Abschnitt Skalierung unter Skalierungslimits Ihr neuer Wert für die gewünschte Kapazität größer als die gewünschte Mindestkapazität und die gewünschte Höchstkapazität ist, wird die gewünschte Höchstkapazität automatisch auf den neuen Wert für die gewünschte Kapazität erhöht. Sie können die Limits bei Bedarf ändern.

4. Wählen Sie für Automatische Skalierung aus, ob Sie eine Skalierungsrichtlinie für die Zielverfolgung erstellen möchten. Sie können diese Richtlinie auch erstellen, nachdem Sie Ihre Auto-Scaling-Gruppe erstellt haben.

   Wenn Sie sich für die Skalierungsrichtlinie für die Zielverfolgung entscheiden, befolgen Sie die Anweisungen unter Erstellen einer Zielverfolgungs-Skalierungsrichtlinie, um die Richtlinie zu erstellen.

5. Wählen Sie im Abschnitt Instance-Wartungsrichtlinie eine der verfügbaren Optionen aus:

   - Vor dem Beenden starten: Eine neue Instance muss zuerst bereitgestellt werden, bevor eine bestehende Instance beendet werden kann. Dies ist eine gute Wahl für Anwendungen, bei denen Verfügbarkeit wichtiger ist als Kosteneinsparungen.
   - Beenden und starten: Neue Instances werden zur gleichen Zeit bereitgestellt, wie Ihre bestehenden Instances beendet werden. Dies ist eine gute Wahl für Anwendungen, bei denen Kosteneinsparungen Vorrang vor der Verfügbarkeit haben. Es ist auch eine gute Wahl für Anwendungen, die nicht mehr Kapazität benötigen, als derzeit verfügbar ist.
   - Benutzerdefinierte Richtlinie: Mit dieser Option können Sie für Ihre Richtlinie einen benutzerdefinierten Mindest- und Höchstbereich für die Kapazität einrichten, die beim Austausch von Instances verfügbar sein soll. Dies kann Ihnen helfen, das richtige Gleichgewicht zwischen Kosten und Verfügbarkeit zu finden.

6. Geben Sie unter Fehlerfreien Prozentsatz festlegen Werte für eines oder beide der folgenden Felder ein. Die aktivierten Felder variieren je nach der Option, die Sie im vorherigen Schritt ausgewählt haben.

   - Min.: Legt den fehlerfreien Mindestprozentsatz fest, der erforderlich ist, um mit dem Ersetzen von Instances fortzufahren.
   - Max.: Legt den maximalen fehlerfreien Prozentsatz fest, der während des Ersetzens von Instances möglich ist.

7. Erweitern Sie den Abschnitt Kapazität bei Ersatz auf Grundlage Ihrer gewünschten Kapazität anzeigen, um zu überprüfen, ob die Werte für Min. und Max für Ihre Gruppe gelten. Welche genauen Werte verwendet werden, hängt vom gewünschten Kapazitätswert ab, der sich ändert, wenn die Gruppe skaliert wird.

8. Fahren Sie mit den Schritten unter Erstellen einer Auto-Scaling-Gruppe mithilfe einer Startvorlage fort.

AWS CLI

Festlegen einer Instance-Wartungsrichtlinie für eine Gruppe (AWS CLI)

Fügen Sie dem Befehl create-auto-scaling-group die Option `--instance-maintenance-policy` hinzu. Im folgenden Beispiel wird eine Instance-Wartungsrichtlinie für eine neue Auto-Scaling-Gruppe mit dem Namen *my-asg* festgelegt.

```
aws autoscaling create-auto-scaling-group \
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \
  --auto-scaling-group-name my-asg \
  --min-size 1 \
  --max-size 10 \
  --desired-capacity 5 \
  --default-instance-warmup 20 \
  --instance-maintenance-policy '{
      "MinHealthyPercentage": 90,
      "MaxHealthyPercentage": 120
    }' \
  --vpc-zone-identifier "subnet-5e6example,subnet-613example,subnet-c93example"
```

Console

Festlegen einer Instance-Wartungsrichtlinie für eine vorhandene Gruppe (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie in der Navigationsleiste oben die AWS-Region aus, in der Sie Ihre Auto-Scaling-Gruppe erstellt haben.
3. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
4. Wählen Sie auf der Registerkarte Details die Option Instance-Wartungsrichtlinie, Bearbeiten aus.
5. Wählen Sie zum Festlegen einer Instance-Wartungsrichtlinie für die Gruppe eine der verfügbaren Optionen aus:

- Vor dem Beenden starten: Eine neue Instance muss zuerst bereitgestellt werden, bevor eine bestehende Instance beendet werden kann. Dies ist eine gute Wahl für Anwendungen, bei denen Verfügbarkeit wichtiger ist als Kosteneinsparungen.
- Beenden und starten: Neue Instances werden zur gleichen Zeit bereitgestellt, wie Ihre bestehenden Instances beendet werden. Dies ist eine gute Wahl für Anwendungen, bei denen Kosteneinsparungen Vorrang vor der Verfügbarkeit haben. Es ist auch eine gute Wahl für Anwendungen, die nicht mehr Kapazität benötigen, als derzeit verfügbar ist.
- Benutzerdefinierte Richtlinie: Mit dieser Option können Sie für Ihre Richtlinie einen benutzerdefinierten Mindest- und Höchstbereich für die Kapazität einrichten, die beim Austausch von Instances verfügbar sein soll. Dies kann Ihnen helfen, das richtige Gleichgewicht zwischen Kosten und Verfügbarkeit zu finden.

6. Geben Sie unter Fehlerfreien Prozentsatz festlegen Werte für eines oder beide der folgenden Felder ein. Die aktivierten Felder variieren je nach der Option, die Sie im vorherigen Schritt ausgewählt haben.

   - Min.: Legt den fehlerfreien Mindestprozentsatz fest, der erforderlich ist, um mit dem Ersetzen von Instances fortzufahren.
   - Max.: Legt den maximalen fehlerfreien Prozentsatz fest, der während des Ersetzens von Instances möglich ist.

7. Erweitern Sie den Abschnitt Kapazität bei Ersatz auf Grundlage Ihrer gewünschten Kapazität anzeigen, um zu überprüfen, ob die Werte für Min. und Max für Ihre Gruppe gelten. Welche genauen Werte verwendet werden, hängt vom gewünschten Kapazitätswert ab, der sich ändert, wenn die Gruppe skaliert wird.
8. Wählen Sie Aktualisieren.

AWS CLI

Festlegen einer Instance-Wartungsrichtlinie für eine vorhandene Gruppe (AWS CLI)

Fügen Sie die Option `--instance-maintenance-policy` dem Befehl update-auto-scaling-group hinzu. Im folgenden Beispiel wird eine Instance-Wartungsrichtlinie für eine spezifizierte Auto-Scaling-Gruppe festgelegt.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --instance-maintenance-policy '{
      "MinHealthyPercentage": 90,
```

```
    "MaxHealthyPercentage": 120
  }'
```

Entfernen einer Instance-Wartungsrichtlinie

Wenn Sie die Verwendung einer Instance-Wartungsrichtlinie mit Ihrer Auto-Scaling-Gruppe beenden möchten, können Sie sie entfernen.

Console

Entfernen einer Instance-Wartungsrichtlinie (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie in der Navigationsleiste oben die AWS-Region aus, in der Sie Ihre Auto-Scaling-Gruppe erstellt haben.
3. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
4. Wählen Sie auf der Registerkarte Details die Option Instance-Wartungsrichtlinie, Bearbeiten aus.
5. Wählen Sie Keine Instance-Wartungsrichtlinie aus.
6. Wählen Sie Aktualisieren.

AWS CLI

Entfernen einer Instance-Wartungsrichtlinie (AWS CLI)

Fügen Sie die Option `--instance-maintenance-policy` dem Befehl update-auto-scaling-group hinzu. Im folgenden Beispiel wird eine Instance-Wartungsrichtlinie einer spezifizierten Auto-Scaling-Gruppe entfernt.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --instance-maintenance-policy '{
      "MinHealthyPercentage": -1,
      "MaxHealthyPercentage": -1
    }'
```

Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling bietet die Möglichkeit, Lebenszyklus-Hooks zu Ihren Auto-Scaling-Gruppen hinzuzufügen. Mit diesen Hooks können Sie Lösungen erstellen, die Ereignisse im Lebenszyklus von Auto-Scaling-Instances erkennen und dann eine benutzerdefinierte Aktion auf Instances ausführen, wenn das entsprechende Lebenszyklusereignis eintritt. Ein Lebenszyklus-Hook gibt eine bestimmte Zeitspanne vor (standardmäßig eine Stunde), in der auf den Abschluss der Aktion gewartet wird, bevor die Instance in den nächsten Zustand übergeht.

Als Beispiel für die Verwendung von Lebenszyklus-Hooks mit Auto-Scaling-Instances:

- Wenn ein horizontales Skalierungsereignis auftritt, schließt die neu gestartete Instance ihre Startsequenz ab und wechselt in einen Wartezustand. Während sich die Instance in einem Wartestatus befindet, können Sie auf ein Skript ausführen, um die erforderlichen Softwarepakete für Ihre Anwendung herunterzuladen und zu installieren. Stellen Sie dabei sicher, dass Ihre Instance vollständig bereit ist, bevor sie mit dem Empfang von Datenverkehr beginnt. Wenn das Skript die Installation der Software abgeschlossen hat, sendet es den complete-lifecycle-action-Befehl, um fortzufahren.
- Wenn ein Scale-In-Ereignis eintritt, pausiert ein Lifecycle-Hook die Instance, bevor sie beendet wird, und sendet Ihnen eine Benachrichtigung über Amazon. EventBridge Während sich die Instance im Wartestatus befindet, können Sie eine AWS Lambda Funktion aufrufen oder eine Verbindung zur Instance herstellen, um Logs oder andere Daten herunterzuladen, bevor die Instance vollständig beendet wird.

Eine beliebte Verwendung von Lebenszyklus-Hooks besteht darin, zu steuern, wann Instances bei Elastic Load Balancing registriert werden. Wenn Sie Ihrer Auto-Scaling-Gruppe einen Start-Lebenszyklus-Hook hinzufügen, können Sie sicherstellen, dass Ihre Bootstrap-Skripte erfolgreich abgeschlossen wurden und die Anwendungen auf den Instances bereit sind, Datenverkehr anzunehmen, bevor sie am Ende des Lebenszyklus-Hooks beim Load Balancer registriert werden.

Inhalt

- Abrufen des Ziellebenszyklus-Status durch Instance-Metadaten
- Lebenszyklus-Hooks hinzufügen
- Eine Lebenszyklus-Aktion abschließen
- Tutorial: Konfigurieren Sie Benutzerdaten zum Abrufen des Ziellebenszyklusstatus über Instance-Metadaten
- Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft

Verfügbarkeit von Lebenszyklus-Hooks

In der folgenden Tabelle finden Sie die Lebenszyklus-Hooks, die für verschiedene Szenarien verfügbar sind

Ereignis	Instance-Start oder -Beendigung[1]	Maximale Instance-Lebensdauer: Ersatz-Instances	Instance-Aktualisierung: Ersatz-Instances	Kapazitätsausgleich: Ersatz-Instances	Warm-Pools: Instances, die den Warm-Pool betreten und verlassen
Startende Instance	✓	✓	✓	✓	✓
Endende Instance	✓	✓	✓	✓	✓

[1] Gilt für alle Starts und Beendigungen, unabhängig davon, ob sie automatisch oder manuell eingeleitet werden, z. B. wenn Sie die Optionen `SetDesiredCapacity` oder `TerminateInstanceInAutoScalingGroup` aufrufen. Gilt nicht, wenn Sie Instances zuordnen oder trennen, Instances in den Standby-Modus verschieben oder die Gruppe mit der Option „Löschen erzwingen“ löschen.

Überlegungen zu und Einschränkungen für Lebenszyklus-Hooks

Bei der Arbeit mit Lebenszyklus-Hooks sind die folgenden Hinweise und Einschränkungen zu beachten:

- Amazon EC2 Auto Scaling bietet einen eigenen Lebenszyklus, der die Verwaltung von Auto-Scaling-Gruppen unterstützt. Dieser Lebenszyklus unterscheidet sich von dem anderer EC2-Instances. Weitere Informationen finden Sie unter Instance-Lebenszyklus bei Amazon EC2 Auto Scaling. Instances in einem Warm Pool haben auch einen eigenen Lebenszyklus, wie unter Lebenszyklusstatusübergänge für Instances in einem Warm Pool beschrieben.
- Sie können Lebenszyklus-Hooks mit Spot-Instances verwenden, aber ein Lebenszyklus-Hook kann nicht verhindern, dass eine Instance beendet wird, wenn keine Kapazität mehr verfügbar ist, was jederzeit innerhalb eines zweiminütigen Unterbrechungshinweises passieren kann. Weitere Informationen finden Sie unter Spot-Instance-Unterbrechungen im Amazon EC2 EC2-Benutzerhandbuch. Sie können jedoch den Kapazitätsausgleich aktivieren, um Spot-Instances proaktiv zu ersetzen, die eine Neuausgleichsempfehlung vom Amazon-EC2-Spot-Service erhalten haben, ein Signal, das gesendet wird, wenn eine Spot-Instance einem erhöhten Unterbrechungsrisiko ausgesetzt ist. Weitere Informationen finden Sie unter Verwenden des Kapazitätsausgleichs, um Amazon-EC2-Spot-Unterbrechungen zu behandeln.
- Instances können eine begrenzte Zeit lang in einem Wartestatus verbleiben. Das Standard-Timeout für einen Lebenszyklus-Hook beträgt eine Stunde (Heartbeat-Timeout). Es gibt auch ein globales Timeout, das die maximale Zeitspanne angibt, für die Sie eine Instance in einem Wartestatus belassen können. Das globale Timeout beträgt 48 Stunden bzw. das 100-fache der Heartbeat-Zeitüberschreitung, je nachdem, welcher Wert kleiner ist.
- Das Ergebnis des Lebenszyklus-Hooks kann entweder Abbrechen oder Fortsetzen sein. Wenn eine Instance gestartet wird, zeigt „Continue“ (Fortfahren) an, dass Ihre Aktionen erfolgreich waren und dass Amazon EC2 Auto Scaling die Instance in Betrieb nehmen kann. Andernfalls zeigt „Abandon“ (Abbruch) an, dass die benutzerdefinierten Aktionen fehlgeschlagen sind, und wir die Instance beenden und diese ersetzen können. Wenn die Instance beendet wird, erlauben sowohl „Abbrechen“ als auch „Fortfahren“ das Beenden der Instance. Allerdings stoppt „Abbrechen“ alle verbleibenden Aktionen, z. B. andere Lebenszyklus-Hooks, und „Fortfahren“ ermöglicht das Abschließen anderer Lebenszyklus-Hooks.
- Amazon EC2 Auto Scaling begrenzt die Rate, mit der Instances gestartet werden können, wenn die Lebenszyklus-Hooks konsistent fehlschlagen. Testen und Beheben Sie daher alle dauerhaften Fehler in Ihren Lebenszyklus-Aktionen.
- Das Erstellen und Aktualisieren von Lifecycle-Hooks mit dem AWS CLI AWS CloudFormation, oder einem SDK bietet Optionen, die beim Erstellen eines Lifecycle-Hooks aus dem nicht verfügbar sind. AWS Management Console Beispielsweise wird das Feld zur Angabe des ARN eines SNS-Themas oder einer SQS-Warteschlange nicht in der Konsole angezeigt, da Amazon EC2 Auto

Scaling bereits Ereignisse an Amazon sendet. EventBridge Diese Ereignisse können gefiltert und nach Bedarf an AWS Dienste wie Lambda, Amazon SNS und Amazon SQS umgeleitet werden.

- Sie können einer Auto Scaling Scaling-Gruppe während der Erstellung mehrere Lifecycle-Hooks hinzufügen, indem Sie die CreateAutoScalingGroupAPI mit dem AWS CLI AWS CloudFormation, oder einem SDK aufrufen. Jeder Hook muss jedoch das gleiche Benachrichtigungsziel und die gleiche IAM-Rolle haben, falls angegeben. Um Lifecycle-Hooks mit unterschiedlichen Benachrichtigungszielen und unterschiedlichen Rollen zu erstellen, erstellen Sie die Lifecycle-Hooks nacheinander in separaten Aufrufen der PutLifecycleHook-API.
- Wenn Sie zum Beispiel einen Lebenszyklus-Hook für den Instance-Start hinzufügen, beginnt die Übergangsfrist für die Zustandsprüfung, sobald die Instance den Status `InService` erreicht hat. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.

Überlegungen zur Skalierung

- Dynamische Skalierungsrichtlinien werden als Reaktion auf CloudWatch metrische Daten wie CPU- und Netzwerk-I/O, die über mehrere Instanzen aggregiert werden, nach innen und außen skaliert. Beim Skalieren zählt Amazon EC2 Auto Scaling eine neue Instance nicht sofort zu den aggregierten Instance-Metriken der Auto-Scaling-Gruppe. Es wartet, bis die Instance den Status `InService` erreicht hat und der Instance-Warmup abgeschlossen ist. Weitere Informationen finden Sie unter Leistungsaspekte der Skalierung im Thema Aufwärmen der Standard-Instance.
- Beim Skalieren spiegeln die aggregierten Instance-Metriken möglicherweise nicht sofort die Entfernung einer beendeten Instance wider. Die terminierende Instance wird nicht mehr zu den aggregierten Instance-Metriken der Gruppe gezählt, kurz nachdem der Amazon EC2 Auto Scaling Terminierungsworkflow beginnt.
- In den meisten Fällen, in denen Lebenszyklus-Hooks aufgerufen werden, werden die Skalierungsaktivitäten aufgrund einfacher Skalierungsrichtlinien angehalten, bis die Lifecycle-Aktionen abgeschlossen sind und die Ruhephase abgelaufen ist. Bei einem Festlegen eines langen Intervalls für die Ruhephase dauert es länger, bis die Skalierung fortgesetzt werden kann. Weitere Informationen finden Sie unter Lebenszyklus-Hooks können zusätzliche Verzögerungen verursachen im Thema Ruhephase. Im Allgemeinen empfehlen wir, keine einfachen Skalierungsrichtlinien zu verwenden, wenn Sie stattdessen entweder eine Stufenskalierung oder eine Zielverfolgungsskalierungs-Richtlinie verwenden können.

Zugehörige Ressourcen

Ein Einführungsvideo finden Sie unter AWS re:Invent 2018: Capacity Management Made Easy with Amazon EC2 Auto Scaling on. YouTube

Wir stellen einige JSON- und YAML-Vorlagenausschnitte zur Verfügung, anhand derer Sie verstehen können, wie Sie Lifecycle-Hooks in Ihren Stack-Vorlagen deklarieren. AWS CloudFormation Weitere Informationen finden Sie in der AWS::AutoScaling::LifecycleHookReferenz im AWS CloudFormation Benutzerhandbuch.

Sie können auch unser GitHubRepository besuchen, um Beispielvorlagen und Benutzerdatenskripte für Lifecycle-Hooks herunterzuladen.

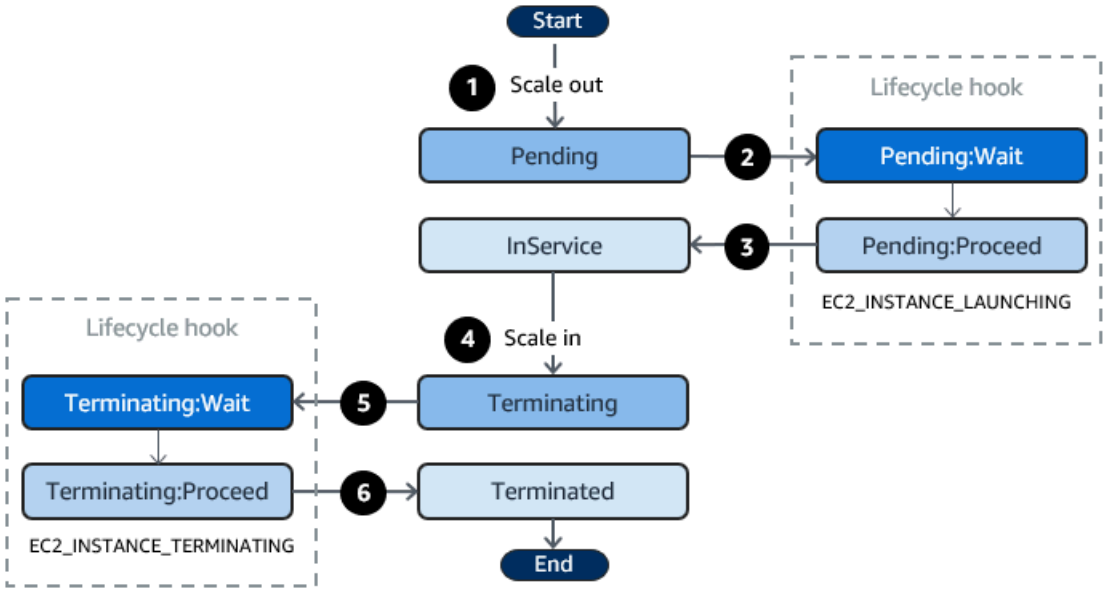
Beispiele für die Verwendung von Lebenszyklus-Hooks finden Sie in den folgenden Blog-Beiträgen.

- Aufbau eines Backup-Systems für skalierte Instances mithilfe von Lambda und Amazon EC2 Run Command
- Code vor dem Beenden einer EC2 Auto Scaling-Instance ausführen.

So funktionieren Lebenszyklus-Hooks

Eine Amazon-EC2-Instance wechselt von dem Zeitpunkt, an dem sie gestartet wird, bis zu ihrer Beendigung durch verschiedene Status. Sie können benutzerdefinierte Aktionen für Ihre Auto-Scaling-Gruppe erstellen, die ausgeführt werden, wenn eine Instance aufgrund eines Lebenszyklus-Hooks in einen Wartezustand übergeht.

Die folgende Abbildung zeigt die Übergänge zwischen Auto Scaling Scaling-Instanzzuständen, wenn Sie Lifecycle-Hooks für Scale-Out und Scale-In verwenden.



Wie im obigen Diagramm gezeigt:

1. Die Auto-Scaling-Gruppe reagiert auf ein horizontales Skalierungsereignis und beginnt mit dem Starten einer Instance.
2. Der Lebenszyklus-Hook versetzt die Instance in einen Wartestatus (`Pending:Wait`) und führt dann eine benutzerdefinierte Aktion aus.

   Die Instance bleibt in einem Wartezustand, bis Sie entweder die Lebenszyklusaktion beenden oder der Timeout-Zeitraum endet. Standardmäßig verbleibt die Instance eine Stunde lang im Wartestatus, dann führt die Auto-Scaling-Gruppe den Start der Instance fort (`Pending:Proceed`). Wird mehr Zeit benötigt, können Sie die Zeit bis zur Zeitüberschreitung zurücksetzen, indem Sie eine Heartbeat-Benachrichtigung aufzeichnen. Wenn Sie die Lebenszyklusaktion bei Vollendung der benutzerdefinierten Aktion und bevor der Zeitüberschreitungszeitraums abgelaufen ist, abschließen, endet der Zeitraum und die Auto-Scaling-Gruppe setzt den Startvorgang fort.
3. Die Instance tritt in den `InService`-Zustand und die Kulanzfrist der Zustandsprüfung beginnt. Bevor die Instance jedoch den `InService`-Status erreicht, wird sie, wenn die Auto-Scaling-Gruppe einem Elastic Load Balancing zugeordnet ist, beim Load Balancer registriert, und der Load Balancer beginnt mit der Überprüfung seines Zustands. Nach Ablauf der Kulanzfrist für die Zustandsprüfung beginnt Amazon EC2 Auto Scaling, den Zustand der Instance zu prüfen.
4. Die Auto-Scaling-Gruppe reagiert auf ein horizontales Skalierungsereignis und beginnt mit dem Beenden einer Instance. Wenn die Auto-Scaling-Gruppe mit Elastic Load Balancing verwendet wird, wird die beendende Instance zunächst beim Load Balancer registriert. Wenn Connection Draining für den Load Balancer aktiviert ist, akzeptiert die Instance keine neuen Verbindungen

und wartet darauf, dass vorhandene Verbindungen abgebaut werden, bevor die Registrierung abgeschlossen wird.

5. Der Lebenszyklus-Hook versetzt die Instance in einen Wartestatus (`Terminating:Wait`) und führt dann eine benutzerdefinierte Aktion aus.

   Die Instance bleibt in einem Wartezustand, bis Sie entweder die Lebenszyklusaktion beenden oder der Timeout-Zeitraum endet (standardmäßig eine Stunde). Nachdem Sie den Lebenszyklus-Hook abgeschlossen haben oder der Timeout-Zeitraum abläuft, wechselt die Instance in den nächsten Status (`Terminating:Proceed`).

6. Die Instance wurde beendet.

> **Important**
>
> Instances in einem Warm Pool haben auch einen eigenen Lebenszyklus mit entsprechenden Wartestatus, wie unter Lebenszyklusstatusübergänge für Instances in einem Warm Pool beschrieben.

Vorbereiten des Hinzufügens eines Lebenszyklus-Hook zu einer Auto-Scaling-Gruppe

Stellen Sie sicher, dass Ihr Benutzerdatenskript oder Ihr Benachrichtigungsziel korrekt eingerichtet ist, bevor Sie Ihrer Auto-Scaling-Gruppe einen Lebenszyklus-Hook hinzufügen.

- Um ein Benutzerdatenskript zu nutzen, um benutzerdefinierte Aktionen für Ihre Instances während des Starts auszuführen, müssen Sie kein Benachrichtigungsziel konfigurieren. Sie müssen jedoch bereits die Startvorlage oder die Startkonfiguration erstellt haben, die Ihr Benutzerdatenskript angibt und es Ihrer Auto-Scaling-Gruppe zuordnet. Weitere Informationen zu Benutzerdatenskripten finden Sie unter Befehle auf Ihrer Linux-Instance beim Start ausführen im Amazon EC2 EC2-Benutzerhandbuch.
- Um Amazon EC2 Auto Scaling zu signalisieren, wenn die Lifecycle-Aktion abgeschlossen ist, müssen Sie dem Skript den CompleteLifecycleAction-API-Aufruf hinzufügen und manuell eine IAM-Rolle mit einer Richtlinie erstellen, die es Auto Scaling Scaling-Instances ermöglicht, diese API aufzurufen. Ihre Startvorlage oder Startkonfiguration muss diese Rolle mithilfe eines IAM-Instance-Profils angeben, das beim Start an Ihre Amazon-EC2-Instances angehängt wird.

Weitere Informationen finden Sie unter Eine Lebenszyklus-Aktion abschließen und IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden.

- Um einen Dienst wie Lambda zum Ausführen einer benutzerdefinierten Aktion zu verwenden, müssen Sie bereits eine EventBridge Regel erstellt und eine Lambda-Funktion als Ziel angegeben haben. Weitere Informationen finden Sie unter Konfigurieren eines Benachrichtigungsziels für Lebenszyklus-Benachrichtigungen.
- Damit Lambda Amazon EC2 Auto Scaling signalisieren kann, wenn die Lebenszyklusaktion abgeschlossen ist, müssen Sie den CompleteLifecycleAction-API-Aufruf zum Funktionscode hinzufügen. Sie müssen auch eine IAM-Richtlinie an die Ausführungsrolle der Funktion angehängt haben, um Lambda die Berechtigung zum Vervollständigen von Lebenszyklus-Aktionen zu erteilen. Weitere Informationen finden Sie unter Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft.
- Um einen Service wie Amazon SNS oder Amazon SQS zum Ausführen einer benutzerdefinierten Aktion verwenden zu können, müssen Sie bereits das SNS-Thema oder die SQS-Warteschlange erstellt haben und über den Amazon-Ressourcennamen (ARN) verfügen. Sie müssen auch bereits die IAM-Rolle erstellt haben, die Amazon EC2 Auto Scaling Zugriff auf Ihr SNS-Thema oder SQS-Ziel ermöglicht, und über deren ARN verfügen. Weitere Informationen finden Sie unter Konfigurieren eines Benachrichtigungsziels für Lebenszyklus-Benachrichtigungen.

> **Note**
>
> Wenn Sie einen Lifecycle-Hook in der Konsole hinzufügen, sendet Amazon EC2 Auto Scaling standardmäßig Lebenszyklusereignisbenachrichtigungen an Amazon EventBridge. Die Verwendung EventBridge eines Benutzerdatenskripts ist eine empfohlene bewährte Methode. Um einen Lifecycle-Hook zu erstellen, der Benachrichtigungen direkt an Amazon SNS oder Amazon SQS sendet, verwenden Sie das,, oder ein SDK AWS CLI AWS CloudFormation, um den Lifecycle-Hook hinzuzufügen.

Konfigurieren eines Benachrichtigungsziels für Lebenszyklus-Benachrichtigungen

Sie können einer Auto-Scaling-Gruppe Lebenszyklus-Hooks hinzufügen, um benutzerdefinierte Aktionen auszuführen, wenn eine Instance in einen Wartestatus wechselt. Sie können einen Zielservice auswählen, der diese Aktionen abhängig von Ihrem bevorzugten Entwicklungsansatz ausführt.

Der erste Ansatz verwendet Amazon EventBridge , um eine Lambda-Funktion aufzurufen, die die gewünschte Aktion ausführt. Der zweite Ansatz umfasst das Erstellen eines Amazon Simple Notification Service (Amazon SNS)-Themas, für das Benachrichtigungen veröffentlicht werden. Kunden können das SNS-Thema abonnieren und veröffentlichte Nachrichten über ein unterstütztes Protokoll empfangen. Der letzte Ansatz umfasst die Verwendung von Amazon Simple Queue Service (Amazon SQS), einem Messaging-System, das von verteilten Anwendungen verwendet wird, um Nachrichten über ein Abfragemodell auszutauschen.

Als bewährte Methode empfehlen wir die Verwendung von. EventBridge Die an Amazon SNS und Amazon SQS gesendeten Benachrichtigungen enthalten dieselben Informationen wie die Benachrichtigungen, an die Amazon EC2 Auto Scaling sendet. EventBridge Bisher bestand die Standardpraxis darin EventBridge, eine Benachrichtigung an SNS oder SQS zu senden und einen anderen Service in SNS oder SQS zu integrieren, um programmatische Aktionen durchzuführen. Heute stehen EventBridge Ihnen mehr Optionen zur Verfügung, auf welche Dienste Sie abzielen können, und erleichtert die Verarbeitung von Ereignissen mithilfe einer serverlosen Architektur.

In den folgenden Verfahren wird beschrieben, wie Sie Ihr Benachrichtigungsziel einrichten.

Denken Sie daran: Wenn Sie über ein Benutzerdatenskript in Ihrer Startvorlage- oder Startkonfiguration verfügen, das Ihre Instances beim Starten konfiguriert, müssen Sie keine Benachrichtigungen erhalten, um benutzerdefinierte Aktionen für Ihre Instances auszuführen.

Inhalt

> **Important**
>
> Die EventBridge Regel, die Lambda-Funktion, das Amazon SNS SNS-Thema und die Amazon SQS SQS-Warteschlange, die Sie mit Lifecycle-Hooks verwenden, müssen sich immer in derselben Region befinden, in der Sie Ihre Auto Scaling Scaling-Gruppe erstellt haben.

Benachrichtigungen an Lambda weiterleiten mit EventBridge

Sie können eine EventBridge Regel so konfigurieren, dass sie eine Lambda-Funktion aufruft, wenn eine Instanz in den Wartezustand wechselt. Amazon EC2 Auto Scaling sendet eine Benachrichtigung EventBridge über ein Lifecycle-Ereignis an die Instance, die gestartet oder beendet wird, sowie ein Token, mit dem Sie die Lifecycle-Aktion steuern können. Beispiele für diese Ereignisse finden Sie unter Ereignis-Referenz für Amazon EC2 Auto Scaling.

> **Note**
>
> Wenn Sie die verwenden, AWS Management Console um eine Ereignisregel zu erstellen, fügt die Konsole automatisch die IAM-Berechtigungen hinzu, die erforderlich sind, um die EventBridge Berechtigung zum Aufrufen Ihrer Lambda-Funktion zu erteilen. Wenn Sie eine Ereignisregel mit AWS CLI erstellen, müssen Sie diese Berechtigung ausdrücklich erteilen. Informationen zum Erstellen von Ereignisregeln in der EventBridge Konsole finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erstellen von EventBridge Amazon-Regeln, die auf Ereignisse reagieren.
> – oder –
> Ein einführendes Tutorial, das sich an Konsolenbenutzer richtet, finden Sie unter Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft. Dieses Tutorial zeigt Ihnen, wie Sie eine einfache Lambda-Funktion erstellen, die auf Startereignisse wartet und diese in ein CloudWatch Logs-Protokoll schreibt.

Um eine EventBridge Regel zu erstellen, die eine Lambda-Funktion aufruft

1. Erstellen Sie mithilfe der Lambda-Konsole eine Lambda-Funktion und notieren Sie ihren Amazon-Ressourcennamen (ARN). Zum Beispiel `arn:aws:lambda:region:123456789012:function:my-function`. Sie benötigen den ARN, um ein EventBridge Ziel zu erstellen. Weitere Informationen finden Sie unter Erste Schritte mit Lambda im AWS Lambda -Entwicklerhandbuch.
2. Um eine Regel zu erstellen, die auf Ereignisse für den Start der Instance passt, verwenden Sie den folgenden put-rule-Befehl.

```
aws events put-rule --name my-rule --event-pattern file://pattern.json --state
 ENABLED
```

Im folgenden Beispiel wird die Aktion `pattern.json` für eine Instance zum Starten des Lebenszyklus veranschaulicht. Ersetzen Sie den Text in *Kursivschrift* mit dem Namen Ihrer Auto-Scaling-Gruppe.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "AutoScalingGroupName": [ "my-asg" ]
   }
}
```

Wenn der Befehl erfolgreich ausgeführt wird, EventBridge antwortet er mit dem ARN der Regel. Notieren Sie diesen ARN. Sie müssen ihn in Schritt 4 eingeben.

Um eine Regel zu erstellen, die mit anderen Ereignissen übereinstimmt, ändern Sie das Ereignismuster. Weitere Informationen finden Sie unter Wird EventBridge zur Behandlung von Auto Scaling Scaling-Ereignissen verwendet.

3. Verwenden Sie Folgendes, um die Lambda-Funktion anzugeben, die als Ziel für die Regel verwendet werden soll:put-targets-Befehl.

```
aws events put-targets --rule my-rule --targets
 Id=1,Arn=arn:aws:lambda:region:123456789012:function:my-function
```

Im vorangehenden Befehl, ist *my-rule* der Name, den Sie in Schritt 2 für die Regel angegeben haben, und der Wert für den Parameter `Arn` ist die ARN der Funktion, die Sie in Schritt 1 erstellt haben.

4. Um Berechtigungen hinzuzufügen, die es der Regel erlauben, Ihre Lambda-Funktion aufzurufen, verwenden Sie den folgenden Lambda add-permission-Befehl. Dieser Befehl vertraut dem EventBridge Dienstprinzipal (`events.amazonaws.com`) und beschränkt die Berechtigungen auf die angegebene Regel.

```
aws lambda add-permission --function-name my-function --statement-id my-unique-id \
  --action 'lambda:InvokeFunction' --principal events.amazonaws.com --source-arn
 arn:aws:events:region:123456789012:rule/my-rule
```

Beim vorhergehenden Befehl:

- *my-function* ist der Name der Lambda-Funktion, die von der Regel als Ziel verwendet werden soll.
- *my-unique-id* ist ein eindeutiger Identifier, den Sie definieren, um die Anweisung in der Lambda-Funktionsrichtlinie zu beschreiben.
- `source-arn`ist der ARN der EventBridge Regel.

Wird der Befehl erfolgreich ausgeführt, erhalten Sie eine Ausgabe ähnlich der folgenden:

```
{
  "Statement": "{\"Sid\":\"my-unique-id\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"events.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-west-2:123456789012:function:my-function\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:events:us-west-2:123456789012:rule/my-rule\"}}}"
}
```

Der `Statement`-Wert ist eine JSON-Zeichenfolgenversion der Anweisung, die der Lambda-Funktionsrichtlinie hinzugefügt wurde.

5. Nachdem Sie diese Anweisungen befolgt haben, fahren Sie mit Lebenszyklus-Hooks hinzufügen fort.

Benachrichtigungen über Amazon SNS erhalten

Sie können Amazon SNS dazu verwenden, ein Benachrichtigungsziel (ein SNS-Thema) für den Empfang von Nachrichten im Falle einer Lebenszyklusaktion einzurichten. Amazon SNS sendet die Benachrichtigungen dann an die abonnierten Empfänger. Solange das Abonnement nicht bestätigt ist, werden keine Benachrichtigungen, die zum Thema veröffentlicht wurden, an die Empfänger gesendet.

Einrichten von Benachrichtigungen mithilfe von Amazon SNS

1. Erstellen Sie ein Amazon SNS-Thema mithilfe der Amazon SNS Konsole oder dem folgenden create-topic-Befehl. Stellen Sie sicher, dass sich das Thema in derselben Region befindet wie

die verwendete Auto-Scaling-Gruppe. Weitere Informationen finden Sie unter Erste Schritte mit Amazon SNS im Benutzerhandbuch für Amazon Simple Notification Service.

```
aws sns create-topic --name my-sns-topic
```

2. Notieren Sie den Amazon-Ressourcennamen (ARN) des Themas, zum Beispiel `arn:aws:sns:region:123456789012:my-sns-topic`. Sie benötigen ihn, um den Lebenszyklus-Hook zu erstellen.
3. Erstellen Sie eine IAM-Servicerolle, um Amazon EC2 Auto Scaling Zugriff auf Ihr Amazon SNS-Benachrichtigungsziel zu gewähren.

   So gewähren Sie Amazon EC2 Auto Scaling Zugriff auf Ihr SNS-Thema

   a. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
   b. Wählen Sie im Navigationsbereich auf der linken Seite Roles (Rollen).
   c. Wählen Sie Rolle erstellen aus.
   d. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die Option AWS -Dienst.
   e. Wählen Sie für Ihren Anwendungsfall unter Use cases for other AWS services (Anwendungsfälle für andere -Dienste), EC2 Auto Scaling (EC2 Auto Scaling) und dann EC2 Auto Scaling Notification Access (Zugriff auf EC2-Auto-Scaling-Benachrichtigungen) aus.
   f. Klicken Sie zweimal auf Next (Weiter), um zur Seite Name, review, and create (Benennen, überprüfen und erstellen) zu gelangen.
   g. Geben Sie für Role Name (Name der Rolle) einen Namen für Ihre Rolle ein (z. B. **`my-notification-role`**) und wählen Sie dann Create role (Rolle erstellen).
   h. Wählen Sie auf der Seite Roles (Rollen) die gerade erstellte Rolle aus, um die Seite Summary (Übersicht) zu öffnen. Notieren Sie sich den ARN der Rolle. z. B. `arn:aws:iam::123456789012:role/my-notification-role`. Sie benötigen ihn, um den Lebenszyklus-Hook zu erstellen.
4. Nachdem Sie diese Anweisungen befolgt haben, fahren Sie mit Hinzufügen von Lebenszyklus-Hooks (AWS CLI) fort.

Benachrichtigungen über Amazon SQS erhalten

Sie können Amazon SQS dazu verwenden, ein Benachrichtigungsziel für den Empfang von Nachrichten im Falle einer Lebenszyklusaktion einzurichten. Ein Warteschlangen-Verbraucher muss dann eine SQS-Warteschlange abfragen, um auf diese Benachrichtigungen zu reagieren.

> **Important**
>
> FIFO-Warteschlangen sind nicht kompatibel mit Lebenszyklus-Hooks.

Einrichten von Benachrichtigungen mithilfe von Amazon SQS

1. Mit der Amazon SQS-Konsole erstellen Sie eine SQS-Warteschlange. Stellen Sie sicher, dass sich die Warteschlange in derselben Region befindet wie die von Ihnen verwendete Auto-Scaling-Gruppe. Weitere Informationen finden Sie unter Erste Schritte mit Amazon SQS im Benutzerhandbuch für Amazon Simple Queue Service.
2. Notieren Sie den ARN der Warteschlange, z. B. `arn:aws:sqs:us-west-2:123456789012:my-sqs-queue`. Sie benötigen ihn, um den Lebenszyklus-Hook zu erstellen.
3. Erstellen Sie eine IAM-Servicerolle, um Amazon EC2 Auto Scaling Zugriff auf Ihr Amazon SQS-Benachrichtigungsziel zu gewähren.

   So gewähren Sie Amazon EC2 Auto Scaling Zugriff auf Ihre SQS-Warteschlange

   a. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
   b. Wählen Sie im Navigationsbereich auf der linken Seite Roles (Rollen).
   c. Wählen Sie Rolle erstellen aus.
   d. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die Option AWS -Dienst.
   e. Wählen Sie für Ihren Anwendungsfall unter Use cases for other AWS services (Anwendungsfälle für andere -Dienste), EC2 Auto Scaling (EC2 Auto Scaling) und dann EC2 Auto Scaling Notification Access (Zugriff auf EC2-Auto-Scaling-Benachrichtigungen) aus.
   f. Klicken Sie zweimal auf Next (Weiter), um zur Seite Name, review, and create (Benennen, überprüfen und erstellen) zu gelangen.
   g. Geben Sie für Role Name (Name der Rolle) einen Namen für Ihre Rolle ein (z. B. **`my-notification-role`**) und wählen Sie dann Create role (Rolle erstellen).

h. Wählen Sie auf der Seite Roles (Rollen) die gerade erstellte Rolle aus, um die Seite Summary (Übersicht) zu öffnen. Notieren Sie sich den ARN der Rolle. z. B. `arn:aws:iam::123456789012:role/my-notification-role`. Sie benötigen ihn, um den Lebenszyklus-Hook zu erstellen.

4. Nachdem Sie diese Anweisungen befolgt haben, fahren Sie mit Hinzufügen von Lebenszyklus-Hooks (AWS CLI) fort.

Beispiel einer Benachrichtigungsnachricht für Amazon SNS und Amazon SQS

Während sich die Instance in einem Wartestatus befindet, wird im Amazon SNS- oder Amazon SQS-Benachrichtigungsziel eine Nachricht veröffentlicht. Die Nachricht enthält die folgenden Informationen:

- `LifecycleActionToken` – Das Token der Lebenszyklusaktion
- `AccountId`— Die AWS-Konto ID.
- `AutoScalingGroupName` – Der Name der Auto-Scaling-Gruppe.
- `LifecycleHookName` – Der Name des Lebenszyklus-Hooks.
- `EC2InstanceId` – Die ID der EC2-Instance.
- `LifecycleTransition` – Die Art des Lebenszyklus-Hooks.
- `NotificationMetadata` – Die Benachrichtigungsmetadaten.

Im Folgenden finden Sie ein Beispiel für eine Benachrichtigungsmeldung.

```
Service: AWS Auto Scaling
Time: 2021-01-19T00:36:26.533Z
RequestId: 18b2ec17-3e9b-4c15-8024-ff2e8ce8786a
LifecycleActionToken: 71514b9d-6a40-4b26-8523-05e7ee35fa40
AccountId: 123456789012
AutoScalingGroupName: my-asg
LifecycleHookName: my-hook
EC2InstanceId: i-0598c7d356eba48d7
LifecycleTransition: autoscaling:EC2_INSTANCE_LAUNCHING
NotificationMetadata: hook message metadata
```

Beispiel für Benachrichtigungsnachricht testen

Wenn Sie zum ersten Mal einen Lebenszyklus-Hook hinzufügen, wird eine Testbenachrichtigung für das Benachrichtigungsziel veröffentlicht. Im Folgenden finden Sie ein Beispiel für eine Testbenachrichtigungsnachricht.

```
Service: AWS Auto Scaling
Time: 2021-01-19T00:35:52.359Z
RequestId: 18b2ec17-3e9b-4c15-8024-ff2e8ce8786a
Event: autoscaling:TEST_NOTIFICATION
AccountId: 123456789012
AutoScalingGroupName: my-asg
AutoScalingGroupARN: arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:042cba90-
ad2f-431c-9b4d-6d9055bcc9fb:autoScalingGroupName/my-asg
```

> **Note**
>
> Beispiele für Ereignisse, die von Amazon EC2 Auto Scaling an übermittelt wurden EventBridge, finden Sie unterEreignis-Referenz für Amazon EC2 Auto Scaling.

Abrufen des Ziellebenszyklus-Status durch Instance-Metadaten

Jede Auto-Scaling-Instance, die Sie starten, durchläuft mehrere Lebenszyklus-Status. Um aus einer Instance heraus benutzerdefinierte Aktionen aufzurufen, die auf bestimmte Lebenszyklusstatus-Übergänge wirken, müssen Sie den Ziellebenszyklus-Status über Instance-Metadaten abrufen.

Beispielsweise benötigen Sie möglicherweise einen Mechanismus, um die Instance-Beendigung innerhalb der Instance zu erkennen, um Code auf der Instance auszuführen, bevor sie beendet wird. Sie können dies tun, indem Sie Code schreiben, der den Lebenszyklus-Status einer Instance direkt von der Instance aus abfragt. Anschließend können Sie der Auto-Scaling-Gruppe einen Lebenszyklus-Hook hinzufügen, um die Instance so lange am Laufen zu halten, bis Ihr Code den Befehl complete-lifecycle-action zum Fortfahren sendet.

Der Lebenszyklus der Auto-Scaling-Instance hat zwei primäre Beharrungszustände – `InService` und `Terminated` – und zwei sekundäre Beharrungszustände – `Detached` und `Standby`. Wenn Sie einen Warm-Pool verwenden, hat der Lebenszyklus vier zusätzliche Beharrungszustände – `Warmed:Hibernated`, `Warmed:Running`, `Warmed:Stopped` und`Warmed:Terminated`.

Wenn sich eine Instance auf den Übergang in einen der vorhergehenden Beharrungszustände vorbereitet, aktualisiert Amazon EC2 Auto Scaling den Wert des Instance-Metadatenelements `autoscaling/target-lifecycle-state`. Um den Ziellebenszyklusstatus innerhalb der Instance abzurufen, müssen Sie den Instance-Metadatendienst verwenden, um ihn aus den Instance-Metadaten abzurufen.

> **Note**
>
> Instance-Metadaten sind Daten über eine Amazon-EC2-Instance, mit denen Anwendungen Instance-Informationen abfragen können. Der Instance-Metadatenservice (IMDS) ist eine On-Instance-Komponente, die von lokalem Code verwendet wird, um auf Instance-Metadaten zuzugreifen. Lokaler Code kann Benutzerdatenskripte oder Anwendungen enthalten, die auf der Instance ausgeführt werden.

Lokaler Code kann auf Instance-Metadaten von einer ausgeführten Instance mit einer von zwei Methoden zugreifen: Instance Metadata Service Version 1 (IMDSv1) oder Instance Metadata Service Version 2 (IMDSv2). IMDSv2 verwendet sitzungsorientierte Anfragen und mildert verschiedene Arten von Sicherheitsschwachstellen, über die versucht werden kann, auf die Instance-Metadaten zuzugreifen. Einzelheiten zu diesen beiden Methoden finden Sie unter Verwenden von IMDSv2 im Amazon EC2 EC2-Benutzerhandbuch.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/autoscaling/target-lifecycle-state
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/autoscaling/target-lifecycle-state
```

Es folgt eine Beispielausgabe.

```
InService
```

Der Ziellebenszyklusstatus ist der Status, in den die Instance wechselt. Der aktuelle Lebenszyklusstatus ist der Status, in dem sich die Instance befindet. Diese können gleich sein, nachdem die Lebenszyklusaktion abgeschlossen ist und die Instance ihren Übergang in den Ziellebenszyklusstatus beendet hat. Sie können den aktuellen Lebenszyklusstatus der Instance nicht aus den Instance-Metadaten abrufen.

Amazon EC2 Auto Scaling begann am 10. März 2022 mit der Generierung des Ziellebenszyklusstatus. Wenn Ihre Instance nach diesem Datum in einen der Ziellebenszyklusstatus wechselt, ist das Ziellebenszyklusstatus-Element in den Instance-Metadaten vorhanden. Andernfalls ist es nicht vorhanden und Sie erhalten einen HTTP-404-Fehler.

Weitere Informationen zum Abrufen von Instance-Metadaten finden Sie unter Instance-Metadaten abrufen im Amazon EC2 EC2-Benutzerhandbuch.

Ein Tutorial, das Ihnen zeigt, wie Sie einen Lebenszyklus-Hook mit einer benutzerdefinierten Aktion in einem Benutzerdatenskript erstellen, das den Ziel-Lebenszyklusstatus verwendet, finden Sie unter Tutorial: Konfigurieren Sie Benutzerdaten zum Abrufen des Ziellebenszyklusstatus über Instance-Metadaten.

> **Important**
>
> Um sicherzustellen, dass Sie so schnell wie möglich eine benutzerdefinierte Aktion aufrufen können, sollte Ihr lokaler Code IMDS häufig abfragen und es bei Fehlern erneut versuchen.

Lebenszyklus-Hooks hinzufügen

Um Ihre Instances mit automatischer Skalierung in einen Wartezustand zu versetzen und benutzerdefinierte Aktionen für sie durchzuführen, können Sie Ihrer Auto-Scaling-Gruppe Lebenszyklus-Hooks hinzufügen. Benutzerdefinierte Aktionen werden beim Start der Instances oder vor dem Beenden ausgeführt. Die Instances bleiben in einem Wartezustand, bis Sie entweder die Lebenszyklusaktion beenden oder der Timeout-Zeitraum endet.

Nachdem Sie aus der eine Auto Scaling Scaling-Gruppe erstellt haben AWS Management Console, können Sie ihr einen oder mehrere Lifecycle-Hooks hinzufügen, bis zu insgesamt 50 Lifecycle-Hooks. Sie können auch das AWS CLI, oder ein SDK verwenden AWS CloudFormation, um einer Auto Scaling Scaling-Gruppe Lifecycle-Hooks hinzuzufügen, während Sie sie erstellen.

Wenn Sie einen Lifecycle-Hook in der Konsole hinzufügen, sendet Amazon EC2 Auto Scaling standardmäßig Lebenszyklusereignisbenachrichtigungen an Amazon EventBridge. Die Verwendung

EventBridge eines Benutzerdatenskripts ist eine empfohlene bewährte Methode. Um einen Lebenszyklus-Hook zu erstellen, der Benachrichtigungen direkt an Amazon SNS oder Amazon SQS sendet, können Sie den Befehl put-lifecycle-hook, wie in den Beispielen in diesem Thema gezeigt, verwenden.

Inhalt

Lebenszyklus-Hooks hinzufügen (Konsole)

Gehen Sie folgendermaßen vor, um Ihrer Auto-Scaling-Gruppe einen Lebenszyklus-Hook hinzuzufügen. Um Lebenszyklus-Hooks zum Aufskalieren (Starten von Instances) und Abskalieren (Beenden von Instances oder bei der Rückkehr zu einem warmen Pool) zu erstellen, müssen Sie zwei separate Hooks erstellen.

Bevor Sie beginnen, vergewissern Sie sich, dass Sie bei Bedarf eine benutzerdefinierte Aktion eingerichtet haben, wie unter Vorbereiten des Hinzufügens eines Lebenszyklus-Hook zu einer Auto-Scaling-Gruppe beschrieben.

So fügen Sie einen Lebenszyklus-Hook für das Aufskalieren hinzu

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe. Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Instance management (Instance-Verwaltung) unter Lebenszyklus-Hooks die Option Create Lebenszyklus hook (Lebenszyklus-Hook erstellen) aus.
4. Gehen Sie wie folgt vor, um einen Lebenszyklus-Hook zum Aufskalieren (Start von Instances) zu definieren:

    a. Geben Sie bei Lebenszyklus hook name (Name des Lebenszyklus-Hooks) einen Namen für den Lebenszyklus-Hook an.

    b. Wählen Sie bei Lifecycle Transition (Lebenszykluswechsel) die Option Instance launch (Instance-Start) aus.

    c. Geben Sie für Heartbeat-Zeitüberschreitung die Zeitspanne in Sekunden an, für die Instances in einem Wartezustand verbleiben sollen, wenn Sie aufskalieren, bevor die

Zeitüberschreitung des Hook erreicht ist. Der Bereich liegt zwischen 30 und 7200 Sekunden. Das Festlegen eines langen Timeout-Zeitraums bietet mehr Zeit für den Abschluss der benutzerdefinierten Aktion. Falls Sie zum Ende kommen, bevor eine Zeitüberschreitung auftritt, verwenden Sie den Befehl complete-lifecycle-action, damit die Instance in den nächsten Status übergehen kann.

d. Definieren Sie für ein Default result (Standardergebnis) die Aktion, die ausgeführt werden soll, wenn für den Lebenszyklus-Hook eine Zeitüberschreitung oder ein unerwarteter Fehler auftritt. Sie können entweder FORTSETZEN oder ABBRUCH auswählen.

   - Wenn Sie FORTSETZEN wählen, kann die Auto-Scaling-Gruppe mit allen anderen Lebenszyklus-Hooks fortfahren und die Instance dann in Betrieb nehmen.
   - Wenn Sie ABBRUCH wählen, beendet die Auto-Scaling-Gruppe alle verbleibenden Aktionen und beendet die Instance sofort.

e. (Optional) Geben Sie bei Benachrichtigungsmetadaten die weiteren Informationen an, die Sie hinzufügen möchten, wenn Amazon EC2 Auto Scaling eine Nachricht an das Benachrichtigungsziel sendet.

5. Wählen Sie Erstellen.

So fügen Sie einen Lebenszyklus-Hook für das Abskalieren hinzu

1. Wählen Sie Lebenszyklus-Hook erstellen, um dort weiterzumachen, wo Sie aufgehört haben, nachdem Sie einen Lebenszyklus-Hook für die horizontale Aufskalierung erstellt haben.
2. Gehen Sie wie folgt vor, um einen Lebenszyklus-Hook für die Abskalierung zu definieren (Instances, die beendet werden oder zu einem warmen Pool zurückkehren):

   a. Geben Sie bei Lebenszyklus hook name (Name des Lebenszyklus-Hooks) einen Namen für den Lebenszyklus-Hook an.

   b. Wählen Sie bei Lifecycle Transition (Lebenszykluswechsel) die Option Instance Terminate (Instance-Beendigung) aus.

   c. Geben Sie für Heartbeat-Zeitüberschreitung die Zeitspanne in Sekunden an, für die Instances in einem Wartezustand verbleiben sollen, wenn Sie aufskalieren, bevor die Zeitüberschreitung des Hook erreicht ist. Wir empfehlen ein kurzes Timeout von ein 30 bis zwei 120 Sekunden, je nachdem, wie viel Zeit Sie für die Ausführung der letzten Aufgaben benötigen, wie z. B. das Abrufen von EC2-Protokollen. CloudWatch

   d. Geben Sie als Default Result (Standardergebnis) die Aktion an, die von der Auto-Scaling-Gruppe ausführt wird, wenn für den Lebenszyklus-Hook ein Timeout oder ein unerwarteter

Fehler auftritt. Sowohl ABANDON (ABBRUCH) als auch CONTINUE (FORTSETZEN) ermöglichen das Beenden der Instance.

- Wenn Sie CONTINUE (FORTSETZEN) wählen, kann die Auto-Scaling-Gruppe vor der Beendigung alle verbleibenden Aktionen, z. B. andere Lebenszyklus-Hooks, ausführen.
- Wenn Sie ABBRUCH wählen, beendet die Auto-Scaling-Gruppe die Instance sofort.

e. (Optional) Geben Sie bei Benachrichtigungsmetadaten die weiteren Informationen an, die Sie hinzufügen möchten, wenn Amazon EC2 Auto Scaling eine Nachricht an das Benachrichtigungsziel sendet.

3. Wählen Sie Erstellen.

Hinzufügen von Lebenszyklus-Hooks (AWS CLI)

Lebenszyklus-Hooks erstellen und aktualisieren Sie mit dem Befehl put-lifecycle-hook.

Verwenden Sie den folgenden Befehl zum Ausführen einer Aktion bei einer horizontalen Skalierung nach oben:

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-launch-hook  \
  --auto-scaling-group-name my-asg \
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING
```

Verwenden Sie hingegen den folgenden Befehl, um bei einer horizontalen Skalierung nach unten eine Aktion durchzuführen:

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-termination-hook  \
  --auto-scaling-group-name my-asg \
  --lifecycle-transition autoscaling:EC2_INSTANCE_TERMINATING
```

Um Benachrichtigungen mit Amazon SNS oder Amazon SQS zu empfangen, fügen Sie die Optionen `--notification-target-arn` und `--role-arn` hinzu.

Im folgenden Beispiel wird ein Lebenszyklus-Hook erstellt, der ein SNS-Thema mit dem Namen *my-sns-topic* als Benachrichtigungsziel definiert.

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-termination-hook  \
  --auto-scaling-group-name my-asg \
  --lifecycle-transition autoscaling:EC2_INSTANCE_TERMINATING \
```

```
  --notification-target-arn arn:aws:sns:region:123456789012:my-sns-topic \
  --role-arn arn:aws:iam::123456789012:role/my-notification-role
```

Das Thema erhält eine Testbenachrichtigung mit dem folgenden Schlüssel-Wert-Paar:

```
"Event": "autoscaling:TEST_NOTIFICATION"
```

Standardmäßig erstellt der Befehl put-lifecycle-hook einen Lebenszyklus-Hook mit einem Heartbeat-Timeout von `3600` Sekunden (eine Stunde).

Um das Heartbeat-Timeout für einen vorhandenen Lebenszyklus-Hook zu ändern, fügen Sie die Option `--heartbeat-timeout` hinzu, wie im folgenden Beispiel gezeigt.

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-termination-hook \
  --auto-scaling-group-name my-asg --heartbeat-timeout 120
```

Wenn sich eine Instance bereits im Wartezustand befindet, können Sie verhindern, dass der Lebenszyklus-Hook das Time-Out hat, indem Sie durch Verwendung des record-lifecycle-action-heartbeat-CLI-Befehls einen Heartbeat aufzeichnen. Diese erweitert die Zeitüberschreitung um den Zeitüberschreitungswert, den Sie bei der Erstellung des Lebenszyklus-Hooks festgelegt haben. Falls Sie zum Ende kommen, bevor eine Zeitüberschreitung auftritt, verwenden Sie den CLI-Befehl complete-lifecycle-action, damit die Instance in den nächsten Status übergehen kann. Weitere Informationen und Beispiele finden Sie unter Eine Lebenszyklus-Aktion abschließen.

Eine Lebenszyklus-Aktion abschließen

Reagiert eine Auto-Scaling-Gruppe auf ein Lebenszyklus-Ereignis, versetzt sie die Instance in einen Wartestatus und sendet eine Ereignisbenachrichtigung. Sie können eine benutzerdefinierte Aktion ausführen, während sich die Instance in einem Wartestatus befindet.

Das Abschließen der Lebenszyklus-Aktion mit dem Ergebnis von `CONTINUE` ist hilfreich, wenn Sie den Vorgang vor Ablauf des Timeouts beenden. Wenn Sie die Lebenszyklus-Aktion nicht abschließen, nimmt der Lebenszyklus-Hook nach Ablauf des Timeout-Zeitraums den Status an, den Sie als Standardergebnis angegeben haben.

Inhalt

Eine Lebenszyklus-Aktion abschließen (manuell)

Das folgende Verfahren gilt für die Befehlszeilenschnittstelle und wird in der Konsole nicht unterstützt. Die zu ersetzenden Informationen wie die Instance-ID oder der Name einer Auto-Scaling-Gruppe werden kursiv dargestellt.

So führen Sie eine Lebenszyklus-Aktion aus (AWS CLI)

1. Falls Sie mehr Zeit für die benutzerdefinierte Aktion benötigen, verwenden Sie den Befehl record-Lebenszyklus-action-heartbeat, um die Zeit für die Zeitüberschreitung zurückzusetzen und die Instance im Wartestatus zu belassen. Beträgt der Zeitüberschreitungszeitraum z. B. eine Stunde und Sie rufen diesen Befehl nach 30 Minuten auf, verbleibt die Instance für eine zusätzliche Stunde bzw. insgesamt 90 Minuten in einem Wartestatus.

   Sie können das Token der Lebenszyklusaktion das Sie mit der Benachrichtigung erhalten haben, wie im folgenden Befehl gezeigt angeben.

   ```
   aws autoscaling record-lifecycle-action-heartbeat --lifecycle-hook-name my-launch-hook \
     --auto-scaling-group-name my-asg --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
   ```

   Alternativ können Sie auch die ID der Instance angeben, die Sie mit der Benachrichtigung erhalten haben, wie im folgenden Befehl gezeigt.

   ```
   aws autoscaling record-lifecycle-action-heartbeat --lifecycle-hook-name my-launch-hook \
     --auto-scaling-group-name my-asg --instance-id i-1a2b3c4d
   ```

2. Wenn Sie die benutzerdefinierte Aktion abschließen, bevor die Zeit für die Zeitüberschreitung erreicht wurde, verwenden Sie den Befehl complete-lifecycle-action, damit die Auto-Scaling-Gruppe den Start bzw. die Beendigung der Instance fortführen kann. Sie können das Token für die Lebenszyklusaktion wie im folgenden Befehl angeben:

   ```
   aws autoscaling complete-lifecycle-action --lifecycle-action-result CONTINUE \
     --lifecycle-hook-name my-launch-hook --auto-scaling-group-name my-asg \
     --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
   ```

   Alternativ können Sie die ID der Instance wie im folgenden Befehl angeben:

```
aws autoscaling complete-lifecycle-action --lifecycle-action-result CONTINUE \
  --instance-id i-1a2b3c4d --lifecycle-hook-name my-launch-hook \
  --auto-scaling-group-name my-asg
```

Eine Lebenszyklus-Aktion abschließen (automatisch)

Wenn Sie ein Skript mit Benutzerdaten haben, das Ihre Instances nach dem Start konfiguriert, müssen Sie die Lebenszyklusaktionen nicht manuell durchführen. Sie können dem Skript den Befehl complete-lifecycle-action hinzufügen. Das Skript kann die Instance-ID aus den Instance-Metadaten abrufen und Amazon EC2 Auto Scaling signalisieren, wenn die Bootstrap-Skripte erfolgreich abgeschlossen wurden.

Wenn Sie nicht bereits dabei sind, aktualisieren Sie das Skript, sodass es die Instance-ID der Instance aus den Instance-Metadaten abruft. Weitere Informationen finden Sie unter Instance-Metadaten abrufen im Amazon EC2 EC2-Benutzerhandbuch.

Wenn Sie Lambda verwenden, können Sie auch einen Rückruf im Code Ihrer Funktion einrichten, damit der Lebenszyklus der Instance fortgesetzt werden kann, wenn die benutzerdefinierte Aktion erfolgreich ist. Weitere Informationen finden Sie unter Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft.

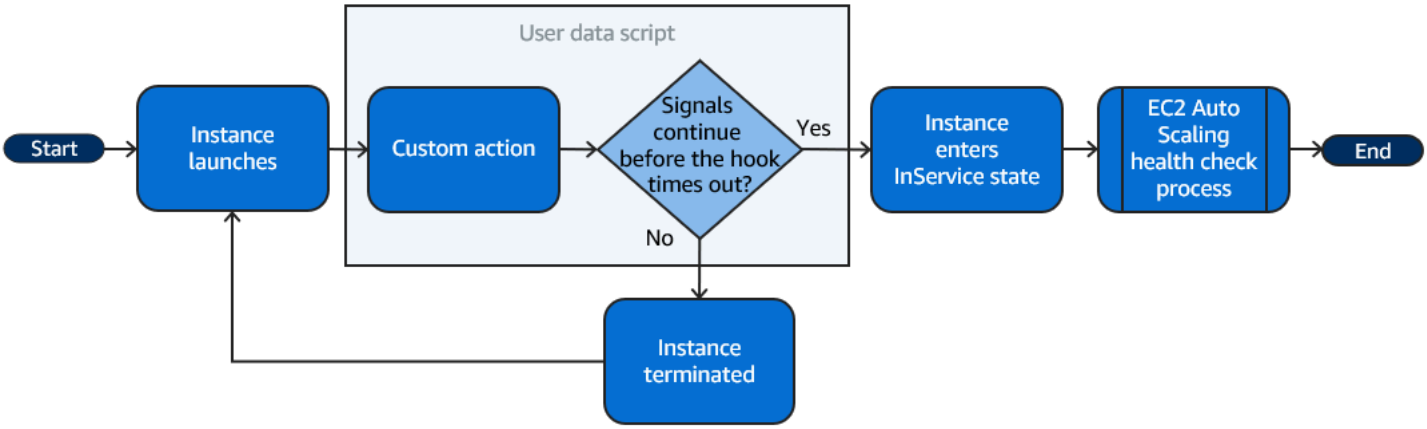
Tutorial: Konfigurieren Sie Benutzerdaten zum Abrufen des Ziellebenszyklusstatus über Instance-Metadaten

Eine gängige Methode, benutzerdefinierte Aktionen für Lifecycle-Hooks zu erstellen, ist die Verwendung von Benachrichtigungen, die Amazon EC2 Auto Scaling an andere Dienste wie Amazon EventBridge sendet. Sie können jedoch vermeiden, dass Sie zusätzliche Infrastruktur erstellen müssen, indem Sie stattdessen ein Benutzerdatenskript verwenden, um den Code, der Instances konfiguriert und die Lebenszyklusaktion abschließt, in die Instances selbst zu verschieben.

Das folgende Tutorial veranschaulicht die Verwendung eines Benutzerdatenskripts und Instance-Metadaten. Sie erstellen eine grundlegende Auto-Scaling-Gruppenkonfiguration mit einem Benutzerdatenskript, das die Ziel-Lebenszyklusstatus der Instances in Ihrer Gruppe liest und eine Rückrufaktion in einer bestimmten Phase des Lebenszyklus einer Instance ausführt, um den Startprozess fortzusetzen.

Die folgende Abbildung fasst den Ablauf für ein Scale-Out-Ereignis zusammen, wenn Sie ein Benutzerdatenskript verwenden, um eine benutzerdefinierte Aktion auszuführen. Nach dem Start

einer Instance wird der Lebenszyklus der Instance angehalten, bis der Lifecycle-Hook abgeschlossen ist, entweder durch eine Zeitüberschreitung oder dadurch, dass Amazon EC2 Auto Scaling ein Signal zum Fortfahren empfängt.



Inhalt

Schritt 1: Erstellen einer IAM-Rolle mit Berechtigungen zum Abschließen von Lebenszyklus-Aktionen

Wenn Sie das AWS CLI oder ein AWS SDK verwenden, um einen Rückruf zu senden, um Lebenszyklusaktionen abzuschließen, müssen Sie eine IAM-Rolle mit Berechtigungen zum Abschließen von Lebenszyklusaktionen verwenden.

So erstellen Sie die Richtlinie

1. Öffnen Sie in der IAM-Konsole Policies (Richtlinien) und wählen Sie dann Create policy (Richtlinie erstellen) aus.

2. Wählen Sie den Tab JSON.
3. Kopieren Sie das folgende Richtliniendokument und fügen Sie es in das Feld Policy Document (Richtliniendokument) ein. Ersetzen Sie den *Beispieltext* mit Ihrer Kontonummer und dem Namen der Auto-Scaling-Gruppe, die Sie erstellen möchten (**TestAutoScalingEvent-group**).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction"
      ],
      "Resource":
 "arn:aws:autoscaling:*:123456789012:autoScalingGroup:*:autoScalingGroupName/
TestAutoScalingEvent-group"
    }
  ]
}
```

4. Wählen Sie Weiter aus.
5. Geben Sie unter Policy name (Richtlinienname) **TestAutoScalingEvent-policy** ein. Wählen Sie Richtlinie erstellen aus.

Wenn Sie die Richtlinie fertig erstellt haben, können Sie eine Rolle erstellen, die sie verwendet.

So erstellen Sie die Rolle

1. Wählen Sie im Navigationsbereich auf der linken Seite Roles (Rollen).
2. Wählen Sie Rolle erstellen aus.
3. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die Option AWS - Service.
4. Wählen Sie für Ihren Anwendungsfall die Option EC2 und dann Next (Weiter) aus.
5. Wählen Sie unter Berechtigungen hinzufügen die Richtlinie aus, die Sie erstellt haben (TestAutoScalingEvent-policy). Wählen Sie anschließend Weiter.

6. Geben Sie auf der Seite Set role name and review (Rollenname festlegen und überprüfen) für Role name (Rollenname) **`TestAutoScalingEvent-role`** ein und wählen Sie Create role (Rolle erstellen) aus.

Schritt 2: Erstellen Sie eine Startvorlage und schließen Sie die IAM-Rolle und ein Benutzerdatenskript ein

Erstellen Sie eine Startvorlage für die Verwendung mit einer Auto-Scaling-Gruppe. Fügen Sie die von Ihnen erstellte IAM-Rolle und das bereitgestellte Beispielbenutzerdatenskript ein.

Eine Startvorlage erstellen

1. Öffnen Sie die Seite Startvorlagen der Amazon EC2 Konsole.
2. Wählen Sie Startvorlage erstellen.
3. Geben Sie für Startvorlagenname **`TestAutoScalingEvent-template`** ein.
4. Unter Auto-Scaling-Anleitung aktivieren Sie das Kontrollkästchen.
5. Wählen Sie für Application and OS Images (Amazon Machine Image) (Anwendungs- und Betriebssystem-Images (Amazon Machine Image)) Amazon Linux 2 (HVM), SSD-Volume-Typ, 64-Bit (x86) aus der Quick Start(Schnellstart)-Liste.
6. Wählen Sie für Instance type (Instance-Typ) einen Typ von Amazon-EC2-Instance (z. B. „t2.micro“) aus.
7. Für Erweiterte Details erweitern Sie den Abschnitt, um die Felder anzuzeigen.
8. Wählen Sie für das IAM-Instanzprofil den Namen des IAM-Instanzprofils Ihrer IAM-Rolle (-role) aus. TestAuto ScalingEvent Ein Instance-Profil ist ein Container für eine IAM-Rolle, mit dem Amazon EC2 einer Instance die IAM-Rolle übergibt, wenn die Instance gestartet wird.

   Wenn Sie eine IAM-Rolle mithilfe der IAM-Konsole erstellt haben, hat die Konsole automatisch ein Instance-Profil mit demselben Namen wie der entsprechenden Rolle erzeugt.
9. Kopieren Sie für User date (Benutzerdaten) das folgende Beispiel-Benutzerdatenskript und fügen Sie es in das Feld ein. Ersetzen Sie den Beispieltext für `group_name` durch den Namen der Auto Scaling Scaling-Gruppe, die Sie erstellen möchten, und `region` durch den Namen, den AWS-Region Ihre Auto Scaling Scaling-Gruppe verwenden soll.

```
#!/bin/bash

function get_target_state {
```

```
    echo $(curl -s http://169.254.169.254/latest/meta-data/autoscaling/target-
lifecycle-state)
}

function get_instance_id {
    echo $(curl -s http://169.254.169.254/latest/meta-data/instance-id)
}

function complete_lifecycle_action {
    instance_id=$(get_instance_id)
    group_name='TestAutoScalingEvent-group'
    region='us-west-2'

    echo $instance_id
    echo $region
    echo $(aws autoscaling complete-lifecycle-action \
      --lifecycle-hook-name TestAutoScalingEvent-hook \
      --auto-scaling-group-name $group_name \
      --lifecycle-action-result CONTINUE \
      --instance-id $instance_id \
      --region $region)
}

function main {
    while true
    do
        target_state=$(get_target_state)
        if [ \"$target_state\" = \"InService\" ]; then
            # Change hostname
            export new_hostname="${group_name}-$instance_id"
            hostname $new_hostname
            # Send callback
            complete_lifecycle_action
            break
        fi
        echo $target_state
        sleep 5
    done
}

main
```

Dieses einfache Benutzerdatenskript führt folgende Aktionen aus:

- Ruft die Instance-Metadaten auf, um den Ziellebenszyklusstatus und die Instance-ID aus den Instance-Metadaten abzurufen
- Ruft den Ziellebenszyklusstatus wiederholt ab, bis er sich auf `InService` ändert
- Ändert den Hostnamen der Instance in die Instance-ID, der dem Namen der Auto-Scaling-Gruppe vorangestellt ist, wenn der Ziellebenszyklusstatus `InService` lautet
- Sendet einen Rückruf durch Aufruf des CLI-Befehls complete-lifecycle-action, um Amazon EC2 Auto Scaling zu signalisieren, den EC2-Startprozess zu `CONTINUE`

10. Wählen Sie Startvorlage erstellen.
11. Wählen Sie auf der Bestätigungsseite Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

> **Note**
>
> Weitere Beispiele, die Sie als Referenz für die Entwicklung Ihres Benutzerdatenskripts verwenden können, finden Sie im GitHub Repository für Amazon EC2 Auto Scaling.

Schritt 3: Erstellen einer Auto-Scaling-Gruppe

Nachdem Sie die Startvorlage erstellt haben, erstellen Sie eine Auto-Scaling-Gruppe.

So erstellen Sie eine Auto Scaling-Gruppe

1. Geben Sie auf der Seite Choose launch template or configuration (Startvorlage oder -konfiguration auswählen) für Auto Scaling group name (Auto-Scaling-Gruppenname) einen Namen für Ihre Auto-Scaling-Gruppe ein (**`TestAutoScalingEvent-group`**).
2. Klicken Sie auf Next (Weiter), um die Seite Choose instance launch options (Wählen Sie Instance-Startoptionen) aufzurufen.
3. Wählen Sie unter Network (Netzwerk) eine VPC aus.
4. Wählen Sie für Availability Zones and subnets (Availability Zones und Subnetze) ein oder mehrere Subnetze aus einer oder mehreren Availability Zones aus.
5. Verwenden Sie im Abschnitt Instance type requirements (Anforderungen an den Instance-Typ) die Standardeinstellung, um diesen Schritt zu vereinfachen. (Setzen Sie die Startvorlage nicht außer Kraft.) In diesem Tutorial werden Sie nur eine On-Demand-Instance mit dem in Ihrer Startvorlage angegebenen Instance-Typ starten.

6. Wählen Sie unten auf dem Bildschirm Überspringen zum Review.
7. Überprüfen Sie auf der Seite Review (Prüfen) die Details Ihrer Auto-Scaling-Gruppe und wählen Sie dann Create Auto Scaling group (Auto-Scaling-Gruppe erstellen).

Schritt 4: Hinzufügen eines Lebenszyklus-Hooks

Fügen Sie einen Lebenszyklus-Hook hinzu, um die Instance in einem Wartezustand zu halten, bis Ihre Lebenszyklusaktion abgeschlossen ist.

So fügen Sie einen Lebenszyklus-Hook hinzu

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe. Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie im unteren Bereich auf der Registerkarte Instance management (Instance-Verwaltung) unter Lebenszyklus-Hooks die Option Create Lebenszyklus hook (Lebenszyklus-Hook erstellen) aus.
4. Gehen Sie wie folgt vor, um einen Lebenszyklus-Hook zum Aufskalieren (Start von Instances) zu definieren:
   a. Geben Sie für Lebenszyklus-Hooks den Wert **`TestAutoScalingEvent-hook`** ein.
   b. Wählen Sie bei Lifecycle Transition (Lebenszykluswechsel) die Option Instance launch (Instance-Start) aus.
   c. Geben Sie für Heartbeat timeout (Heartbeat-Zeitüberschreitung) den Wert **`300`** für die Anzahl der Sekunden ein, die auf einen Rückruf von Ihrem Benutzerdatenskript gewartet werden soll.
   d. Für Standardergebnis wählen Sie ABBRECHEN aus. Wenn die Zeitüberschreitung des Hooks erreicht ist, ohne einen Rückruf von Ihrem Benutzerdatenskript erhalten zu haben, beendet die Auto-Scaling-Gruppe die neue Instance.
   e. (Optional) Halten Sie Notification metadata (Benachrichtigungs-Metadaten) frei.
5. Wählen Sie Erstellen.

Schritt 5: Testen und Prüfen der Funktionalität

Um die Funktionalität zu testen, aktualisieren Sie die Auto-Scaling-Gruppe, indem Sie die gewünschte Kapazität der Auto-Scaling-Gruppe um 1 erhöhen. Das Benutzerdatenskript wird

ausgeführt und überprüft den Ziellebenszyklusstatus der Instance kurz nach dem Start der Instance. Das Skript ändert den Hostnamen und sendet eine Rückrufaktion, wenn der Ziellebenszyklusstatus `InService` ist. Dieser Vorgang dauert normalerweise nur ein paar Sekunden.

So erhöhen Sie die Größe der Auto-Scaling-Gruppe

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe. Zeigen Sie Details in einem unteren Bereich an, während Sie weiterhin die oberen Zeilen des oberen Bereichs sehen.
3. Wählen Sie im unteren Bereich auf der Registerkarte Details die Option Gruppendetails, Bearbeiten aus.
4. Erhöhen Sie für Desired capacity (Gewünschte Kapazität den aktuellen Wert um 1.
5. Wählen Sie Aktualisieren. Während eine Instance gestartet wird, zeigt die Spalte Status den Status Updating capacity (Kapazität aktualisieren) an.

Nachdem Sie die gewünschte Kapazität erhöht haben, können Sie überprüfen, ob die Instance erfolgreich gestartet wurde und nicht von der Beschreibung der Skalierungsaktivitäten beendet wurde.

Ansehen der Skalierungsaktivität

1. Wählen Sie auf der Seite Auto-Scaling-Gruppen Ihre Gruppe aus.
2. Auf der Registerkarte Activity (Aktivität) wird unter Activity history (Aktivitätsverlauf) in der Spalte Status angezeigt, ob Ihre Auto-Scaling-Gruppe Instances erfolgreich gestartet hat.
3. Wenn das Benutzerdatenskript fehlschlägt, sehen Sie nach Ablauf des Timeout-Zeitraums eine Skalierungsaktivität mit dem Status `Canceled` und eine Statusmeldung `Instance failed to complete user's Lifecycle Action: Lifecycle Action with token e85eb647-4fe0-4909-b341-a6c42EXAMPLE was abandoned: Lifecycle Action Completed with ABANDON Result`.

Schritt 6: Bereinigen

Wenn Sie mit den Ressourcen gearbeitet haben, die Sie für dieses Tutorial erstellt haben, führen Sie die folgenden Schritte aus, um sie zu löschen.

So löschen Sie den Lebenszyklus-Hook

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.
3. Wählen Sie auf der Registerkarte Instance management (Instance-Verwaltung) unter Lebenszyklus-Hooks den Lebenszyklus (`TestAutoScalingEvent-hook`) aus.
4. Wählen Sie Actions (Aktionen), Delete (Löschen) aus.
5. Um dies zu bestätigen, wählen Sie erneut Delete (Löschen) aus.

Löschen Sie die Startvorlage wie folgt:

1. Öffnen Sie die Seite Startvorlagen der Amazon EC2 Konsole.
2. Wählen Sie Ihre Startvorlage (`TestAutoScalingEvent-template`) und anschließend Actions (Aktionen) und Delete template (Vorlage löschen) aus.
3. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`Delete`** ein, um das Löschen der angegebenen Auto-Scaling-Gruppe zu bestätigen, wählen Sie dann Löschen.

Wenn Sie mit der Beispiel-Auto-Scaling-Gruppe fertig sind, löschen Sie sie. Sie können auch die von Ihnen erstellte IAM-Rolle und Berechtigungsrichtlinie löschen.

Löschen Sie die Auto-Scaling-Gruppe wie folgt:

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe (`TestAutoScalingEvent-group`) und wählen Sie Delete (Löschen) aus.
3. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`delete`** ein, um das Löschen der angegebenen Auto-Scaling-Gruppe zu löschen, wählen Sie dann Löschen.

   Ein Ladesymbol in der Spalte Name zeigt an, dass die Auto-Scaling-Gruppe gelöscht wird. Es dauert einige Minuten, bis die Instances beendet werden und die Gruppe gelöscht wird.

Löschen Sie die IAM-Rolle wie folgt:

1. Öffnen Sie die Seite Roles (Rollen) in der IAM-Konsole.
2. Wählen Sie die Rolle der Funktion (`TestAutoScalingEvent-role`).

3. Wählen Sie Löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den Namen der Rolle ein und wählen Sie dann Delete (Löschen).

Löschen der IAM-Richtlinie

1. Öffnen Sie die Seite Richtlinien in der IAM-Konsole.
2. Wählen Sie die Richtlinie aus, die Sie erstellt haben (`TestAutoScalingEvent-policy`).
3. Wählen Sie Aktionen, Löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den Namen der Richtlinie ein und wählen Sie dann Delete (Löschen).

Zugehörige Ressourcen

Die folgenden damit verbundenen Themen können hilfreich sein, wenn Sie Code entwickeln, der auf der Grundlage der in den Instance-Metadaten verfügbaren Daten Aktionen für Instances aufruft.

- Abrufen des Ziellebenszyklus-Status durch Instance-Metadaten. In diesem Abschnitt wird der Lebenszyklus-Status für andere Anwendungsfälle, wie z. B. die Beendigung der Instance, beschrieben.
- Lebenszyklus-Hooks hinzufügen (Konsole). Diese Prozedur zeigt Ihnen, wie Sie Lebenszyklus-Hooks sowohl für Aufskalieren (Start von Instances) als auch Abskalieren (Beendigung von Instances oder Rückkehr zu einem warmen Pool) hinzufügen können.
- Kategorien von Instance-Metadaten im Amazon EC2 EC2-Benutzerhandbuch. In diesem Thema sind alle Kategorien von Instance-Metadaten aufgeführt, mit denen Sie Aktionen auf EC2-Instances aufrufen können.

Ein Tutorial, das Ihnen zeigt, wie Sie Amazon verwenden, um Regeln EventBridge zu erstellen, die Lambda-Funktionen auf der Grundlage von Ereignissen aufrufen, die mit den Instances in Ihrer Auto Scaling Scaling-Gruppe passieren, finden Sie unter. Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft

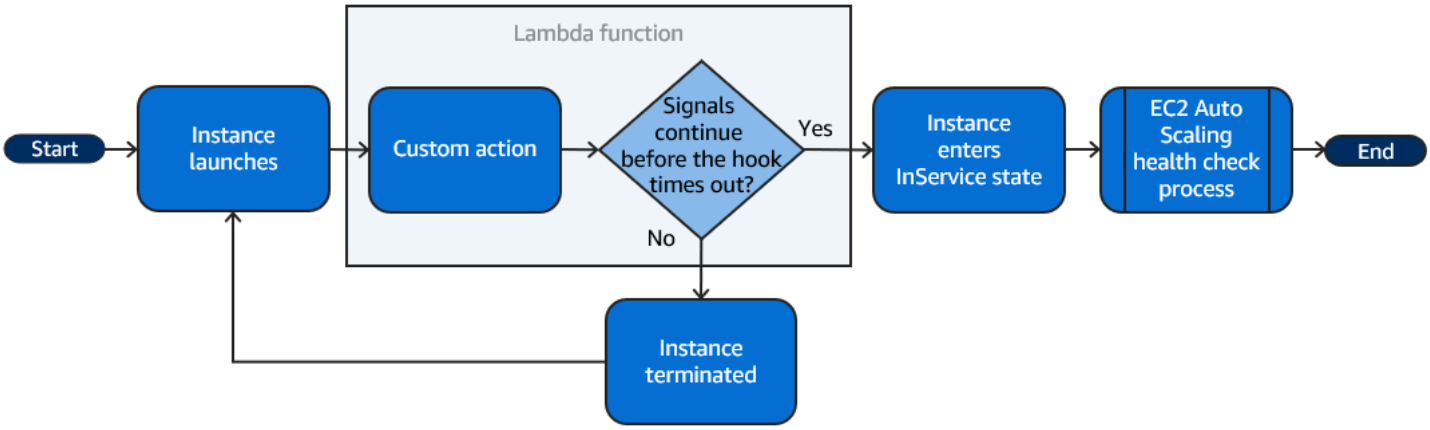
Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft

In dieser Übung erstellen Sie eine EventBridge Amazon-Regel, die ein Filtermuster enthält, das bei Übereinstimmung eine AWS Lambda Funktion als Regelziel aufruft. Wir stellen das Filtermuster und Beispielfunktionscode zur Verfügung.

Wenn alles richtig konfiguriert ist, führt die Lambda-Funktion am Ende dieses Tutorials beim Start von Instances eine benutzerdefinierte Aktion aus. Die benutzerdefinierte Aktion protokolliert einfach das Ereignis im CloudWatch Logs-Protokollstream, der der Lambda-Funktion zugeordnet ist.

Die Lambda-Funktion führt auch einen Rückruf durch, damit der Lebenszyklus der Instance fortgesetzt werden kann, wenn diese Aktion erfolgreich ist. Die Instance kann jedoch den Start abbrechen und beendet werden, wenn die Aktion fehlschlägt.

In der folgenden Abbildung wird der Ablauf für ein Scale-Out-Ereignis zusammengefasst, wenn Sie eine Lambda-Funktion verwenden, um eine benutzerdefinierte Aktion auszuführen. Nach dem Start einer Instance wird der Lebenszyklus der Instance angehalten, bis der Lifecycle-Hook abgeschlossen ist, entweder durch eine Zeitüberschreitung oder dadurch, dass Amazon EC2 Auto Scaling ein Signal zum Fortfahren empfängt.



Inhalt

Voraussetzungen

Erstellen Sie vor Beginn dieses Tutorials eine Auto-Scaling-Gruppe, falls noch keine vorhanden ist. Öffnen Sie zum Erstellen einer Auto-Scaling-Gruppe die Seite Auto-Scaling-Gruppen der Amazon EC2-Konsole und wählen Sie Eine Auto-Scaling-Gruppe erstellen aus.

Schritt 1: Erstellen einer IAM-Rolle mit Berechtigungen zum Abschließen von Lebenszyklus-Aktionen

Bevor Sie eine Lambda-Funktion erstellen, müssen Sie zunächst eine Ausführungsrolle und eine Berechtigungsrichtlinie erstellen, damit Lambda Lebenszyklus-Hooks abschließen kann.

So erstellen Sie die Richtlinie

1. Öffnen Sie in der IAM-Konsole Policies (Richtlinien) und wählen Sie dann Create policy (Richtlinie erstellen) aus.
2. Wählen Sie den Tab JSON.
3. Im Kästchen Policy Document fügen Sie das folgende Richtliniendokument in das Feld ein und ersetzen den Text in *Kursivschrift* mit Ihrer Kontonummer und dem Namen Ihrer Auto-Scaling-Gruppe.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction"
      ],
      "Resource":
 "arn:aws:autoscaling:*:123456789012:autoScalingGroup:*:autoScalingGroupName/my-
asg"
    }
  ]
```

```
}
```

4. Wählen Sie Weiter aus.
5. Geben Sie unter Policy name (Richtlinienname) **`LogAutoScalingEvent-policy`** ein. Wählen Sie Richtlinie erstellen aus.

Wenn Sie die Richtlinie fertig erstellt haben, können Sie eine Rolle erstellen, die sie verwendet.

So erstellen Sie die Rolle

1. Wählen Sie im Navigationsbereich auf der linken Seite Roles (Rollen).
2. Wählen Sie Rolle erstellen aus.
3. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die Option AWS - Dienst.
4. Wählen Sie für Ihren Anwendungsfall Lambda und dann Next (Weiter) aus.
5. Wählen Sie unter Berechtigungen hinzufügen die Richtlinie aus, die Sie erstellt haben (LogAutoScalingEvent-policy), und die benannte Richtlinie aus. AWSLambdaBasicExecutionRole Wählen Sie anschließend Weiter.

   > **Note**
   >
   > Die AWSLambdaBasicExecutionRoleRichtlinie verfügt über die Berechtigungen, die die Funktion benötigt, um Protokolle in Logs zu CloudWatch schreiben.

6. Geben Sie auf der Seite Set role name and review (Rollenname festlegen und überprüfen) für Role name (Rollenname) **`LogAutoScalingEvent-role`** ein und wählen Sie Create role (Rolle erstellen) aus.

Schritt 2: Erstellen einer Lambda-Funktion

Erstellen Sie eine Lambda-Funktion, die als Ziel für Ereignisse dienen soll. Die in Node.js geschriebene Lambda-Beispielfunktion wird aufgerufen, EventBridge wenn ein entsprechendes Ereignis von Amazon EC2 Auto Scaling ausgelöst wird.

Eine Lambda-Funktion erstellen

1. Öffnen Sie die Funktions-Seite in der Lambda-Konsole.

2. Wählen Sie Funktion erstellen und Von Grund auf neu erstellen aus.
3. Geben Sie unter Basic Information (Grundlegende Informationen) für Function name (Funktionsname) **`LogAutoScalingEvent`** ein.
4. Wählen Sie unter Laufzeit die Option Node.js 18.x aus.
5. Scrollen Sie nach unten und wählen Sie Ändern der standardmäßigen Ausführungsrolle und dann unter Ausführungsrolle Verwenden einer vorhandenen Rolle aus.
6. Wählen Sie für Existing role die Option -role aus. LogAuto ScalingEvent
7. Übernehmen Sie im Übrigen die Standardwerte.
8. Wählen Sie Funktion erstellen. Sie kehren zum Code und zur Konfiguration der Funktion zurück.
9. Fügen Sie bei geöffneter `LogAutoScalingEvent`-Funktion in der Konsole unter Code-Quelle im Editor den folgenden Beispielcode in die Datei index.mjs ein.

```
import { AutoScalingClient, CompleteLifecycleActionCommand } from "@aws-sdk/client-
auto-scaling";
export const handler = async(event) => {
  console.log('LogAutoScalingEvent');
  console.log('Received event:', JSON.stringify(event, null, 2));
  var autoscaling = new AutoScalingClient({ region: event.region });
  var eventDetail = event.detail;
  var params = {
    AutoScalingGroupName: eventDetail['AutoScalingGroupName'], /* required */
    LifecycleActionResult: 'CONTINUE', /* required */
    LifecycleHookName: eventDetail['LifecycleHookName'], /* required */
    InstanceId: eventDetail['EC2InstanceId'],
    LifecycleActionToken: eventDetail['LifecycleActionToken']
  };
  var response;
  const command = new CompleteLifecycleActionCommand(params);
  try {
    var data = await autoscaling.send(command);
    console.log(data); // successful response
    response = {
      statusCode: 200,
      body: JSON.stringify('SUCCESS'),
    };
  } catch (err) {
    console.log(err, err.stack); // an error occurred
    response = {
      statusCode: 500,
      body: JSON.stringify('ERROR'),
```

```
    };
  }
  return response;
};
```

Dieser Code protokolliert einfach das Ereignis, sodass Sie am Ende dieses Tutorials sehen können, dass ein Ereignis im CloudWatch Log-Log-Stream erscheint, das mit dieser Lambda-Funktion verknüpft ist.

10. Wählen Sie Bereitstellen.

Schritt 3: Erstellen Sie eine Regel EventBridge

Erstellen Sie eine EventBridge Regel, um Ihre Lambda-Funktion auszuführen. Weitere Hinweise zur Verwendung finden Sie EventBridge unterWird EventBridge zur Behandlung von Auto Scaling Scaling-Ereignissen verwendet.

So erstellen Sie eine Regel mithilfe der Konsole

1. Öffnen Sie die EventBridge-Konsole.
2. Wählen Sie im Navigationsbereich Rules aus.
3. Wählen Sie Regel erstellen aus.
4. Zum Define rule detail (Festlegen der Regeldetails) gehen Sie folgendermaßen vor:
   a. Geben Sie unter Name **`LogAutoScalingEvent-rule`** ein.
   b. Bei Event bus (Ereignisbus) wählen Sie default (Standard) aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis generiert, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.
   c. Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus.
   d. Wählen Sie Weiter aus.
5. Bei Build event pattern (Ereignis-Muster erstellen) gehen Sie wie folgt vor:
   a. Wählen Sie als Eventquelle AWS Events oder EventBridge Partnerevents aus.
   b. Scrollen Sie nach unten zu Ereignis-Muster und gehen Sie wie folgt vor:
   c. i. Wählen Sie für Ereignisquelle die Option AWS-Services aus.
      ii. Für AWS-Service, wählen Sie Auto Scaling aus.

iii. Wählen Sie in Event Type (Ereignistyp) die Option Instance Launch and Terminate (Starten und Beenden von Instances) aus.

iv. Standardmäßig entspricht die Regel jedem Abskalierungs- oder Aufskalierungs-Ereignis. Um eine Regel zu erstellen, die Sie benachrichtigt, wenn ein Aufskalierungs-Ereignis vorliegt und eine Instance aufgrund eines Lebenszyklus-Hook in einen Wartezustand versetzt wird, wählen Sie Specific instance event(s) (Bestimmte Instance-Ereignisse) und wählen Sie EC2 Instance-launch Lifecycle Action (Lebenszyklusaktion beim Start von EC2-Instances) aus.

v. Standardmäßig stimmt die Regel mit jeder Auto-Scaling-Gruppe in der Region überein. Damit die Regel mit einer bestimmten Auto-Scaling-Gruppe übereinstimmt, wählen Sie Specific group name(s) und wählen Sie dann die Gruppe aus.

vi. Wählen Sie Weiter aus.

6. Bei Select target(s) (Ziel(e) auswählen) gehen Sie wie folgt vor:

   a. Für Target types (Zieltypen), wählen Sie AWS-Service aus.

   b. Für Select a target (Ein Ziel auswählen), wählen die Option Lambda function (Lambda-Funktion) aus.

   c. Wählen Sie für Funktion die Option LogAutoScalingEvent.

   d. Klicken Sie zweimal auf Weiter.

7. Wählen Sie auf der Seite Überprüfen und erstellen die Option Regel erstellen aus.

Schritt 4: Hinzufügen eines Lebenszyklus-Hooks

In diesem Abschnitt fügen Sie einen Lebenszyklus-Hook hinzu, damit Lambda Ihre Funktion beim Start auf Instances ausführt.

So fügen Sie einen Lebenszyklus-Hook hinzu

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe. Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie im unteren Bereich auf der Registerkarte Instance management (Instance-Verwaltung) unter Lebenszyklus-Hooks die Option Create Lebenszyklus hook (Lebenszyklus-Hook erstellen) aus.

4. Gehen Sie wie folgt vor, um einen Lebenszyklus-Hook zum Aufskalieren (Start von Instances) zu definieren:

   a. Geben Sie für Lebenszyklus-Hooks den Wert **`LogAutoScalingEvent-hook`** ein.

   b. Wählen Sie bei Lifecycle Transition (Lebenszykluswechsel) die Option Instance launch (Instance-Start) aus.

   c. Für Heartbeat-Zeitüberschreitung geben Sie den Wert **`300`** für die Anzahl an Sekunden ein, um auf einen Rückruf von Ihrer Lambda-Funktion zu warten.

   d. Für Standardergebnis wählen Sie ABBRECHEN aus. Dies bedeutet, dass die Auto-Scaling-Gruppe eine neue Instance beendet, wenn die Zeitüberschreitung des Hook erreicht ist, ohne einen Rückruf von Ihrer Lambda-Funktion erhalten zu haben.

   e. (Optional) Lassen Sie Benachrichtigungs-Metadaten leer. Die Ereignisdaten, an die wir übergeben, EventBridge enthalten alle notwendigen Informationen, um die Lambda-Funktion aufzurufen.

5. Wählen Sie Erstellen.

Schritt 5: Testen und Prüfen des Ereignisses

Um das Ereignis zu testen, aktualisieren Sie die Auto-Scaling-Gruppe, indem Sie die gewünschte Kapazität der Auto-Scaling-Gruppe um 1 erhöhen. Ihre Lambda-Funktion wird innerhalb weniger Sekunden nach der Erhöhung der gewünschten Kapazität aufgerufen.

So erhöhen Sie die Größe der Auto-Scaling-Gruppe

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe, um Details in einem unteren Bereich anzuzeigen und weiterhin die oberen Zeilen des oberen Bereichs anzuzeigen.
3. Wählen Sie im unteren Bereich auf der Registerkarte Details die Option Gruppendetails, Bearbeiten aus.
4. Erhöhen Sie für Desired capacity (Gewünschte Kapazität den aktuellen Wert um 1.
5. Wählen Sie Aktualisieren. Während eine Instance gestartet wird, zeigt die Spalte Status den Status Updating capacity (Kapazität aktualisieren) an.

Nachdem Sie die gewünschte Kapazität erhöht haben, können Sie prüfen, ob die Lambda-Funktion aufgerufen wurde.

Anzeigen der Ausgabe aus der Lambda-Funktion

1. Öffnen Sie die Seite Protokollgruppen der CloudWatch Konsole.
2. Wählen Sie den Namen der Protokollgruppe für Ihre Lambda-Funktion aus (`/aws/lambda/LogAutoScalingEvent`).
3. Wählen Sie den Namen des Protokoll-Streams aus, um die von der Funktion für die Lebenszyklus-Aktion bereitgestellten Daten anzuzeigen.

Als Nächstes können Sie anhand der Beschreibung der Skalierungsaktivitäten prüfen, ob die Instance erfolgreich gestartet wurde.

Ansehen der Skalierungsaktivität

1. Wählen Sie auf der Seite Auto-Scaling-Gruppen Ihre Gruppe aus.
2. Auf der Registerkarte Activity (Aktivität) wird unter Activity history (Aktivitätsverlauf) in der Spalte Status angezeigt, ob Ihre Auto-Scaling-Gruppe Instances erfolgreich gestartet hat.
   - Wenn die Aktion erfolgreich war, hat die Skalierungsaktivität den Status „Erfolgreich“.
   - Wenn es fehlgeschlagen ist, sehen Sie nach einigen Minuten eine Skalierungsaktivität mit dem Status „Abgebrochen“ und die Statusmeldung „Instance konnte nicht abgeschlossen werden: Lebenszyklusaktion des Benutzers: Lebenszyklusaktion mit Token E85EB647-4FE0-4909-B341-A6C42Beispiel wurde abgebrochen: Lebenszyklusaktion mit ABBRUCH-Ergebnis abgeschlossen“.

So verkleinern Sie die Auto-Scaling-Gruppe

Wenn Sie die zusätzliche Instance, die Sie für diesen Test gestartet haben, nicht benötigen, können Sie die Registerkarte Details öffnen und Desired capacity (Gewünschte Kapazität) um 1 reduzieren.

Schritt 6: Bereinigen

Wenn Sie mit den Ressourcen gearbeitet haben, die Sie speziell für dieses Tutorial erstellt haben, führen Sie die folgenden Schritte aus, um sie zu löschen.

So löschen Sie den Lebenszyklus-Hook

1. Öffnen Sie die Seite Auto-Scaling-Gruppen in der Amazon-EC2-Konsole.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

3. Wählen Sie auf der Registerkarte Instance management (Instance-Verwaltung) unter Lebenszyklus-Hooks den Lebenszyklus (`LogAutoScalingEvent-hook`) aus.
4. Wählen Sie Actions (Aktionen), Delete (Löschen) aus.
5. Um dies zu bestätigen, wählen Sie erneut Delete (Löschen) aus.

Um die EventBridge Amazon-Regel zu löschen

1. Öffnen Sie die Seite Regeln in der EventBridge Amazon-Konsole.
2. Wählen Sie in Event bus (Ereignisbus) den Ereignisbus aus, der der Regel zugeordnet ist (`Default`).
3. Aktivieren Sie das Kontrollkästchen neben Ihrer Regel (`LogAutoScalingEvent-rule`).
4. Wählen Sie Löschen aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den Namen der Anwendung ein, und wählen Sie dann Delete aus.

Wenn Sie mit der Beispielfunktion fertig sind, löschen Sie sie. Sie können auch die Protokollgruppe löschen, welche die Protokolle der Funktion speichert, sowie die von Ihnen erstellte Ausführungsrolle und Berechtigungsrichtlinie.

So löschen Sie eine Lambda-Funktion

1. Öffnen Sie die Seite Funktionen der Lambda-Konsole.
2. Wählen Sie die Funktion (`LogAutoScalingEvent`) aus.
3. Wählen Sie Aktionen, Löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie**`delete`**um die angegebene Funktion zu löschen und wählen Sie dannLöschen.

So löschen Sie die Protokollgruppe

1. Öffnen Sie die Seite Protokollgruppen der CloudWatch Konsole.
2. Wählen Sie die Protokollgruppe der Funktion (`/aws/lambda/LogAutoScalingEvent`).
3. Wählen Sie Actions (Aktionen), Delete log group(s) (Protokollgruppe(n) löschen) aus.
4. Wählen Sie im Dialogfeld Delete log group(s) (Protokollgruppe(n) löschen) die Option Delete (Löschen) aus.

So löschen Sie die Ausführungsrolle

1. Öffnen Sie die Seite Roles (Rollen) in der IAM-Konsole.
2. Wählen Sie die Rolle der Funktion (`LogAutoScalingEvent-role`).
3. Wählen Sie Löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den Namen der Rolle ein und wählen Sie dann Delete (Löschen).

Löschen der IAM-Richtlinie

1. Öffnen Sie die Seite Richtlinien in der IAM-Konsole.
2. Wählen Sie die Richtlinie aus, die Sie erstellt haben (`LogAutoScalingEvent-policy`).
3. Wählen Sie Aktionen, Löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den Namen der Richtlinie ein und wählen Sie dann Delete (Löschen).

Zugehörige Ressourcen

Die folgenden verwandten Themen können hilfreich sein, wenn Sie EventBridge Regeln erstellen, die auf Ereignissen basieren, die den Instances in Ihrer Auto Scaling Scaling-Gruppe passieren.

- Wird EventBridge zur Behandlung von Auto Scaling Scaling-Ereignissen verwendet. In diesem Abschnitt finden Sie Beispiele für Ereignisse für andere Anwendungsfälle, einschließlich Ereignisse für die Skalierung.
- Lebenszyklus-Hooks hinzufügen (Konsole). Diese Prozedur zeigt Ihnen, wie Sie Lebenszyklus-Hooks sowohl für Aufskalieren (Start von Instances) als auch Abskalieren (Beendigung von Instances oder Rückkehr zu einem warmen Pool) hinzufügen können.

Ein Tutorial, das Ihnen zeigt, wie Sie den Instance Metadata Service (IMDS) verwenden, um eine Aktion innerhalb der Instance selbst aufzurufen, finden Sie unter Tutorial: Konfigurieren Sie Benutzerdaten zum Abrufen des Ziellebenszyklusstatus über Instance-Metadaten.

Warm-Pools für Amazon EC2 Auto Scaling

Ein Warm-Pool bietet Ihnen die Möglichkeit, die Latenz für Anwendungen mit außergewöhnlich langen Startzeiten zu verringern, z. B. weil Instances große Datenmengen auf die Festplatte

schreiben müssen. Mit Warm-Pools müssen Sie Ihre Auto-Scaling-Gruppen nicht mehr übermäßig bereitstellen, um die Latenz zu verwalten, damit die Anwendungsleistung verbessert werden kann. Weitere Informationen finden Sie im Blog-Beitrag Scaling your applications faster with EC2 Auto Scaling Warm Pools.

> **Important**
>
> Das Erstellen eines Warm-Pools, wenn dieser nicht erforderlich ist, kann zu unnötigen Kosten führen. Wenn Ihre erste Startzeit keine merklichen Latenzprobleme für Ihre Anwendung verursacht, benötigen Sie wahrscheinlich keinen Warm-Pool.

Themen

Schlüsselkonzepte

Bevor Sie beginnen, sollten Sie sich mit folgenden Kernkonzepten vertraut machen:

Warm Pool

Ein Warm Pool ist ein Pool vorinitialisierter EC2-Instances, der sich neben einer Auto-Scaling-Gruppe befindet. Jedes Mal, wenn Ihre Anwendung skaliert werden muss, kann die Auto-Scaling-Gruppe auf den Warm-Pool zurückgreifen, um die neue gewünschte Kapazität zu erfüllen. Er stellt sicher, dass Instances den Anwendungsdatenverkehr schnell bedienen können, wodurch die Reaktion auf eine Aufskalierung beschleunigt wird. Wenn Instances den Warm-Pool verlassen, zählen sie zur gewünschten Kapazität der Gruppe. Dies wird als Warmstart bezeichnet.

Während sich Instances im Warm Pool befinden, skalieren Ihre Skalierungsrichtlinien nur, wenn der Metrikwert von Instances, die sich im `InService`-Zustand befinden, größer ist als der hohe Alarmschwellenwert der Skalierungsrichtlinie (welcher der Zielauslastung einer Skalierungsrichtlinie für die Zielverfolgung entspricht).

Warm-Pool-Größe

Die Größe des Warm Pool wird standardmäßig als Differenz zwischen zwei Zahlen berechnet: die maximale Kapazität der Auto-Scaling-Gruppe und die gewünschte Kapazität. Wenn beispielsweise die gewünschte Kapazität Ihrer Auto-Scaling-Gruppe sechs ist und die maximale Kapazität zehn beträgt, beträgt die Größe Ihres Warm-Pools vier, wenn Sie den Warm-Pool zum ersten Mal einrichten und der Pool initialisiert wird.

Um die maximale Kapazität des warmen Pools separat anzugeben, verwenden Sie die Option benutzerdefinierte Spezifikation (`MaxGroupPreparedCapacity`) und legen Sie dafür einen benutzerdefinierten Wert fest, der größer als die aktuelle Kapazität der Gruppe ist. Wenn Sie einen benutzerdefinierten Wert angeben, wird die Größe des warmen Pools als Differenz zwischen dem benutzerdefinierten Wert und der aktuell gewünschten Kapazität der Gruppe berechnet. Wenn die gewünschte Kapazität Ihrer Auto Scaling Scaling-Gruppe beispielsweise 6 ist, wenn die maximale Kapazität 20 ist und wenn der benutzerdefinierte Wert 8 ist, wird die Größe Ihres warmen Pools 2 sein, wenn Sie den warmen Pool zum ersten Mal einrichten und der Pool initialisiert wird.

Möglicherweise müssen Sie die Option benutzerdefinierte Spezifikation (`MaxGroupPreparedCapacity`) nur verwenden, wenn Sie mit großen Auto Scaling Scaling-Gruppen arbeiten, um die Kostenvorteile eines warmen Pools zu nutzen. So könnte zum Beispiel eine Auto-Scaling-Gruppe mit 1 000 Instances, einer maximalen Kapazität von 1 500 (um zusätzliche Kapazität für Notfall-Datenverkehrsspitzen bereitzustellen) und einem Warm Pool mit 100 Instances Ihre Ziele besser erreichen als mit einem Warm Pool mit 500 Instances.

Mindestanzahl des Warm-Pools

Erwägen Sie, die Mindestanzahl der Instances, die im Warm-Pool verwaltet werden sollen, standardmäßig festzulegen. Standardmäßig ist keine Mindestgröße festgelegt.

Status der Warm-Pool-Instance

Sie können Instances im Warm Pool in einem von drei Status belassen: `Stopped`, `Running` oder `Hibernated`. Wenn Instances in einem `Stopped`-Zustand gelassen werden, können Kosten minimiert werden. Bei gestoppten Instances zahlen Sie nur für die Volumes, die Sie verwenden, und die Elastic-IP-Adressen, die den Instances angefügt sind.

Alternativ können Sie Instances auch in einem `Hibernated`-Status belassen, um Instances zu stoppen, ohne ihren Speicherinhalt (RAM) zu löschen. Wenn eine Instance in den Ruhezustand versetzt wird, signalisiert dies dem Betriebssystem, den Inhalt Ihres Arbeitsspeichers auf Ihrem Amazon-EBS-Root-Volume zu speichern. Wenn die Instance wieder gestartet wird, wird das Stamm-Volume im vorherigen Status wiederhergestellt und der RAM-Inhalt wird neu geladen. Während sich die Instances im Ruhezustand befinden, zahlen Sie nur für die EBS-Volumes, einschließlich des Speichers für die RAM-Inhalte und die mit den Instances verbundenen Elastic-IP-Adressen.

Instances in einem `Running` Zustand innerhalb des Warm-Pools zu halten, ist ebenfalls möglich, wird aber dringend abgeraten, um unnötige Gebühren zu vermeiden. Wenn Instances gestoppt oder in den Ruhezustand versetzt werden, sparen Sie die Kosten für die Instances selbst. Sie zahlen für die Instances nur, wenn sie ausgeführt werden.

Lebenszyklus-Hooks

Mit Lebenszyklus-Hooks können Sie Instances in einen Wartestatus versetzen, damit Sie benutzerdefinierte Aktionen für die Instances ausführen können. Benutzerdefinierte Aktionen werden beim Start der Instances oder vor dem Beenden ausgeführt.

In einer Warm-Pool-Konfiguration verzögern Lebenszyklus-Hooks, dass Instances gestoppt oder in den Ruhezustand versetzt werden und während des Aufskalierens in Betrieb genommen werden, bis sie die Initialisierung abgeschlossen haben. Wenn Sie Ihrer Auto-Scaling-Gruppe einen Warm Pool ohne Lebenszyklus-Hook hinzufügen, können Instances, deren Initialisierung lange dauert, gestoppt oder in den Ruhezustand versetzt und dann während der Aufskalierung in Betrieb genommen werden, bevor sie fertig sind.

Richtlinie für die Instance-Wiederverwendung

Standardmäßig beendet Amazon EC2 Auto Scaling Ihre Instances, wenn Ihre Auto-Scaling-Gruppe abskaliert. Dann startet die Lösung neue Instances im Warm Pool, um die beendeten Instances zu ersetzen.

Wenn Sie stattdessen Instances an den Warm Pool zurückgeben möchten, können Sie eine Richtlinie für die Instance-Wiederverwendung angeben. Auf diese Weise können Sie Instances wiederverwenden, die bereits für die Bereitstellung des Anwendungsdatenverkehrs konfiguriert sind. Um sicherzustellen, dass Ihr Warm Pool nicht überdimensioniert wird, kann Amazon EC2 Auto Scaling Instances im Warm Pool beenden, um seine Größe zu reduzieren, wenn sie die ursprünglichen Einstellungen übersteigt. Wenn Instances im Warm Pool beendet werden, wird

die Standardbeendigungsrichtlinie verwendet, um auszuwählen, welche Instances zuerst beendet werden sollen.

> **Important**
>
> Wenn Sie Instances beim Abskalieren in den Ruhezustand versetzen möchten und Instances in der Auto-Scaling-Gruppe vorhanden sind, müssen diese die Anforderungen für den Instance-Ruhezustand erfüllen. Wenn dies nicht der Fall ist, werden Instances gestoppt, und nicht in den Ruhezustand versetzt, wenn sie in den Warm Pool zurückkehren.

> **Note**
>
> Derzeit können Sie eine Richtlinie für die Instance-Wiederverwendung nur mithilfe der AWS CLI oder eines SDK angeben. Diese Funktion ist in der Konsole nicht verfügbar.

Voraussetzungen

Bevor Sie einen warmen Pool für Ihre Auto-Scaling-Gruppe erstellen, entscheiden Sie, wie Sie Lebenszyklus-Hooks verwenden, um neue Instances mit einem geeigneten Anfangszustand zu initialisieren.

Um benutzerdefinierte Aktionen für Instances auszuführen, während sich diese aufgrund eines Lebenszyklus-Hooks im Wartestatus befinden, haben Sie zwei Möglichkeiten:

- Für einfache Szenarien, in denen Sie beim Start Befehle für Ihre Instances ausführen möchten, können Sie ein Benutzerdatenskript einbeziehen, wenn Sie eine Startvorlage erstellen oder eine Konfiguration für Ihre Auto-Scaling-Gruppe starten. Benutzerdatenskripte sind nur normale Shell-Skripte oder cloud-init-Anweisungen, die von cloud-init ausgeführt werden, wenn Ihre Instances beginnen. Das Skript kann auch steuern, wann Ihre Instances in den nächsten Status übergehen, indem Sie die ID der Instance verwenden, auf der sie ausgeführt wird. Wenn Sie nicht bereits dabei sind, aktualisieren Sie das Skript, sodass es die Instance-ID der Instance aus den Instance-Metadaten abruft. Weitere Informationen finden Sie unter Instance-Metadaten abrufen im Amazon EC2 EC2-Benutzerhandbuch.

> **Tip**
>
> Um Benutzerdatenskripte beim Neustart einer Instance auszuführen, müssen die Benutzerdaten im mehrteiligen MIME-Format vorliegen und Folgendes im Abschnitt `#cloud-config` der Benutzerdaten angeben:
>
> ```
> #cloud-config
> cloud_final_modules:
>  - [scripts-user, always]
> ```

- Für fortgeschrittene Szenarien, in denen Sie einen Dienst benötigen, AWS Lambda um beispielsweise etwas zu tun, wenn Instances den warmen Pool betreten oder verlassen, können Sie einen Lifecycle-Hook für Ihre Auto Scaling Scaling-Gruppe erstellen und den Zieldienst so konfigurieren, dass er benutzerdefinierte Aktionen auf der Grundlage von Lebenszyklusbenachrichtigungen ausführt. Weitere Informationen finden Sie unter Unterstützte Benachrichtigungsziele.

Instances auf Ruhezustand vorbereiten

Um Auto Scaling Scaling-Instances für die Verwendung des `Hibernated` Pool-Status vorzubereiten, erstellen Sie eine neue Startvorlage oder Startkonfiguration, die korrekt eingerichtet ist, um den Instance-Ruhezustand zu unterstützen, wie im Thema Voraussetzungen für den Ruhezustand im Amazon EC2 EC2-Benutzerhandbuch beschrieben. Ordnen Sie dann die neue Startvorlage oder Startkonfiguration der Auto-Scaling-Gruppe zu und starten Sie eine Instance-Aktualisierung, um die Instances zu ersetzen, die einer vorherigen Startvorlage oder Startkonfiguration zugeordnet sind. Weitere Informationen finden Sie unter Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren.

Aktualisieren der Instances in einem warmen Pool

Um die Instances in einem warmen Pool zu aktualisieren, erstellen Sie eine neue Startvorlage oder Startkonfiguration und verknüpfen sie mit der Auto-Scaling-Gruppe. Alle neuen Instances werden mithilfe des neuen AMI und anderer Updates gestartet, die in der Startvorlage oder Startkonfiguration angegeben sind, bestehende Instances sind jedoch nicht davon betroffen.

Um das Starten von warmen Ersatz-Pool-Instances zu erzwingen, die die neue Startvorlage oder Startkonfiguration verwenden, können Sie eine Instance-Aktualisierung starten, um eine fortlaufende

Aktualisierung Ihrer Gruppe durchzuführen. Eine Instance-Aktualisierung ersetzt zuerst `InService`-Instances. Dann ersetzt sie Instances im Warm-Pool. Weitere Informationen finden Sie unter Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren.

Zugehörige Ressourcen

In unserem GitHubRepository finden Sie Beispiele für Lifecycle-Hooks für warme Pools.

Einschränkungen

- Sie können einer Auto Scaling Scaling-Gruppe, die über eine Richtlinie für gemischte Instanzen verfügt, keinen warmen Pool hinzufügen. Sie können auch keinen warmen Pool zu einer Auto Scaling Scaling-Gruppe hinzufügen, die über eine Startvorlage oder eine Startkonfiguration verfügt, die Spot-Instances anfordert.
- Amazon EC2 Auto Scaling kann eine Instance nur dann in einen `Stopped`– oder `Hibernated`-Status verschieben, wenn es ein Amazon-EBS-Volume als Stammgerät hat. Instances, die Instance-Speicher als Stammgerät verwenden, können nicht beendet oder in den Ruhezustand versetzt werden.
- Amazon EC2 Auto Scaling kann eine Instance nur dann in einen `Hibernated` Zustand versetzen, wenn sie alle im Thema Voraussetzungen für den Ruhezustand im Amazon EC2 EC2-Benutzerhandbuch aufgeführten Anforderungen erfüllt.
- Wenn Ihr Warm-Pool bei einem Scale-Out-Ereignis erschöpft ist, werden Instances direkt in der Auto-Scaling-Gruppe (ein Kaltstart) starten. Es kann auch zu einem Kaltstart kommen, wenn eine Availability Zone nicht genügend Kapazität hat.
- Wenn eine Instance innerhalb des Warmpools während des Startvorgangs auf ein Problem stößt, das sie daran hindert, den `InService` Status zu erreichen, wird die Instance als fehlgeschlagener Start betrachtet und beendet. Dies gilt unabhängig von der zugrunde liegenden Ursache, z. B. einem Fehler bei unzureichender Kapazität oder einem anderen Faktor.
- Wenn Sie versuchen, einen Warm-Pool mit einer von Amazon Elastic Kubernetes Service (Amazon EKS) verwalteten Knotengruppe zu verwenden, registrieren sich Instances, die noch initialisiert werden, möglicherweise bei Ihrem Amazon-EKS-Cluster. Infolgedessen kann der Cluster Jobs für eine Instance planen, da er sich darauf vorbereitet, gestoppt oder in den Ruhezustand versetzt zu werden.
- Wenn Sie versuchen, einen Warm-Pool mit einem Amazon ECS-Cluster zu verwenden, registrieren sich die Instances eventuell beim Cluster, bevor sie ihre Initialisierung abgeschlossen haben. Um

dieses Problem zu lösen, müssen Sie eine Startvorlage oder Startkonfiguration konfigurieren, die eine spezielle Agentenkonfigurationsvariable in den Benutzerdaten enthält. Weitere Informationen finden Sie unter Verwendung eines Warm-Pools für Ihre Auto-Scaling-Gruppe im Amazon Elastic Container Service-Entwicklerhandbuch.

- Hibernation-Unterstützung für warme Pools ist in allen kommerziellen Bereichen verfügbar, in AWS-Regionen denen Amazon EC2 Auto Scaling und Hibernation verfügbar sind, mit Ausnahme der folgenden:
  - Asien-Pazifik (Hyderabad)
  - Asien-Pazifik (Melbourne)
  - Kanada West (Calgary)
  - Region China (Peking)
  - Region China (Ningxia)
  - Europa (Spain)
  - Israel (Tel Aviv)

Verwenden von Lebenszyklus-Hooks mit einem Warm Pool

Instances in einem Warm Pool verwalten ihren eigenen, unabhängigen Lebenszyklus, um Ihnen bei der Erstellung der entsprechenden benutzerdefinierten Aktion für jeden Übergang zu helfen. Dieser Lebenszyklus soll Ihnen helfen, Aktionen in einem Zielservice (z. B. einer Lambda-Funktion) aufzurufen, während eine Instance noch initialisiert wird und bevor sie in Betrieb genommen wird.

> **Note**
>
> Die API-Vorgänge, die Sie zum Hinzufügen und Verwalten von Lebenszyklus-Hooks und zum Abschließen von Lebenszyklusaktionen verwenden, werden nicht geändert. Nur der Instance-Lebenszyklus wird geändert.

Weitere Informationen über das Hinzufügen von Lebenszyklus-Hooks finden Sie unter Lebenszyklus-Hooks hinzufügen. Weitere Informationen über das Abschließen einer Lebenszyklus-Aktion finden Sie unter Eine Lebenszyklus-Aktion abschließen.

Für Instances, die in den Warm Pool aufgenommen werden, benötigen Sie möglicherweise aus einem der folgenden Gründe einen Lebenszyklus-Hook:

- Sie möchten EC2-Instances von einem AMI aus starten, dessen Initialisierung lange dauert.
- Sie möchten Benutzerdatenskripte ausführen, um die EC2-Instances zu laden.

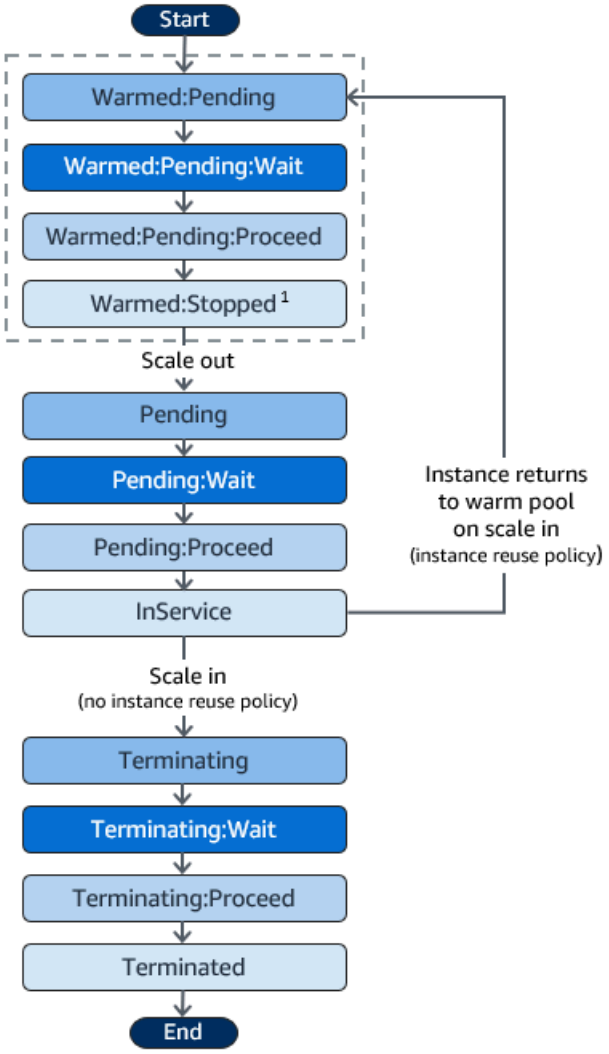
Für Instances, die den Warm Pool verlassen, benötigen Sie möglicherweise aus einem der folgenden Gründe einen Lebenszyklus-Hook:

- Sie können etwas mehr Zeit gebrauchen, um EC2-Instances auf die Verwendung vorzubereiten. Sie haben möglicherweise Services, die gestartet werden müssen, wenn eine Instance neu gestartet wird, bevor Ihre Anwendung ordnungsgemäß funktionieren kann.
- Sie möchten Cache-Daten vorab ausfüllen, um sicherzustellen, dass ein neuer Server nicht mit einem leeren Cache gestartet wird.
- Sie möchten neue Instances als verwaltete Instances mit Ihrem Konfigurationsverwaltungsservice registrieren.

Lebenszyklusstatusübergänge für Instances in einem Warm Pool

Eine Instance mit automatischer Skalierung kann im Verlauf ihres Lebenszyklus in verschiedene Status übergehen.

Im folgenden Diagramm wird der Übergang zwischen Status für automatische Skalierung veranschaulicht, wenn Sie einen Warm Pool verwenden:



[1] Dieser Status variiert je nach Einstellung des Pool-Status des Warm Pools. Wenn der Pool-Status auf `Running` gesetzt ist, ist dieser Status stattdessen `Warmed:Running`. Wenn der Pool-Status auf `Hibernated` gesetzt ist, ist dieser Status stattdessen `Warmed:Hibernated`.

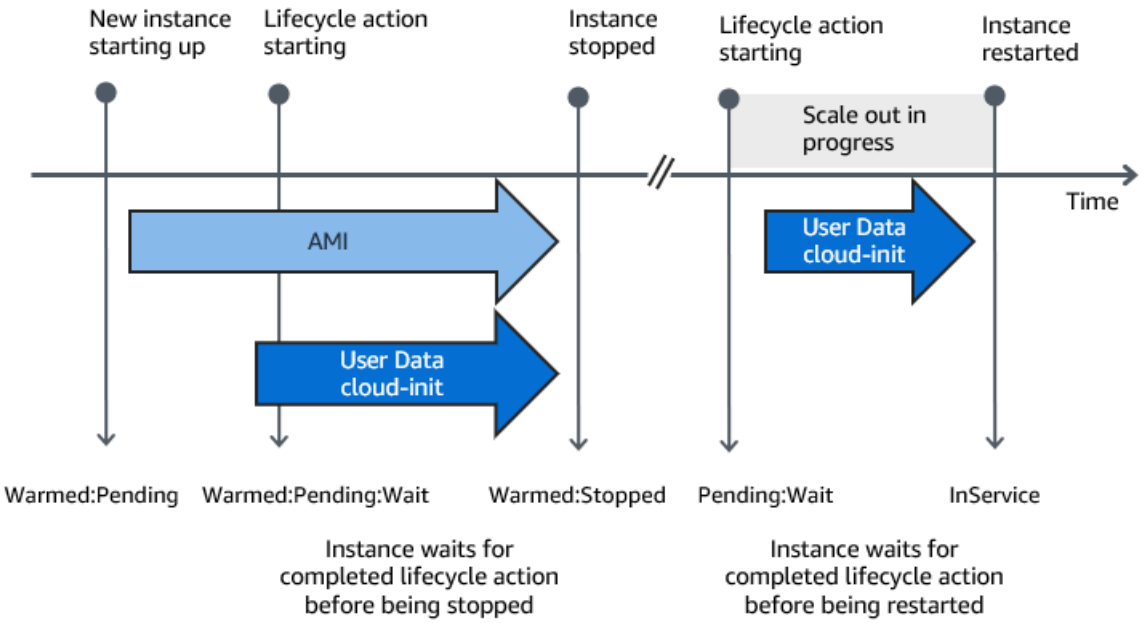
Berücksichtigen Sie beim Hinzufügen von Lebenszyklus-Hooks Folgendes:

- Wenn ein Lebenszyklus-Hook für die `autoscaling:EC2_INSTANCE_LAUNCHING`-Lebenszyklusaktion konfiguriert ist, wird eine neu gestartete Instance zunächst angehalten, um eine benutzerdefinierte Aktion auszuführen, wenn sie den `Warmed:Pending:Wait`-Zustand erreicht, und dann erneut, wenn die Instance neu gestartet wird und den `Pending:Wait`-Zustand erreicht.
- Wenn ein Lebenszyklus-Hook für die `EC2_INSTANCE_TERMINATING`-Lebenszyklusaktion konfiguriert ist, wird eine beendende Instance angehalten, um eine benutzerdefinierte Aktion auszuführen, wenn sie den `Terminating:Wait`-Zustand erreicht. Wenn Sie jedoch eine

Richtlinie für die Wiederverwendung von Instances so festlegen, dass Instances bei der Abwärtsskalierung in den Warm-Pool zurückgeführt werden, anstatt sie zu beenden, dann wird eine Instanz, die in den Warm-Pool zurückkehrt, angehalten, um eine benutzerdefinierte Aktion im `Warmed:Pending:Wait`-Zustand für die `EC2_INSTANCE_TERMINATING`-Lebenszyklusaktion durchzuführen.

- Wenn die Nachfrage nach Ihrer Anwendung den Warm Pool leert, kann Amazon EC2 Auto Scaling Instances direkt in der Auto-Scaling-Gruppe starten, sofern die Gruppe ihre maximale Kapazität noch nicht erreicht hat. Wenn die Instances direkt in der Gruppe gestartet werden, werden sie nur angehalten, um eine benutzerdefinierte Aktion im `Pending:Wait`-Zustand auszuführen.
- Um zu steuern, wie lange eine Instance in einem Wartezustand verbleibt, bevor sie in den nächsten Status übergeht, konfigurieren Sie Ihre benutzerdefinierte Aktion so, dass sie den complete-lifecycle-action-Befehl verwendet. Mit Lebenszyklus-Hooks bleiben Instances in einem Wartezustand, bis Sie Amazon EC2 Auto Scaling benachrichtigen, dass die angegebene Lebenszyklusaktion abgeschlossen ist, oder bis eine Zeitüberschreitung auftritt (standardmäßig eine Stunde).

Im Folgenden wird der Ablauf für ein Aufskalierungsereignis zusammengefasst.



Wenn Instances einen Wartezustand erreichen, sendet Amazon EC2 Auto Scaling eine Benachrichtigung. Beispiele für diese Benachrichtigungen finden Sie im EventBridge Abschnitt dieses Handbuchs. Weitere Informationen finden Sie unter Beispielereignisse und -muster in einem warmen Pool.

Unterstützte Benachrichtigungsziele

Amazon EC2 Auto Scaling bietet Unterstützung bei der Definition einer der folgenden Optionen als Benachrichtigungsziele für Lebenszyklusbenachrichtigungen:

- EventBridge Regeln
- Amazon SNS-Themen
- Amazon SQS-Warteschlangen

> **Important**
>
> Denken Sie daran: Wenn Sie ein Benutzerdatenskript in Ihrer Startvorlage oder Startkonfiguration haben, das Ihre Instances beim Start konfiguriert, müssen Sie keine Benachrichtigungen erhalten, um benutzerdefinierte Aktionen für Instances auszuführen, die gestartet oder neu gestartet werden.

Die folgenden Abschnitte enthalten Links zur Dokumentation, in der beschrieben wird, wie Benachrichtigungsziele konfiguriert werden:

EventBridge Regeln: Um Code auszuführen, wenn Amazon EC2 Auto Scaling eine Instance in einen Wartestatus versetzt, können Sie eine EventBridge Regel erstellen und eine Lambda-Funktion als Ziel angeben. Um verschiedene Lambda-Funktionen basierend auf verschiedenen Lebenszyklusbenachrichtigungen aufzurufen, können Sie mehrere Regeln erstellen und jede Regel einem bestimmten Ereignismuster und einer Lambda-Funktion zuordnen. Weitere Informationen finden Sie unter Erstellen Sie EventBridge Regeln für Ereignisse im warmen Pool.

Amazon-SNS-Themen: Um eine Benachrichtigung zu erhalten, wenn eine Instance in einen Wartestatus versetzt wird, erstellen Sie ein Amazon-SNS-Thema und richten Sie dann die Amazon-SNS-Nachrichtenfilterung ein, um unterschiedliche Lebenszyklusbenachrichtigungen basierend auf dem Nachrichtenattribut zu liefern. Weitere Informationen finden Sie unter Benachrichtigungen über Amazon SNS erhalten.

Amazon-SQS-Warteschlangen: Um einen Bereitstellungspunkt für Lebenszyklusbenachrichtigungen einzurichten, an dem ein relevanter Verbraucher sie abholen und verarbeiten kann, können Sie eine Amazon-SQS-Warteschlange und einen Warteschlangenverbraucher erstellen, der Nachrichten aus der SQS-Warteschlange verarbeitet. Wenn der Warteschlangenverbraucher

Lebenszyklusbenachrichtigungen basierend auf einem Nachrichtenattribut unterschiedlich verarbeiten soll, müssen Sie auch den Warteschlangenverbraucher so einrichten, dass er die Nachricht analysiert und dann auf die Nachricht reagiert, wenn ein bestimmtes Attribut dem gewünschten Wert entspricht. Weitere Informationen finden Sie unter Benachrichtigungen über Amazon SQS erhalten.

Erstellen eines warmen Pools für eine Auto-Scaling-Gruppe

In diesem Thema wird beschrieben, wie Sie einen warmen Pool für Ihre Auto-Scaling-Gruppe erstellen.

> **Important**
>
> Bevor Sie fortfahren, sollten Sie die Voraussetzungen für die Erstellung eines warmen Pools erfüllen und bestätigen, dass Sie einen Lebenszyklus-Hook für Ihre Auto-Scaling-Gruppe erstellt haben.

Erstellen eines Warm Pool

Führen Sie die folgenden Schritte aus, um einen warmen Pool für Ihre Auto-Scaling-Gruppe zu erstellen.

So erstellen Sie einen Warm Pool (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben einer vorhandenen Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie die Registerkarte Instance-Management aus.
4. Unter Warm-Pool wählen Sie Erstellen eines Warm-Pools aus.
5. Um einen Warm-Pool zu konfigurieren, gehen Sie folgendermaßen vor:

   a. Für Warm-Pool-Instancestatus wählen Sie aus, in welchen Zustand Ihre Instances wechseln sollen, wenn sie in den Warm-Pool gelangen. Der Standardwert ist `Stopped`.

   b. Für Mindesgröße des Warm-Pools geben Sie die Mindestanzahl der Instances ein, die im Warm-Pool verwaltet werden sollen.

c. Aktivieren Sie für die Wiederverwendung von Instances das Kontrollkästchen Skaliert wiederverwenden, damit Instances in der Auto Scaling-Gruppe in den warmen Pool zurückkehren können.

d. Wählen Sie für Größe des warmen Pools eine der verfügbaren Optionen aus:

- Standardspezifikation: Die Größe des warmen Pools wird durch die Differenz zwischen der maximalen und der gewünschten Kapazität der Auto Scaling Scaling-Gruppe bestimmt. Diese Option optimiert die Verwaltung von warmen Pools. Nachdem Sie den warmen Pool erstellt haben, kann seine Größe einfach aktualisiert werden, indem Sie einfach die maximale Kapazität der Gruppe anpassen.
- Benutzerdefinierte Spezifikation: Die Größe des warmen Pools wird durch die Differenz zwischen einem benutzerdefinierten Wert und der gewünschten Kapazität der Auto Scaling Scaling-Gruppe bestimmt. Diese Option bietet Ihnen die Flexibilität, die Größe Ihres warmen Pools unabhängig von der maximalen Kapazität der Gruppe zu verwalten.

6. Sehen Sie sich den Abschnitt Geschätzte Größe des warmen Pools auf der Grundlage der aktuellen Einstellungen an, um zu überprüfen, ob die Standard- oder benutzerdefinierte Spezifikation für die Größe des Warmwasserbeckens gilt. Denken Sie daran, dass die Größe des warmen Pools von der gewünschten Kapazität der Auto Scaling Scaling-Gruppe abhängt, die sich ändert, wenn die Gruppe skaliert wird.
7. Wählen Sie Erstellen.

Löschen eines Warm-Pools

Wenn Sie eine DHCP-Optionsliste nicht mehr benötigen, können Sie sie mit dem folgenden Verfahren löschen.

So löschen Sie Ihren Warm Pool (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben einer vorhandenen Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie die Registerkarte Instance-Management aus.
4. Wählen Sie für Warm pool (Warm Pool) Actions (Aktionen), Delete (Löschen) aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).

Anzeigen des Status der Zustandsprüfung und dem Grund für Zustandsprüfungsfehler

Mithilfe von Zustandsprüfungen kann Amazon EC2 Auto Scaling feststellen, wann eine Instance fehlerhaft ist und beendet werden sollte. Für Warm-Pool-Instances, die in einem `Stopped`-Zustand belassen werden, verwendet es die Kenntnisse, die Amazon EBS über eine `Stopped`-Instance hat, um fehlerhafte Instances zu identifizieren. Dies geschieht durch Aufrufen der `DescribeVolumeStatus`-API, um den Zustand des EBS-Volume zu ermitteln, das an die Instance angefügt ist. Für Warm-Pool-Instances, die in einem `Running`-Zustand belassen werden, stützt es sich auf EC2-Zustandsprüfungen, um den Instance-Zustand zu ermitteln. Zwar gibt es keine Zustandsprüfungsfrist für Warm-Pool-Instances, Amazon EC2 Auto Scaling startet die Prüfung der Instance-Integrität aber erst, wenn der Lebenszyklus-Hook abgeschlossen ist.

Wenn eine Instance als fehlerhaft erachtet wird, löscht Amazon EC2 Auto Scaling die fehlerhafte Instance automatisch und erstellt eine neue Instance, um sie zu ersetzen. Instances werden normalerweise innerhalb weniger Minuten nach erfolgter Zustandsprüfung beendet. Weitere Informationen finden Sie unter Anzeigen des Grundes für Fehler bei Zustandsprüfung.

Benutzerdefinierte Zustandsprüfungen werden ebenfalls unterstützt. Dies kann hilfreich sein, wenn Sie über ein eigenes Zustandsprüfsystem verfügen, das den Zustand einer Instance erkennt und diese Informationen an Amazon EC2 Auto Scaling sendet. Weitere Informationen finden Sie unter Benutzerdefinierte Zustandsprüfungen.

In der Amazon EC2 Auto Scaling-Konsole können Sie den Zustand (fehlerfrei oder fehlerhaft) Ihrer Warm-Pool-Instances einsehen. Sie können ihren Gesundheitszustand auch mithilfe des AWS CLI oder eines der SDKs einsehen.

So zeigen Sie den Zustand Ihrer Warm-Pool-Instances an (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet.
3. Auf der Registerkarte Instance management (Instance-Verwaltung) wird unter Warm pool instances (Warm-Pool-Instances) in der Spalte Lifecycle (Lebenszyklus) der Zustand Ihrer Instances angezeigt.

Die Spalte Zustand zeigt die Bewertung an, die Amazon EC2 Auto Scaling über den Instance-Zustand gemacht hat.

> **Note**
>
> Neue Instances beginnen fehlerfrei. Bis der Lebenszyklus-Hook abgeschlossen ist, wird die Integrität einer Instance nicht überprüft.

So zeigen Sie den Grund für den Ausfall einer Zustandsprüfung an (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet.
3. Auf der Registerkarte Activity (Aktivität) wird unter Activity history (Aktivitätsverlauf) in der Spalte Status angezeigt, ob Ihre Auto-Scaling-Gruppe Instances erfolgreich gestartet oder beendet hat.

   Wenn es Instances fehlerhaft beendet hat, zeigt die Spalte Ursache das Datum und die Uhrzeit der Beendigung und den Grund für den Fehler der Zustandsprüfung an. Beispiel: „Bei 2021-04-01T21:48:35Z wurde eine Instance aufgrund eines Fehlers der EBS-Volume-Zustandsprüfung außer Betrieb gesetzt“.

Anzeigen des Status Ihrer Warm-Pool-Instances (AWS CLI)

Anzeigen des Warm-Pools für eine Auto-Scaling-Gruppe, indem Sie den Befehl describe-warm-pool verwenden.

```
aws autoscaling describe-warm-pool --auto-scaling-group-name my-asg
```

Beispielausgabe.

```
{
    "WarmPoolConfiguration": {
        "MinSize": 0,
        "PoolState": "Stopped"
```

```
    },
    "Instances": [
        {
            "InstanceId": "i-0b5e5e7521cfaa46c",
            "InstanceType": "t2.micro",
            "AvailabilityZone": "us-west-2a",
            "LifecycleState": "Warmed:Stopped",
            "HealthStatus": "Healthy",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-08c4cd42f320d5dcd",
                "LaunchTemplateName": "my-template-for-auto-scaling",
                "Version": "1"
            }
        },
        {
            "InstanceId": "i-0e21af9dcfb7aa6bf",
            "InstanceType": "t2.micro",
            "AvailabilityZone": "us-west-2a",
            "LifecycleState": "Warmed:Stopped",
            "HealthStatus": "Healthy",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-08c4cd42f320d5dcd",
                "LaunchTemplateName": "my-template-for-auto-scaling",
                "Version": "1"
            }
        }
    ]
}
```

So zeigen Sie den Grund für den Ausfall einer Zustandsprüfung an (AWS CLI)

Verwenden Sie den folgenden describe-scaling-activities-Befehl.

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

Nachfolgend finden Sie eine Beispielantwort, wobei `Description` angibt, dass Ihre Auto-Scaling-Gruppe eine Instance beendet hat, und`Cause` den Grund für den Fehler bei der Zustandsprüfung angibt.

Skalierungsaktivitäten werden nach Startzeit sortiert. Die noch laufenden Aktivitäten werden zuerst beschrieben.

```
{
```

```
  "Activities": [
    {
      "ActivityId": "4c65e23d-a35a-4e7d-b6e4-2eaa8753dc12",
      "AutoScalingGroupName": "my-asg",
      "Description": "Terminating EC2 instance: i-04925c838b6438f14",
      "Cause": "At 2021-04-01T21:48:35Z an instance was taken out of service in
 response to EBS volume health check failure.",
      "StartTime": "2021-04-01T21:48:35.859Z",
      "EndTime": "2021-04-01T21:49:18Z",
      "StatusCode": "Successful",
      "Progress": 100,
      "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-west-2a
\"...}",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
    },
...
  ]
}
```

Beispiele für die Erstellung und Verwaltung von Warmpools mit dem AWS CLI

Sie können warme Pools mit den SDKs AWS Management Console, AWS Command Line Interface (AWS CLI) oder verwalten.

Die folgenden Beispiele zeigen, wie Sie Warm Pools mithilfe der AWS CLI erstellen und verwalten.

Inhalt

Beispiel 1: Instances im Zustand **`Stopped`** belassen

Das folgende put-warm-pool-Beispiel erstellt einen Warm Pool, der Instances in einem `Stopped`-Status belässt.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped
```

Beispiel 2: Instances im Zustand **`Running`** belassen

Das folgende put-warm-pool-Beispiel erstellt einen Warm-Pool, der Instances in einem `Running`-Status anstelle eines `Stopped`-Status belässt.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Running
```

Beispiel 3: Instances im Zustand **`Hibernated`** belassen

Das folgende put-warm-pool-Beispiel erstellt einen Warm Pool, der Instances in einem `Hibernated`-Status anstelle eines `Stopped`-Status belässt. Auf diese Weise können Sie Instances stoppen, ohne ihren Speicherinhalt (RAM) zu löschen.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Hibernated
```

Beispiel 4: Instances beim Scale-In wieder in den Warm Pool verschieben

Das folgende put-warm-pool-Beispiel erstellt einen Warm Pool, der Instances in einem `Stopped`-Status belässt und die `--instance-reuse-policy`-Option umfasst. Der Wert `'{"ReuseOnScaleIn": true}'` der Richtlinie für die Instance-Wiederverwendung weist Amazon EC2 Auto Scaling an, Instances an den Warm Pool zurückzugeben, wenn Ihre Auto-Scaling-Gruppe skaliert wird.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --instance-reuse-policy '{"ReuseOnScaleIn": true}'
```

Beispiel 5: Angeben der Mindestanzahl der Instances im Warm Pool

Das folgende put-warm-pool-Beispiel erstellt einen Warm Pool, der mindestens vier Instances beibehält, sodass mindestens vier Instances verfügbar sind, um Datenverkehrsspitzen zu handhaben.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --min-size 4
```

Beispiel 6: Definieren Sie die Größe des warmen Pools mithilfe einer benutzerdefinierten Spezifikation

Standardmäßig verwaltet Amazon EC2 Auto Scaling die Größe Ihres warmen Pools als Differenz zwischen der maximalen und der gewünschten Kapazität der Auto Scaling Scaling-Gruppe. Sie können die Größe des warmen Pools jedoch unabhängig von der maximalen Kapazität der Gruppe verwalten, indem Sie die `--max-group-prepared-capacity` Option verwenden.

Das folgende Put-Warm-Pool-Beispiel erstellt einen warmen Pool und legt die maximale Anzahl von Instanzen fest, die gleichzeitig sowohl in der Warm-Pool- als auch in der Auto Scaling Scaling-Gruppe existieren können. Wenn die Gruppe eine gewünschte Kapazität von 800 hat, hat der warme Pool zunächst eine Größe von 100, da er nach der Ausführung dieses Befehls initialisiert wird.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --max-group-prepared-capacity 900
```

Um eine Mindestanzahl von Instances im Warm-Pool beizubehalten, fügen Sie die `--min-size`-Option mit dem Befehl wie folgt ein.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --max-group-prepared-capacity 900 --min-size 25
```

Beispiel 7: Definieren einer absoluten Warm Pool-Größe

Wenn Sie die `--max-group-prepared-capacity`-und `--min-size`-Optionen auf den gleichen Wert setzen, wird der Warm Pool eine absolute Größe haben. Das folgende put-warm-pool-Beispiel erstellt einen Warm-Pool, der eine konstante Warm-Pool-Größe von zehn Instances beibehält.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --min-size 10 --max-group-prepared-capacity 10
```

Beispiel 8: Einen Warm Pool löschen

Verwenden Sie den Befehl delete-warm-pool, um einen Warm Pool zu löschen.

```
aws autoscaling delete-warm-pool --auto-scaling-group-name my-asg
```

Wenn sich Instances im Warm-Pool befinden oder Skalierungsaktivitäten ausgeführt werden, verwenden Sie den Befehl delete-warm-pool mit der `--force-delete`-Option. Diese Option beendet auch die Amazon-EC2-Instances und alle ausstehenden Lebenszyklusaktionen.

```
aws autoscaling delete-warm-pool --auto-scaling-group-name my-asg --force-delete
```

Instanzen trennen oder anhängen

Sie können Instances von Ihrer Auto Scaling Scaling-Gruppe trennen. Nachdem eine Instanz getrennt wurde, wird diese Instanz unabhängig und kann entweder eigenständig verwaltet oder an eine andere Auto Scaling Scaling-Gruppe angehängt werden, unabhängig von der ursprünglichen Gruppe, zu der sie gehörte. Dies kann beispielsweise nützlich sein, wenn Sie Tests mit vorhandenen Instances durchführen möchten, auf denen Ihre Anwendung bereits ausgeführt wird.

Dieses Thema enthält Anweisungen zum Trennen und Anhängen von Instanzen. Beim Anhängen von Instanzen können Sie statt einer getrennten Instanz auch eine bestehende Instanz verwenden.

Anstatt eine Instanz zu trennen und erneut derselben Gruppe zuzuordnen, empfehlen wir, das Standby-Verfahren zu verwenden, um die Instanz vorübergehend aus der Gruppe zu entfernen. Weitere Informationen finden Sie unter Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe.

Inhalt

Überlegungen zum Trennen von Instanzen

Beachten Sie beim Trennen von Instances die folgenden Punkte:

- Sie können eine Instance nur trennen, wenn sie sich im `InService` Status befindet.

- Nachdem Sie eine Instance getrennt haben, läuft sie weiter und es fallen Gebühren an. Um unnötige Gebühren zu vermeiden, sollten Sie getrennte Instances erneut anhängen oder beenden, wenn sie nicht mehr benötigt werden.
- Sie können sich dafür entscheiden, die gewünschte Kapazität um die Anzahl der Instances zu verringern, die Sie trennen. Wenn Sie sich dafür entscheiden, die Kapazität nicht zu verringern, startet Amazon EC2 Auto Scaling neue Instances, um die getrennten Instanzen zu ersetzen, um die gewünschte Kapazität aufrechtzuerhalten.
- Wenn die Anzahl der Instances, die Sie trennen, dazu führt, dass die Auto Scaling Scaling-Gruppe ihre Mindestkapazität unterschreitet, müssen Sie die Mindestkapazität verringern.
- Wenn Sie mehrere Instances von derselben Availability Zone trennen, ohne die gewünschte Kapazität zu verringern, nimmt die Gruppe das Gleichgewicht von selbst wieder auf, sofern Sie den Vorgang nicht unterbrechen. `AZRebalance` Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling Scaling-Prozesse aussetzen und fortsetzen.
- Wird eine Instance von einer Auto-Scaling-Gruppe mit einer Load Balancer-Zielgruppe oder einem Classic Load Balancer entfernt, wird die Instance beim Load Balancer abgemeldet. Wenn Connection Draining (Abmeldeverzögerung) für Ihren Load Balancer aktiviert ist, wartet Amazon EC2 Auto Scaling auf den Abschluss von eingehenden Anforderungen.

> **Note**
>
> Seien Sie vorsichtig, wenn Sie Instances trennen, die sich im `Standby`-Zustand befinden. Der Versuch, Instances zu trennen, nachdem sie in den `Standby`-Zustand versetzt wurden, kann dazu führen, dass andere Instances unerwartet beendet werden.

Überlegungen zum Anhängen von Instances

Beachten Sie beim Anhängen von Instanzen Folgendes:

- Amazon EC2 Auto Scaling behandelt angehängte Instances genauso wie Instances, die von der Gruppe selbst gestartet wurden. Das bedeutet, dass angehängte Instances bei Scale-In-Ereignissen beendet werden können, sofern sie ausgewählt werden. Die von der AWSServiceRoleForAutoScaling serviceverknüpften Rolle gewährten Berechtigungen ermöglichen Amazon EC2 Auto Scaling, dies zu tun.
- Beim Hinzufügen von Instances erhöht sich die gewünschte Kapazität der Gruppe um die Anzahl der Instances, die hinzugefügt werden. Wenn die gewünschte Kapazität nach dem Hinzufügen

der neuen Instances die maximale Gruppengröße überschreitet, schlägt die Anforderung zum Anhängen weiterer Instances fehl.

- Wenn Sie Instances zu Ihrer Gruppe hinzufügen, was zu einer ungleichmäßigen Verteilung auf die Availability Zones führt, gleicht Amazon EC2 Auto Scaling die Gruppe neu aus, um eine gleichmäßige Verteilung wiederherzustellen, sofern Sie den Vorgang nicht unterbrechen. `AZRebalance` Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling Scaling-Prozesse aussetzen und fortsetzen.
- Wird einer Auto-Scaling-Gruppe mit einer Load Balancer-Zielgruppe oder einem Classic Load Balancer eine Instance hinzugefügt, wird die Instance beim Load Balancer registriert.

Eine zuzuweisende Instance muss die folgenden Kriterien erfüllen:

- Die Instance befindet sich bei Amazon EC2 im Zustand `running`.
- Das AMI zum Starten der Instance ist noch vorhanden.
- Die Instance gehört keiner anderen Auto-Scaling-Gruppe an.
- Die Instance wird in einer der Availability Zones gestartet, die in der Auto Scaling Scaling-Gruppe definiert sind.
- Wenn die Auto-Scaling-Gruppe über eine angefügte Load-Balancer-Zielgruppe oder einen Classic Load Balancer verfügt, müssen sich sowohl die Instance als auch der Load Balancer in derselben VPC befinden.

Verschieben Sie eine Instance mithilfe von Trennen und Anhängen in eine andere Gruppe

Verwenden Sie eines der folgenden Verfahren, um eine Instance von Ihrer Auto Scaling Scaling-Gruppe zu trennen und sie einer anderen Auto Scaling-Gruppe zuzuordnen.

Informationen zum Erstellen einer neuen Auto Scaling Scaling-Gruppe aus einer getrennten Instance finden Sie unter Eine Auto-Scaling-Gruppe unter Verwendung von Parametern einer bestehenden Instance erstellen (nicht empfohlen, erstellt eine Startkonfiguration).

Console

So trennen Sie eine Instance von einer Auto Scaling Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Instance Management (Instance-Verwaltung) unter Instances eine Instance aus, und wählen Sie Actions (Aktionen), Detach (Trennen).
4. Lassen Sie im Dialogfeld Instanz trennen das Kontrollkästchen Instance ersetzen aktiviert, um eine Ersatz-Instance zu starten. Deaktivieren Sie dieses Kontrollkästchen, um die gewünschte Kapazität zu verringern.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`detach`** ein, um das Entfernen der angegebenen Instance aus der Auto-Scaling-Gruppe zu bestätigen. Wählen Sie dann Instance trennen aus.

Sie können die Instance jetzt einer anderen Auto Scaling Scaling-Gruppe zuordnen.

So fügen Sie eine Instance zu einer Auto-Scaling-Gruppe hinzu

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. (Optional) Wählen Sie im Navigationsbereich unter Auto Scaling die Option Auto Scaling Groups (Auto-Scaling-Gruppen) aus. Wählen Sie die Auto-Scaling-Gruppe aus und stellen Sie sicher, dass die Höchstgröße der Auto-Scaling-Gruppe ausreicht, um eine weitere Instance hinzuzufügen. Erhöhen Sie andernfalls auf der Registerkarte Details die maximale Kapazität.
3. Wählen Sie im Navigationsbereich unter Instances die Option Instances und dann eine Instance aus.
4. Wählen Sie Actions, Instance Settings und Attach to Auto Scaling Group aus.
5. Geben Sie auf der Seite Attach to Auto Scaling Group (An Auto-Scaling-Gruppe anhängen) für Auto Scaling group einen Namen für die Gruppe ein und wählen Sie anschließend Attach (Anhängen) aus.
6. Wenn die Instance die Kriterien nicht erfüllt, wird eine Fehlermeldung mit entsprechenden Details ausgegeben. Zum Beispiel befindet sich die Instance möglicherweise nicht in

derselben Availability Zone wie die Auto-Scaling-Gruppe. Wählen Sie Schließen und versuchen Sie es erneut mit einer Auto Scaling Scaling-Gruppe, die die Kriterien erfüllt.

AWS CLI

Verwenden Sie die folgenden Beispielbefehle, um eine Instanz zu trennen und anzuhängen. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

So trennen Sie eine Instance von einer Auto Scaling Scaling-Gruppe

1. Verwenden Sie den folgenden Befehl describe-auto-scaling-instances, um die aktuellen Instanzen zu beschreiben.

```
aws autoscaling describe-auto-scaling-instances \
  --query 'AutoScalingInstances[?AutoScalingGroupName==`my-asg`]'
```

Das folgende Beispiel zeigt die Ausgabe, die erzeugt wird, wenn Sie diesen Befehl ausführen.

Notieren Sie sich die ID der Instanz, die Sie aus der Gruppe entfernen möchten. Sie benötigen diese ID im nächsten Schritt.


```
{
    "AutoScalingInstances": [
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-05b4f7d5be44822a6",
            "InstanceType": "t3.micro",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
```

```
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-0c20ac468fa3049e8",
            "InstanceType": "t3.micro",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-0787762faf1c28619",
            "InstanceType": "t3.micro",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-0f280a4c58d319a8a",
            "InstanceType": "t3.micro",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        }
    ]
}
```


2. Verwenden Sie den folgenden Befehl detach-instances, um eine Instance zu trennen, ohne die gewünschte Kapazität zu verringern.

```
aws autoscaling detach-instances --instance-ids i-05b4f7d5be44822a6 \
  --auto-scaling-group-name my-asg
```

Um eine Instance zu trennen und die gewünschte Kapazität zu verringern, fügen Sie die Option hinzu. `--should-decrement-desired-capacity`

```
aws autoscaling detach-instances --instance-ids i-05b4f7d5be44822a6 \
  --auto-scaling-group-name my-asg --should-decrement-desired-capacity
```

Sie können die Instance jetzt einer anderen Auto Scaling Scaling-Gruppe zuordnen.

So fügen Sie eine Instance zu einer Auto-Scaling-Gruppe hinzu

1. Verwenden Sie den folgenden Befehl attach-instances, um die Instance an eine andere Auto Scaling Scaling-Gruppe anzuhängen.

```
aws autoscaling attach-instances --instance-ids i-05b4f7d5be44822a6 --auto-
scaling-group-name my-asg-for-testing
```

2. Verwenden Sie den folgenden Befehl describe-auto-scaling-groups, um die Größe der Auto Scaling-Gruppe nach dem Anhängen einer Instance zu überprüfen.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg-
for-testing
```

Die folgende Beispielantwort zeigt, dass die Gruppe über zwei laufende Instances verfügt, von denen eine die Instance ist, die Sie angehängt haben.

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg-for-testing",
            "AutoScalingGroupARN": "arn",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "2",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "MinSize": 1,
```

```
            "MaxSize": 5,
            "DesiredCapacity": 2,
            ...
            "Instances": [
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-05b4f7d5be44822a6",
                    "InstanceType": "t3.micro",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService"
                },
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "2",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-00dcdfffdf5175890",
                    "InstanceType": "t3.micro",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService"
                }
            ],
            ...
        }
    ]
}
```

Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe

Sie können eine Instance mit dem Status `InService` in den Status `Standby` versetzen, die Instance aktualisieren oder Probleme mit ihr beheben und sie dann wieder in Betrieb nehmen. Instances im

Standby-Status sind immer noch Teil der Auto-Scaling-Gruppe, beteiligen sich aber nicht aktiv an der Verarbeitung von Load Balancer-Datenverkehr.

Diese Funktion hilft Ihnen, die Instances zu stoppen und zu starten oder sie neu zu starten, ohne sich Gedanken darüber zu machen, dass Amazon EC2 Auto Scaling die Instances im Rahmen seiner Zustandsprüfungen oder während Scale-In-Ereignissen beendet.

Sie können beispielsweise das Amazon Machine Image (AMI) einer Auto-Scaling-Gruppe jederzeit ändern, indem Sie die Startvorlage oder die Startkonfiguration ändern. Alle nachfolgenden Instances, die von der Auto-Scaling-Gruppe gestartet werden, verwenden dieses AMI. Allerdings aktualisiert die Auto-Scaling-Gruppe keine Instances, die derzeit verwendet werden. Sie können diese Instances beenden und Amazon EC2 Auto Scaling ersetzen lassen oder die Instance-Aktualisierungsfunktion verwenden, um die Instances zu beenden und zu ersetzen. Sie können auch die Instances in den Standby-Status versetzen, die Software aktualisieren und die Instances daraufhin wieder in Betrieb nehmen.

Das Trennen von Instances von einer Auto-Scaling-Gruppe ähnelt dem Setzen von Instances in den Standby-Modus. Das Trennen von Instances kann nützlich sein, wenn Sie sie einer anderen Gruppe zuordnen oder die Instances wie eigenständige EC2-Instances verwalten und sie möglicherweise beenden möchten. Weitere Informationen finden Sie unter Instanzen trennen oder anhängen.

Inhalt

So funktioniert der Standby-Status

Der Standby-Status ermöglicht das vorübergehende Entfernen einer Instance aus der Auto-Scaling-Gruppe wie folgt:

1. Sie versetzen eine Instance in den Standby-Status. Die Instance verbleibt in diesem Status, bis Sie den Standby-Status beenden.
2. Ist eine Load Balancer-Zielgruppe oder ein Classic Load Balancer mit der Auto-Scaling-Gruppe verknüpft, wird die Instance vom Load Balancer abgemeldet. Ist der Connection Draining für den

Load Balancer aktiviert, wartet Elastic Load Balancing standardmäßig 300 Sekunden, bevor die Registrierung abgeschlossen wird. So können laufende Anfragen abgeschlossen werden.

3. Sie können die Instance aktualisieren oder Probleme mit ihr beheben.
4. Die Instance wird wieder in Betrieb genommen, indem der Standby-Status aufgehoben wird.
5. Ist eine Load Balancer-Zielgruppe oder ein Classic Load Balancer mit der Auto-Scaling-Gruppe verknüpft, wird die Instance mit Load Balancer angemeldet.

Weitere Informationen zum Lebenszyklus der Instances in einer Auto-Scaling-Gruppe finden Sie unter Instance-Lebenszyklus bei Amazon EC2 Auto Scaling.

Überlegungen

Beim Verschieben von Instances in den Standby-Status und aus dem Standby-Status ist Folgendes zu beachten:

- Wenn Sie eine Instance in den Standby-Status versetzen, können Sie die gewünschte Kapazität entweder durch diesen Vorgang verringern oder den Wert beibehalten.
  - Wenn Sie die gewünschte Kapazität der Auto-Scaling-Gruppe nicht verringern möchten, startet Amazon EC2 Auto Scaling eine Instance, um diejenige im Standby-Status zu ersetzen. Ziel ist es, Ihnen dabei zu helfen, die Kapazität für Ihre Anwendung aufrechtzuerhalten, während sich eine oder mehrere Instances im Standby-Modus befinden.
  - Wenn Sie die gewünschte Kapazität der Auto-Scaling-Gruppe verringern möchten, wird der Start einer Instance verhindert, mit dem die Instance im Standby-Status ersetzt wird.
- Nachdem Sie die Instance wieder in Betrieb genommen haben, wird die gewünschte Kapazität entsprechend der Anzahl der Instances in der Auto-Scaling-Gruppe erhöht.
- Um das Erhöhen (und Verringern) durchführen zu können, muss die neue gewünschte Kapazität zwischen der minimalen und der maximalen Gruppengröße liegen. Andernfalls schlägt die Operation fehl.
- Wenn zu irgendeinem Zeitpunkt, nachdem Sie eine Instance in den Standby-Modus versetzt oder die Instance durch Verlassen des Standby-Status wieder in Betrieb genommen haben, festgestellt wird, dass Ihre Auto-Scaling-Gruppe nicht zwischen den Availability Zones ausgeglichen wird, gleicht Amazon EC2 Auto Scaling dies aus, indem es die Availability Zones neu verteilt, sofern Sie den `AZRebalance`-Prozess nicht unterbrechen. Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling Scaling-Prozesse aussetzen und fortsetzen.
- Instances im Standby-Status werden Ihnen in Rechnung gestellt.

Zustand einer Instance im Standby-Status

Amazon EC2 Auto Scaling prüft den Zustand von Instances, die sich im Standby-Modus befinden, nicht. Solange sich die Instance in einem Standby-Status befindet, hat sie denselben Zustand wie vor dem Standby. Amazon EC2 Auto Scaling prüft den Zustand der Instance nicht, bis sie wieder in Betrieb genommen wird.

Wenn Sie beispielsweise eine funktionierende Instance in den Standby-Modus versetzen und dann beenden, zeigt Amazon EC2 Auto Scaling die Instance weiterhin als fehlerfrei an. Wenn Sie versuchen, die beendete Standby-Instance erneut in Betrieb zu nehmen, prüft Amazon EC2 Auto Scaling den Zustand der Instance. Falls dabei festgestellt wird, dass sie beendet wird und fehlerhaft ist, wird eine Ersatz-Instance gestartet. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Entfernen Sie eine Instance vorübergehend, indem Sie sie in den Standby-Modus versetzen

Verwenden Sie eines der folgenden Verfahren, um eine Instanz vorübergehend außer Betrieb zu setzen, indem Sie sie in den Standby-Status versetzen.

Console

So entfernen Sie eine Instance vorübergehend:

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Instance management (Instance-Verwaltung) unter Instances eine Instance aus.
4. Wählen Sie Actions, Set to Standby aus.
5. Lassen Sie im Dialogfeld Auf Standby setzen das Kontrollkästchen Instance ersetzen aktiviert, um eine Ersatz-Instance zu starten. Deaktivieren Sie dieses Kontrollkästchen, um die gewünschte Kapazität zu verringern.
6. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`standby`** ein, um zu bestätigen, dass die angegebene Instance in den Status `Standby` versetzt wird, und wählen Sie dann Auf Standby setzen aus.

7. Sie können die Instance nach Bedarf aktualisieren oder Probleme mit ihr beheben. Wenn Sie fertig sind, fahren Sie mit dem nächsten Schritt fort, um die Instance erneut in Betrieb zu nehmen.
8. Wählen Sie die Instanz aus, wählen Sie Aktionen, Set to aus InService. Wählen Sie im InService Dialogfeld „Set to“ die Option „Set to“ aus InService.

AWS CLI

Verwenden Sie die folgenden Beispielbefehle, um eine Instance vorübergehend aus Ihrer Auto Scaling Scaling-Gruppe zu entfernen. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

So entfernen Sie eine Instance vorübergehend:

1. Verwenden Sie den folgenden describe-auto-scaling-instances-Befehl, um die zu aktualisierende Instance zu identifizieren:

```
aws autoscaling describe-auto-scaling-instances \
  --query 'AutoScalingInstances[?AutoScalingGroupName==`my-asg`]'
```

Das folgende Beispiel zeigt die Ausgabe, die erzeugt wird, wenn Sie diesen Befehl ausführen.

Notieren Sie sich die ID der Instanz, die Sie aus der Gruppe entfernen möchten. Sie benötigen diese ID im nächsten Schritt.

```
{
    "AutoScalingInstances": [
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-05b4f7d5be44822a6",
            "InstanceId": "t3.micro",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
```

```
        },
        ...
    ]
}
```

2. Verwenden Sie den folgenden `Standby`enter-standby-Befehl, um die Instance in einen -Status zu versetzen. Die Option `--should-decrement-desired-capacity` senkt die gewünschte Kapazität, so dass die Auto-Scaling-Gruppe keine Ersatz-Instance startet.

```
aws autoscaling enter-standby --instance-ids i-05b4f7d5be44822a6 \
  --auto-scaling-group-name my-asg --should-decrement-desired-capacity
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "Activities": [
        {
            "ActivityId": "3b1839fe-24b0-40d9-80ae-bcd883c2be32",
            "AutoScalingGroupName": "my-asg",
            "Description": "Moving EC2 instance to Standby:
 i-05b4f7d5be44822a6",
            "Cause": "At 2023-12-15T21:31:26Z instance i-05b4f7d5be44822a6 was
 moved to standby
              in response to a user request, shrinking the capacity from 4 to
 3.",
            "StartTime": "2023-12-15T21:31:26.150Z",
            "StatusCode": "InProgress",
            "Progress": 50,
            "Details": "{\"Subnet ID\":\"subnet-c934b782\",\"Availability Zone
\":\"us-west-2a\"}"
        }
    ]
}
```

3. (Optional) Überprüfen Sie, ob sich die Instance im Modus `Standby` befindet, indem Sie den folgenden describe-auto-scaling-Instances-Befehl verwenden.

```
aws autoscaling describe-auto-scaling-instances --instance-
ids i-05b4f7d5be44822a6
```

Nachfolgend finden Sie eine Beispielantwort. Der Status der Instance lautet jetzt `Standby`.

```
{
    "AutoScalingInstances": [
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-05b4f7d5be44822a6",
            "InstanceType": "t3.micro",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "Standby"
        },
        ...
    ]
}
```

4. Sie können die Instance nach Bedarf aktualisieren oder Probleme mit ihr beheben. Wenn Sie fertig sind, fahren Sie mit dem nächsten Schritt fort, um die Instance erneut in Betrieb zu nehmen.
5. Verwenden Sie den folgenden exit-standby-Befehl, um die Instance wieder in Betrieb zu nehmen:

```
aws autoscaling exit-standby --instance-ids i-05b4f7d5be44822a6 --auto-scaling-
group-name my-asg
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "Activities": [
        {
            "ActivityId": "db12b166-cdcc-4c54-8aac-08c5935f8389",
            "AutoScalingGroupName": "my-asg",
            "Description": "Moving EC2 instance out of Standby:
 i-05b4f7d5be44822a6",
            "Cause": "At 2023-12-15T21:46:14Z instance i-05b4f7d5be44822a6 was
 moved out of standby in
```

```
                response to a user request, increasing the capacity from 3 to
 4.",
            "StartTime": "2023-12-15T21:46:14.678Z",
            "StatusCode": "PreInService",
            "Progress": 30,
            "Details": "{\"Subnet ID\":\"subnet-c934b782\",\"Availability Zone
\":\"us-west-2a\"}"
        }
    ]
}
```

6. (Optional) Verwenden Sie den folgenden Befehl `describe-auto-scaling-instances`, um zu überprüfen, ob die Instance wieder in Betrieb ist:

```
aws autoscaling describe-auto-scaling-instances --instance-
ids i-05b4f7d5be44822a6
```

Nachfolgend finden Sie eine Beispielantwort. Der Status der Instance lautet `InService`.

```
{
    "AutoScalingInstances": [
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-05b4f7d5be44822a6",
            "InstanceType": "t3.micro",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
       ...
    ]
}
```

Löschen der Auto-Scaling-Infrastruktur

Führen Sie die folgenden Schritte aus, um die Skalierungsinfrastruktur vollständig zu löschen.

Aufgaben

- Löschen Ihrer Auto-Scaling-Gruppe
- (Optional) Löschen der Startkonfiguration
- (Optional) Löschen Sie die Startvorlage
- (Optional) Löschen des Load Balancers und der Zielgruppen
- (Optional) CloudWatch Alarme löschen

Löschen Ihrer Auto-Scaling-Gruppe

Beim Löschen einer Auto-Scaling-Gruppe werden die gewünschte, minimale und maximale Größe auf 0 eingestellt. Dadurch werden die Instances beendet. Durch das Löschen einer Instance werden auch alle zugehörigen Protokolle oder Daten sowie alle Volumes der Instance gelöscht. Wenn Sie nicht möchten, dass eine oder mehrere Instances beendet werden, können Sie sie trennen, bevor Sie die Auto-Scaling-Gruppe löschen. Wenn die Gruppe Skalierungsrichtlinien besitzt, werden durch Löschen der Gruppe auch die Richtlinien, die zugrunde liegenden Alarmaktionen und alle Alarme gelöscht, denen keine Aktion mehr zugeordnet ist.

So löschen Sie die Auto-Scaling-Gruppe (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe und wählen Sie Aktionen, Löschen aus.
3. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`delete`** ein, um das Löschen der angegebenen Auto-Scaling-Gruppe zu löschen, wählen Sie dann Löschen.

   Ein Ladesymbol in der Spalte Name zeigt an, dass die Auto-Scaling-Gruppe gelöscht wird. Die Spalten Desired (Gewünscht), Min und Max zeigen `0`-Instances der Auto-Scaling-Gruppe an. Es dauert einige Minuten, bis die Instance beendet und die Gruppe gelöscht werden. Aktualisieren Sie die Liste, um den aktuellen Status anzuzeigen.

So löschen Sie Ihre Auto-Scaling-Gruppe (AWS CLI)

Verwenden Sie den folgenden delete-auto-scaling-group-Befehl zum Löschen der Auto-Scaling-Gruppe: Dieser Vorgang funktioniert nicht, wenn die Gruppe über EC2-Instances verfügt; er ist nur für Gruppen mit null Instanzen vorgesehen.

```
aws autoscaling delete-auto-scaling-group --auto-scaling-group-name my-asg
```

Wenn in der Gruppe Instances oder Skalierungsaktivitäten ausgeführt werden, verwenden Sie den Befehl delete-auto-scaling-group mit der Option `--force-delete`. Dadurch werden auch die EC2-Instances beendet. Wenn Sie eine Auto Scaling-Gruppe aus der Amazon EC2 Auto Scaling-Konsole löschen, verwendet die Konsole diesen Vorgang, um alle EC2-Instances zu beenden und gleichzeitig die Gruppe zu löschen.

```
aws autoscaling delete-auto-scaling-group --auto-scaling-group-name my-asg --force-
delete
```

(Optional) Löschen der Startkonfiguration

Sie können diesen Schritt überspringen, um die Startkonfiguration für die spätere Verwendung beizubehalten.

So löschen Sie die Startkonfiguration (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im linken Navigationsbereich unter Auto Scaling Auto-Scaling-Gruppen aus.
3. Wählen Sie oben auf der Seite Startkonfigurationen aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Startkonfigurationen anzeigen aus, um zu bestätigen, dass Sie die Seite Startkonfigurationen aufrufen möchten.
4. Wählen Sie Ihre Startkonfiguration und anschließend Aktionen, Startkonfiguration löschen aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).

So löschen Sie die Startkonfiguration (AWS CLI)

Verwenden Sie den folgenden delete-launch-configuration-Befehl:

```
aws autoscaling delete-launch-configuration --launch-configuration-name my-launch-
config
```

(Optional) Löschen Sie die Startvorlage

Sie können Ihre Startvorlage oder nur eine Version Ihrer Startvorlage löschen. Beim Löschen einer Startvorlage werden alle Versionen davon gelöscht.

Sie können diesen Schritt überspringen, um die Startvorlage für die spätere Verwendung aufzubewahren.

So löschen Sie eine Startvorlage (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Launch Templates aus.
3. Wählen Sie Ihre Startvorlage aus und führen Sie dann einen der folgenden Schritte aus:

   - Wählen Sie Actions (Aktionen) und Delete template (Vorlage löschen) aus. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`Delete`** ein, um das Löschen der angegebenen Auto-Scaling-Gruppe zu bestätigen, wählen Sie dann Löschen.
   - Wählen Sie Actions (Aktionen) und Delete template version (Vorlagenversion löschen) aus. Wählen Sie die zu löschende Version aus und wählen Sie Delete (Löschen).

So löschen Sie die Startvorlage (AWS CLI)

Verwenden Sie den folgenden delete-launch-template-Befehl, um Ihre Vorlage und alle Versionen davon zu löschen.

```
aws ec2 delete-launch-template --launch-template-id lt-068f72b72934aff71
```

Alternativ können Sie den delete-launch-template-versions-Befehl verwenden, um eine bestimmte Version einer Startvorlage zu löschen.

```
aws ec2 delete-launch-template-versions --launch-template-id lt-068f72b72934aff71 --
versions 1
```

(Optional) Löschen des Load Balancers und der Zielgruppen

Sie können diesen Schritt überspringen, wenn die Auto-Scaling-Gruppe nicht bei einem Elastic Load Balancing-Load Balancer angemeldet ist, oder wenn Sie den Load Balancer behalten und in Zukunft weiter benutzen möchten.

So löschen Sie den Load Balancer (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter LOAD BALANCING die Option Load Balancers aus.
3. Wählen Sie den Load Balancer aus und klicken Sie auf Actions (Aktionen) und dann auf Delete (Löschen).
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Delete (Ja, löschen).

So löschen Sie Ihre Zielgruppe (Konsole)

1. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
2. Wählen Sie Ihre Zielgruppe aus und klicken Sie dann auf Actions (Aktionen), Delete (Löschen).
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Ja, löschen.

So löschen Sie den mit der Auto-Scaling-Gruppe verknüpften Load Balancer (AWS CLI)

Verwenden Sie für Application Load Balancer und Network Load Balancer die folgenden delete-load-balancer- und delete-targetgroup-Befehle.

```
aws elbv2 delete-load-balancer --load-balancer-arn my-load-balancer-arn
aws elbv2 delete-target-group --target-group-arn my-target-group-arn
```

Verwenden Sie für Classic Load Balancer den folgenden delete-load-balancer-Befehl.

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

(Optional) CloudWatch Alarme löschen

Gehen Sie wie folgt vor, um die mit Ihrer Auto Scaling Scaling-Gruppe verknüpften CloudWatch Alarme zu löschen. So können Sie beispielsweise Alarme im Zusammenhang mit schrittweiser Skalierung oder einfachen Skalierungsrichtlinien haben.

> **Note**
>
> Durch das Löschen einer Auto Scaling-Gruppe werden automatisch die CloudWatch Alarme gelöscht, die Amazon EC2 Auto Scaling für eine Skalierungsrichtlinie zur Zielverfolgung verwaltet.

Sie können diesen Schritt überspringen, wenn Ihre Auto Scaling Scaling-Gruppe keinen CloudWatch Alarmen zugeordnet ist oder wenn Sie die Alarme für die future Verwendung behalten möchten.

Um die CloudWatch Alarme zu löschen (Konsole)

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms (Alarme) aus.
3. Wählen Sie die Alarme aus und klicken Sie dann auf Action (Aktion), Delete (Löschen).
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).

Um die CloudWatch Alarme zu löschen (AWS CLI)

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-AddCapacity` und `Step-Scaling-AlarmLow-RemoveCapacity` zu löschen.

```
aws cloudwatch delete-alarms --alarm-name Step-Scaling-AlarmHigh-AddCapacity Step-Scaling-AlarmLow-RemoveCapacity
```

Beispiele für die Erstellung und Verwaltung von Auto Scaling Scaling-Gruppen mit den AWS SDKs

Sie können eine Auto Scaling Scaling-Gruppe mit dem AWS Management Console AWS CLI, dem, einem AWS SDK und erstellen AWS CloudFormation.

Die folgenden Codebeispiele zeigen, wie Sie mithilfe der AWS SDKs eine Auto Scaling Scaling-Gruppe in Ihrer bevorzugten unterstützten Programmiersprache erstellen, aktualisieren, beschreiben und löschen.

Inhalt

Eine Auto Scaling Scaling-Gruppe mithilfe eines AWS SDK erstellen

Die folgenden Codebeispiele zeigen, wie man es benutzt`CreateAutoScalingGroup`.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
/// <summary>
/// Create a new Amazon EC2 Auto Scaling group.
/// </summary>
/// <param name="groupName">The name to use for the new Auto Scaling
/// group.</param>
/// <param name="launchTemplateName">The name of the Amazon EC2 Auto Scaling
/// launch template to use to create instances in the group.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> CreateAutoScalingGroupAsync(
    string groupName,
    string launchTemplateName,
    string availabilityZone)
{
    var templateSpecification = new LaunchTemplateSpecification
    {
        LaunchTemplateName = launchTemplateName,
    };
```

```
        var zoneList = new List<string>
            {
                availabilityZone,
            };

        var request = new CreateAutoScalingGroupRequest
        {
            AutoScalingGroupName = groupName,
            AvailabilityZones = zoneList,
            LaunchTemplate = templateSpecification,
            MaxSize = 6,
            MinSize = 1
        };

        var response = await
_amazonAutoScaling.CreateAutoScalingGroupAsync(request);
        Console.WriteLine($"{groupName} Auto Scaling Group created");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Einzelheiten zur API finden Sie CreateAutoScalingGroupin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::AutoScaling::AutoScalingClient autoScalingClient(clientConfig);
```

```
        Aws::AutoScaling::Model::CreateAutoScalingGroupRequest request;
        request.SetAutoScalingGroupName(groupName);
        Aws::Vector<Aws::String> availabilityGroupZones;
        availabilityGroupZones.push_back(
                availabilityZones[availabilityZoneChoice - 1].GetZoneName());
        request.SetAvailabilityZones(availabilityGroupZones);
        request.SetMaxSize(1);
        request.SetMinSize(1);

        Aws::AutoScaling::Model::LaunchTemplateSpecification
launchTemplateSpecification;
        launchTemplateSpecification.SetLaunchTemplateName(templateName);
        request.SetLaunchTemplate(launchTemplateSpecification);

        Aws::AutoScaling::Model::CreateAutoScalingGroupOutcome outcome =
                autoScalingClient.CreateAutoScalingGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "Created Auto Scaling group '" << groupName << "'..."
                      << std::endl;
        }
        else if (outcome.GetError().GetErrorType() ==
                 Aws::AutoScaling::AutoScalingErrors::ALREADY_EXISTS_FAULT) {
            std::cout << "Auto Scaling group '" << groupName << "' already
exists."
                      << std::endl;
        }
        else {
            std::cerr << "Error with AutoScaling::CreateAutoScalingGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;

        }
```

- Einzelheiten zur API finden Sie CreateAutoScalingGroupin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Beispiel 1: So erstellen Sie eine Auto Scaling Scaling-Gruppe

Im folgenden `create-auto-scaling-group` Beispiel wird eine Auto Scaling Scaling-Gruppe in Subnetzen in mehreren Availability Zones innerhalb einer Region erstellt. Die Instances werden mit der Standardversion der angegebenen Startvorlage gestartet. Beachten Sie, dass Standardwerte für die meisten anderen Einstellungen verwendet werden, z. B. für die Kündigungsrichtlinien und die Konfiguration der Integritätsprüfung.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12 \
    --min-size 1 \
    --max-size 5 \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zu Auto-Scaling-Gruppen finden Sie im Benutzerhandbuch für Amazon EC2 Auto Scaling.

Beispiel 2: So fügen Sie einen Application Load Balancer, Network Load Balancer oder Gateway Load Balancer an

In diesem Beispiel wird der ARN einer Zielgruppe für einen Load Balancer angegeben, der den erwarteten Traffic unterstützt. Der Integritätsprüfungstyp gibt an, `ELB` dass, wenn Elastic Load Balancing eine Instance als fehlerhaft meldet, die Auto Scaling Scaling-Gruppe sie ersetzt. Der Befehl definiert auch eine Übergangszeit von `600` Sekunden für die Integritätsprüfung. Die Übergangszeit trägt dazu bei, eine vorzeitige Kündigung neu gestarteter Instances zu verhindern.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12 \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/943f017f100becff \
    --health-check-type ELB \
    --health-check-grace-period 600 \
    --min-size 1 \
    --max-size 5 \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Elastic Load Balancing und Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.

Beispiel 3: Um eine Platzierungsgruppe anzugeben und die neueste Version der Startvorlage zu verwenden

In diesem Beispiel werden Instances in einer Platzierungsgruppe innerhalb einer einzelnen Availability Zone gestartet. Dies kann für Gruppen mit niedriger Latenz und HPC-Workloads nützlich sein. In diesem Beispiel werden auch die Mindestgröße, die Maximalgröße und die gewünschte Kapazität der Gruppe angegeben.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest' \
    --min-size 1 \
    --max-size 5 \
    --desired-capacity 3 \
    --placement-group my-placement-group \
    --vpc-zone-identifier "subnet-6194ea3b"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Platzierungsgruppen im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

Beispiel 4: Um eine Auto Scaling Scaling-Gruppe für eine einzelne Instanz anzugeben und eine bestimmte Version der Startvorlage zu verwenden

In diesem Beispiel wird eine Auto Scaling Scaling-Gruppe erstellt, deren Mindest- und Höchstkapazität auf festgelegt sind, 1 um zu erzwingen, dass eine Instance ausgeführt wird. Der Befehl gibt auch Version 1 einer Startvorlage an, in der die ID einer vorhandenen ENI angegeben ist. Wenn Sie eine Startvorlage verwenden, die eine vorhandene ENI für eth0 angibt, müssen Sie eine Availability Zone für die Auto Scaling Scaling-Gruppe angeben, die der Netzwerkschnittstelle entspricht, ohne auch eine Subnetz-ID in der Anfrage anzugeben.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg-single-instance \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='1'
 \
    --min-size 1 \
    --max-size 1 \
```

```
    --availability-zones us-west-2a
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zu Auto-Scaling-Gruppen finden Sie im Benutzerhandbuch für Amazon EC2 Auto Scaling.

Beispiel 5: Um eine andere Kündigungsrichtlinie anzugeben

In diesem Beispiel wird eine Auto Scaling Scaling-Gruppe mithilfe einer Startkonfiguration erstellt und die Kündigungsrichtlinie so festgelegt, dass die ältesten Instances zuerst beendet werden. Der Befehl weist der Gruppe und ihren Instances außerdem ein Tag mit dem Schlüssel `Role` und dem Wert von zu`WebServer`.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-configuration-name my-lc \
    --min-size 1 \
    --max-size 5 \
    --termination-policies "OldestInstance" \
    --tags "ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=Role,Value=WebServer,PropagateAtLaunch=true" \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Arbeiten mit Amazon EC2 Auto Scaling Scaling-Kündigungsrichtlinien im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 6: So geben Sie einen Launch-Lifecycle-Hook an

In diesem Beispiel wird eine Auto Scaling Scaling-Gruppe mit einem Lifecycle-Hook erstellt, der eine benutzerdefinierte Aktion beim Instance-Start unterstützt.

```
aws autoscaling create-auto-scaling-group \
    --cli-input-json file://~/config.json
```

Inhalt der `config.json` Datei:

```
{
    "AutoScalingGroupName": "my-asg",
```

```
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-1234567890abcde12"
    },
    "LifecycleHookSpecificationList": [{
        "LifecycleHookName": "my-launch-hook",
        "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
        "NotificationTargetARN": "arn:aws:sqs:us-west-2:123456789012:my-sqs-
queue",
        "RoleARN": "arn:aws:iam::123456789012:role/my-notification-role",
        "NotificationMetadata": "SQS message metadata",
        "HeartbeatTimeout": 4800,
        "DefaultResult": "ABANDON"
    }],
    "MinSize": 1,
    "MaxSize": 5,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
    "Tags": [{
        "ResourceType": "auto-scaling-group",
        "ResourceId": "my-asg",
        "PropagateAtLaunch": true,
        "Value": "test",
        "Key": "environment"
    }]
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Lebenszyklus-Hooks für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.

Beispiel 7: Um einen Termination-Lifecycle-Hook anzugeben

In diesem Beispiel wird eine Auto Scaling Scaling-Gruppe mit einem Lifecycle-Hook erstellt, der eine benutzerdefinierte Aktion beim Beenden der Instanz unterstützt.

```
aws autoscaling create-auto-scaling-group \
    --cli-input-json file://~/config.json
```

Inhalt von `config.json`:

```
{
    "AutoScalingGroupName": "my-asg",
    "LaunchTemplate": {
```

```
        "LaunchTemplateId": "lt-1234567890abcde12"
    },
    "LifecycleHookSpecificationList": [{
        "LifecycleHookName": "my-termination-hook",
        "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",
        "HeartbeatTimeout": 120,
        "DefaultResult": "CONTINUE"
    }],
    "MinSize": 1,
    "MaxSize": 5,
    "TargetGroupARNs": [
        "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067"
    ],
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Lebenszyklus-Hooks für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.

Beispiel 8: Um eine benutzerdefinierte Kündigungsrichtlinie anzugeben

In diesem Beispiel wird eine Auto Scaling-Gruppe erstellt, die eine benutzerdefinierte Richtlinie zur Beendigung von Lambda-Funktionen spezifiziert, die Amazon EC2 Auto Scaling mitteilt, welche Instances sicher bei der Skalierung beendet werden können.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg-single-instance \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling \
    --min-size 1 \
    --max-size 5 \
    --termination-policies "arn:aws:lambda:us-
west-2:123456789012:function:HelloFunction:prod" \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Erstellen einer benutzerdefinierten Kündigungsrichtlinie mit Lambda im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter CreateAutoScalingGroup AWS CLIBefehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.core.waiters.WaiterResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.autoscaling.AutoScalingClient;
import software.amazon.awssdk.services.autoscaling.model.AutoScalingException;
import
 software.amazon.awssdk.services.autoscaling.model.CreateAutoScalingGroupRequest;
import
 software.amazon.awssdk.services.autoscaling.model.DescribeAutoScalingGroupsRequest;
import
 software.amazon.awssdk.services.autoscaling.model.DescribeAutoScalingGroupsResponse;
import
 software.amazon.awssdk.services.autoscaling.model.LaunchTemplateSpecification;
import software.amazon.awssdk.services.autoscaling.waiters.AutoScalingWaiter;

/**
 * Before running this SDK for Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateAutoScalingGroup {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <groupName> <launchTemplateName> <serviceLinkedRoleARN>
 <vpcZoneId>

                Where:
```

```
                    groupName - The name of the Auto Scaling group.
                    launchTemplateName - The name of the launch template.\s
                    vpcZoneId - A subnet Id for a virtual private cloud (VPC)
where instances in the Auto Scaling group can be created.
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String groupName = args[0];
        String launchTemplateName = args[1];
        String vpcZoneId = args[2];
        AutoScalingClient autoScalingClient = AutoScalingClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createAutoScalingGroup(autoScalingClient, groupName, launchTemplateName,
vpcZoneId);
        autoScalingClient.close();
    }

    public static void createAutoScalingGroup(AutoScalingClient
autoScalingClient,
            String groupName,
            String launchTemplateName,
            String vpcZoneId) {

        try {
            AutoScalingWaiter waiter = autoScalingClient.waiter();
            LaunchTemplateSpecification templateSpecification =
LaunchTemplateSpecification.builder()
                    .launchTemplateName(launchTemplateName)
                    .build();

            CreateAutoScalingGroupRequest request =
CreateAutoScalingGroupRequest.builder()
                    .autoScalingGroupName(groupName)
                    .availabilityZones("us-east-1a")
                    .launchTemplate(templateSpecification)
                    .maxSize(1)
                    .minSize(1)
                    .vpcZoneIdentifier(vpcZoneId)
```

```
                    .build();

            autoScalingClient.createAutoScalingGroup(request);
            DescribeAutoScalingGroupsRequest groupsRequest =
 DescribeAutoScalingGroupsRequest.builder()
                    .autoScalingGroupNames(groupName)
                    .build();

            WaiterResponse<DescribeAutoScalingGroupsResponse> waiterResponse =
 waiter
                    .waitUntilGroupExists(groupsRequest);
            waiterResponse.matched().response().ifPresent(System.out::println);
            System.out.println("Auto Scaling Group created");

        } catch (AutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie CreateAutoScalingGroupin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun createAutoScalingGroup(
    groupName: String,
    launchTemplateNameVal: String,
    serviceLinkedRoleARNVal: String,
    vpcZoneIdVal: String
) {
```

```
    val templateSpecification =
        LaunchTemplateSpecification {
            launchTemplateName = launchTemplateNameVal
        }

    val request =
        CreateAutoScalingGroupRequest {
            autoScalingGroupName = groupName
            availabilityZones = listOf("us-east-1a")
            launchTemplate = templateSpecification
            maxSize = 1
            minSize = 1
            vpcZoneIdentifier = vpcZoneIdVal
            serviceLinkedRoleArn = serviceLinkedRoleARNVal
        }

    // This object is required for the waiter call.
    val groupsRequestWaiter =
        DescribeAutoScalingGroupsRequest {
            autoScalingGroupNames = listOf(groupName)
        }

    AutoScalingClient { region = "us-east-1" }.use { autoScalingClient ->
        autoScalingClient.createAutoScalingGroup(request)
        autoScalingClient.waitUntilGroupExists(groupsRequestWaiter)
        println("$groupName was created!")
    }
}
```

- Einzelheiten zur API finden Sie CreateAutoScalingGroupin der API-Referenz zum AWS SDK für Kotlin.

PHP

SDK für PHP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public function createAutoScalingGroup(
        $autoScalingGroupName,
        $availabilityZones,
        $minSize,
        $maxSize,
        $launchTemplateId
    ) {
        return $this->autoScalingClient->createAutoScalingGroup([
            'AutoScalingGroupName' => $autoScalingGroupName,
            'AvailabilityZones' => $availabilityZones,
            'MinSize' => $minSize,
            'MaxSize' => $maxSize,
            'LaunchTemplate' => [
                'LaunchTemplateId' => $launchTemplateId,
            ],
        ]);
    }
```

- Einzelheiten zur API finden Sie CreateAutoScalingGroupin der AWS SDK for PHP API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Auto Scaling Scaling-Gruppe mit dem angegebenen Namen und den angegebenen Attributen erstellt. Die standardmäßig gewünschte Kapazität ist die Mindestgröße. Daher startet diese Auto Scaling Scaling-Gruppe zwei Instances, eine in jeder der angegebenen zwei Availability Zones.

```
New-ASAutoScalingGroup -AutoScalingGroupName my-asg -LaunchConfigurationName my-
lc -MinSize 2 -MaxSize 6 -AvailabilityZone @("us-west-2a", "us-west-2b")
```

- Einzelheiten zur API finden Sie unter CreateAutoScalingGroup AWS Tools for PowerShellCmdlet-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class AutoScalingWrapper:
    """Encapsulates Amazon EC2 Auto Scaling actions."""

    def __init__(self, autoscaling_client):
        """
        :param autoscaling_client: A Boto3 Amazon EC2 Auto Scaling client.
        """
        self.autoscaling_client = autoscaling_client


    def create_group(
        self, group_name, group_zones, launch_template_name, min_size, max_size
    ):
        """
        Creates an Auto Scaling group.

        :param group_name: The name to give to the group.
        :param group_zones: The Availability Zones in which instances can be
 created.
        :param launch_template_name: The name of an existing Amazon EC2 launch
 template.
                                     The launch template specifies the
 configuration of
                                     instances that are created by auto scaling
 activities.
        :param min_size: The minimum number of active instances in the group.
        :param max_size: The maximum number of active instances in the group.
        """
        try:
            self.autoscaling_client.create_auto_scaling_group(
                AutoScalingGroupName=group_name,
                AvailabilityZones=group_zones,
```

```
                LaunchTemplate={
                    "LaunchTemplateName": launch_template_name,
                    "Version": "$Default",
                },
                MinSize=min_size,
                MaxSize=max_size,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create group %s. Here's why: %s: %s",
                group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie CreateAutoScalingGroupin AWS SDK for Python (Boto3) API Reference.

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
async fn create_group(client: &Client, name: &str, id: &str) -> Result<(), Error>
 {
    client
        .create_auto_scaling_group()
        .auto_scaling_group_name(name)
        .instance_id(id)
        .min_size(1)
        .max_size(5)
        .send()
        .await?;
```

```
    println!("Created AutoScaling group");

    Ok(())
}
```

- Einzelheiten zur API finden Sie CreateAutoScalingGroupin der API-Referenz zum AWS SDK für Rust.

Beispiele, die Sie beim Erstellen von Gruppen mit gemischten Instanzen verwenden können, finden Sie in den folgenden Ressourcen.

- AWS SDK for .NET
- AWS SDK for Go
- AWS SDK für JavaScript
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

Aktualisieren Sie eine Auto Scaling Scaling-Gruppe mithilfe eines AWS SDK

Die folgenden Codebeispiele zeigen, wie man es benutzt`UpdateAutoScalingGroup`.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Update the capacity of an Auto Scaling group.
    /// </summary>
```

```
    /// <param name="groupName">The name of the Auto Scaling group.</param>
    /// <param name="launchTemplateName">The name of the EC2 launch template.</
param>
    /// <param name="maxSize">The maximum number of instances that can be
    /// created for the Auto Scaling group.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> UpdateAutoScalingGroupAsync(
        string groupName,
        string launchTemplateName,
        int maxSize)
    {
        var templateSpecification = new LaunchTemplateSpecification
        {
            LaunchTemplateName = launchTemplateName,
        };

        var groupRequest = new UpdateAutoScalingGroupRequest
        {
            MaxSize = maxSize,
            AutoScalingGroupName = groupName,
            LaunchTemplate = templateSpecification,
        };

        var response = await
 _amazonAutoScaling.UpdateAutoScalingGroupAsync(groupRequest);
        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"You successfully updated the Auto Scaling group
 {groupName}.");
            return true;
        }
        else
        {
            return false;
        }
    }
```

- Einzelheiten zur API finden Sie UpdateAutoScalingGroupin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::AutoScaling::AutoScalingClient autoScalingClient(clientConfig);

        Aws::AutoScaling::Model::UpdateAutoScalingGroupRequest request;
        request.SetAutoScalingGroupName(groupName);
        request.SetMaxSize(3);

        Aws::AutoScaling::Model::UpdateAutoScalingGroupOutcome outcome =
                autoScalingClient.UpdateAutoScalingGroup(request);

        if (!outcome.IsSuccess()) {
            std::cerr << "Error with AutoScaling::UpdateAutoScalingGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;

        }
```

- Einzelheiten zur API finden Sie UpdateAutoScalingGroupin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Beispiel 1: So aktualisieren Sie die Größenbeschränkungen einer Auto Scaling Scaling-Gruppe

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe mit einer Mindestgröße von 2 und einer Maximalgröße von 10 aktualisiert.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --min-size 2 \
    --max-size 10
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Festlegen von Kapazitätsgrenzen für Ihre Auto Scaling Scaling-Gruppe im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: So fügen Sie Elastic Load Balancing Health Checks hinzu und geben an, welche Availability Zones und Subnetze verwendet werden sollen

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe aktualisiert, um Elastic Load Balancing Health Checks hinzuzufügen. Dieser Befehl aktualisiert auch den Wert von `--vpc-zone-identifier` mit einer Liste von Subnetz-IDs in mehreren Availability Zones.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --health-check-type ELB \
    --health-check-grace-period 600 \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Elastic Load Balancing und Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.

Beispiel 3: Um die Platzierungsgruppe und die Kündigungsrichtlinie zu aktualisieren

In diesem Beispiel werden die zu verwendende Platzierungsgruppe und die Kündigungsrichtlinie aktualisiert.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --placement-group my-placement-group \
    --termination-policies "OldestInstance"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zu Auto-Scaling-Gruppen finden Sie im Benutzerhandbuch für Amazon EC2 Auto Scaling.

Beispiel 4: Um die neueste Version der Startvorlage zu verwenden

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe aktualisiert, sodass sie die neueste Version der angegebenen Startvorlage verwendet.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Startvorlagen im Benutzerhandbuch für Amazon EC2 Auto Scaling.

Beispiel 5: Um eine bestimmte Version der Startvorlage zu verwenden

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe so aktualisiert, dass sie eine bestimmte Version einer Startvorlage anstelle der neuesten Version oder Standardversion verwendet.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Startvorlagen im Benutzerhandbuch für Amazon EC2 Auto Scaling.

Beispiel 6: Um eine Richtlinie für gemischte Instanzen zu definieren und einen Kapazitätsausgleich zu ermöglichen

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe so aktualisiert, dass sie eine Richtlinie für gemischte Instanzen verwendet, und ermöglicht einen Kapazitätsausgleich. Mit dieser Struktur können Sie Gruppen mit Spot- und On-Demand-Kapazitäten angeben und unterschiedliche Startvorlagen für unterschiedliche Architekturen verwenden.

```
aws autoscaling update-auto-scaling-group \
```

```
    --cli-input-json file://~/config.json
```

Inhalt von `config.json`:

```
{
    "AutoScalingGroupName": "my-asg",
    "CapacityRebalance": true,
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template-for-x86",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "c6g.large",
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateName": "my-launch-template-for-arm",
                        "Version": "$Latest"
                    }
                },
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 50,
            "SpotAllocationStrategy": "capacity-optimized"
        }
    }
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen im Amazon EC2 Auto Scaling-Benutzerhandbuch.

- Einzelheiten zur API finden Sie UpdateAutoScalingGroupin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void updateAutoScalingGroup(AutoScalingClient
autoScalingClient, String groupName,
           String launchTemplateName) {
       try {
           AutoScalingWaiter waiter = autoScalingClient.waiter();
           LaunchTemplateSpecification templateSpecification =
LaunchTemplateSpecification.builder()
                   .launchTemplateName(launchTemplateName)
                   .build();

           UpdateAutoScalingGroupRequest groupRequest =
UpdateAutoScalingGroupRequest.builder()
                   .maxSize(3)
                   .autoScalingGroupName(groupName)
                   .launchTemplate(templateSpecification)
                   .build();

           autoScalingClient.updateAutoScalingGroup(groupRequest);
           DescribeAutoScalingGroupsRequest groupsRequest =
DescribeAutoScalingGroupsRequest.builder()
                   .autoScalingGroupNames(groupName)
                   .build();

           WaiterResponse<DescribeAutoScalingGroupsResponse> waiterResponse =
waiter
                   .waitUntilGroupInService(groupsRequest);
           waiterResponse.matched().response().ifPresent(System.out::println);
           System.out.println("You successfully updated the auto scaling group
" + groupName);

       } catch (AutoScalingException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
```

```
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie UpdateAutoScalingGroupin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun updateAutoScalingGroup(
    groupName: String,
    launchTemplateNameVal: String,
    serviceLinkedRoleARNVal: String
) {
    val templateSpecification =
        LaunchTemplateSpecification {
            launchTemplateName = launchTemplateNameVal
        }

    val groupRequest =
        UpdateAutoScalingGroupRequest {
            maxSize = 3
            serviceLinkedRoleArn = serviceLinkedRoleARNVal
            autoScalingGroupName = groupName
            launchTemplate = templateSpecification
        }

    val groupsRequestWaiter =
        DescribeAutoScalingGroupsRequest {
            autoScalingGroupNames = listOf(groupName)
        }
```

```
    AutoScalingClient { region = "us-east-1" }.use { autoScalingClient ->
        autoScalingClient.updateAutoScalingGroup(groupRequest)
        autoScalingClient.waitUntilGroupExists(groupsRequestWaiter)
        println("You successfully updated the Auto Scaling group  $groupName")
    }
}
```

- Einzelheiten zur API finden Sie UpdateAutoScalingGroupin der API-Referenz zum AWS SDK für Kotlin.

PHP

SDK für PHP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public function updateAutoScalingGroup($autoScalingGroupName, $args)
    {
        if (array_key_exists('MaxSize', $args)) {
            $maxSize = ['MaxSize' => $args['MaxSize']];
        } else {
            $maxSize = [];
        }
        if (array_key_exists('MinSize', $args)) {
            $minSize = ['MinSize' => $args['MinSize']];
        } else {
            $minSize = [];
        }
        $parameters = ['AutoScalingGroupName' => $autoScalingGroupName];
        $parameters = array_merge($parameters, $minSize, $maxSize);
        return $this->autoScalingClient->updateAutoScalingGroup($parameters);
    }
```

- Einzelheiten zur API finden Sie UpdateAutoScalingGroupin der AWS SDK for PHP API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Mindest- und Höchstgröße der angegebenen Auto Scaling Scaling-Gruppe aktualisiert.

```
Update-ASAutoScalingGroup -AutoScalingGroupName my-asg -MaxSize 5 -MinSize 1
```

Beispiel 2: In diesem Beispiel wird die Standard-Abklingzeit der angegebenen Auto Scaling Scaling-Gruppe aktualisiert.

```
Update-ASAutoScalingGroup -AutoScalingGroupName my-asg -DefaultCooldown 10
```

Beispiel 3: In diesem Beispiel werden die Availability Zones der angegebenen Auto Scaling Scaling-Gruppe aktualisiert.

```
Update-ASAutoScalingGroup -AutoScalingGroupName my-asg -AvailabilityZone @("us-
west-2a", "us-west-2b")
```

Beispiel 4: In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe aktualisiert, sodass sie Elastic Load Balancing Health Checks verwendet.

```
Update-ASAutoScalingGroup -AutoScalingGroupName my-asg -HealthCheckType ELB -
HealthCheckGracePeriod 60
```

- Einzelheiten zur API finden Sie unter UpdateAutoScalingGroup AWS Tools for PowerShellCmdlet-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class AutoScalingWrapper:
    """Encapsulates Amazon EC2 Auto Scaling actions."""

    def __init__(self, autoscaling_client):
        """
        :param autoscaling_client: A Boto3 Amazon EC2 Auto Scaling client.
        """
        self.autoscaling_client = autoscaling_client


    def update_group(self, group_name, **kwargs):
        """
        Updates an Auto Scaling group.

        :param group_name: The name of the group to update.
        :param kwargs: Keyword arguments to pass through to the service.
        """
        try:
            self.autoscaling_client.update_auto_scaling_group(
                AutoScalingGroupName=group_name, **kwargs
            )
        except ClientError as err:
            logger.error(
                "Couldn't update group %s. Here's why: %s: %s",
                group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie UpdateAutoScalingGroupin AWS SDK for Python (Boto3) API Reference.

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
async fn update_group(client: &Client, name: &str, size: i32) -> Result<(),
 Error> {
    client
        .update_auto_scaling_group()
        .auto_scaling_group_name(name)
        .max_size(size)
        .send()
        .await?;

    println!("Updated AutoScaling group");

    Ok(())
}
```

- Einzelheiten zur API finden Sie UpdateAutoScalingGroupin der API-Referenz zum AWS SDK für Rust.

Beschreiben Sie eine Auto Scaling Scaling-Gruppe mithilfe eines AWS SDK

Die folgenden Codebeispiele zeigen die Verwendung`DescribeAutoScalingGroups`.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Get data about the instances in an Amazon EC2 Auto Scaling group.
    /// </summary>
    /// <param name="groupName">The name of the Amazon EC2 Auto Scaling group.</
param>
    /// <returns>A list of Amazon EC2 Auto Scaling details.</returns>
    public async Task<List<AutoScalingInstanceDetails>>
 DescribeAutoScalingInstancesAsync(
        string groupName)
    {
        var groups = await DescribeAutoScalingGroupsAsync(groupName);
        var instanceIds = new List<string>();
        groups!.ForEach(group =>
        {
            if (group.AutoScalingGroupName == groupName)
            {
                group.Instances.ForEach(instance =>
                {
                    instanceIds.Add(instance.InstanceId);
                });
            }
        });

        var scalingGroupsRequest = new DescribeAutoScalingInstancesRequest
        {
            MaxRecords = 10,
            InstanceIds = instanceIds,
        };

        var response = await
 _amazonAutoScaling.DescribeAutoScalingInstancesAsync(scalingGroupsRequest);
        var instanceDetails = response.AutoScalingInstances;

        return instanceDetails;
    }
```

- Einzelheiten zur API finden Sie DescribeAutoScalingGroupsin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::AutoScaling::AutoScalingClient autoScalingClient(clientConfig);

Aws::AutoScaling::Model::DescribeAutoScalingGroupsRequest request;
Aws::Vector<Aws::String> groupNames;
groupNames.push_back(groupName);
request.SetAutoScalingGroupNames(groupNames);

Aws::AutoScaling::Model::DescribeAutoScalingGroupsOutcome outcome =
        client.DescribeAutoScalingGroups(request);

if (outcome.IsSuccess()) {
    autoScalingGroup = outcome.GetResult().GetAutoScalingGroups();
}
else {
    std::cerr << "Error with AutoScaling::DescribeAutoScalingGroups. "
              << outcome.GetError().GetMessage()
              << std::endl;
}
```

- Einzelheiten zur API finden Sie DescribeAutoScalingGroupsin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Beispiel 1: Um die angegebene Auto Scaling Scaling-Gruppe zu beschreiben

Dieses Beispiel beschreibt die angegebene Auto Scaling Scaling-Gruppe.

```
aws autoscaling describe-auto-scaling-groups \
    --auto-scaling-group-name my-asg
```

Ausgabe:

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-
a11a-7b0acd480f03:autoScalingGroupName/my-asg",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-1234567890abcde12"
            },
            "MinSize": 0,
            "MaxSize": 1,
            "DesiredCapacity": 1,
            "DefaultCooldown": 300,
            "AvailabilityZones": [
                "us-west-2a",
                "us-west-2b",
                "us-west-2c"
            ],
            "LoadBalancerNames": [],
            "TargetGroupARNs": [],
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": 0,
            "Instances": [
                {
                    "InstanceId": "i-06905f55584de02da",
                    "InstanceType": "t2.micro",
                    "AvailabilityZone": "us-west-2a",
```

```
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService",
                    "ProtectedFromScaleIn": false,
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-1234567890abcde12"
                    }
                }
            ],
            "CreatedTime": "2023-10-28T02:39:22.152Z",
            "SuspendedProcesses": [],
            "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-
c934b782",
            "EnabledMetrics": [],
            "Tags": [],
            "TerminationPolicies": [
                "Default"
            ],
            "NewInstancesProtectedFromScaleIn": false,
            "ServiceLinkedRoleARN":"arn",
            "TrafficSources": []
        }
    ]
}
```

Beispiel 2: Um die ersten 100 angegebenen Auto Scaling Scaling-Gruppe zu beschreiben

In diesem Beispiel werden die angegebenen Auto Scaling Scaling-Gruppen beschrieben. Es ermöglicht Ihnen, bis zu 100 Gruppennamen anzugeben.

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 100 \
    --auto-scaling-group-name "group1" "group2" "group3" "group4"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Beispiel 3: Um eine Auto Scaling Scaling-Gruppe in der angegebenen Region zu beschreiben

Dieses Beispiel beschreibt die Auto Scaling Scaling-Gruppen in der angegebenen Region, bis zu einem Maximum von 75 Gruppen.

```
aws autoscaling describe-auto-scaling-groups \
```

```
    --max-items 75 \
    --region us-east-1
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Beispiel 4: Um die angegebene Anzahl von Auto Scaling Scaling-Gruppen zu beschreiben

Um eine bestimmte Anzahl von Auto Scaling Scaling-Gruppen zurückzugeben, verwenden Sie die `--max-items` Option.

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 1
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Wenn die Ausgabe ein `NextToken` Feld enthält, gibt es mehr Gruppen. Um die zusätzlichen Gruppen abzurufen, verwenden Sie den Wert dieses Felds mit der `--starting-token` Option in einem nachfolgenden Aufruf wie folgt.

```
aws autoscaling describe-auto-scaling-groups \
    --starting-token Z3M3LMPEXAMPLE
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Beispiel 5: Um Auto Scaling Scaling-Gruppen zu beschreiben, die Startkonfigurationen verwenden

In diesem Beispiel wird die `--query` Option verwendet, um Auto Scaling Scaling-Gruppen zu beschreiben, die Startkonfigurationen verwenden.

```
aws autoscaling describe-auto-scaling-groups \
    --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

Ausgabe:

```
[
    {
        "AutoScalingGroupName": "my-asg",
        "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-
a11a-7b0acd480f03:autoScalingGroupName/my-asg",
```

```
        "LaunchConfigurationName": "my-lc",
        "MinSize": 0,
        "MaxSize": 1,
        "DesiredCapacity": 1,
        "DefaultCooldown": 300,
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2b",
            "us-west-2c"
        ],
        "LoadBalancerNames": [],
        "TargetGroupARNs": [],
        "HealthCheckType": "EC2",
        "HealthCheckGracePeriod": 0,
        "Instances": [
            {
                "InstanceId": "i-088c57934a6449037",
                "InstanceType": "t2.micro",
                "AvailabilityZone": "us-west-2c",
                "HealthStatus": "Healthy",
                "LifecycleState": "InService",
                "LaunchConfigurationName": "my-lc",
                "ProtectedFromScaleIn": false
            }
        ],
        "CreatedTime": "2023-10-28T02:39:22.152Z",
        "SuspendedProcesses": [],
        "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
        "EnabledMetrics": [],
        "Tags": [],
        "TerminationPolicies": [
            "Default"
        ],
        "NewInstancesProtectedFromScaleIn": false,
        "ServiceLinkedRoleARN":"arn",
        "TrafficSources": []
    }
]
```


Weitere Informationen finden Sie unter AWS CLI-Ausgabe filtern im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

- Einzelheiten zur API finden Sie DescribeAutoScalingGroupsunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.autoscaling.AutoScalingClient;
import software.amazon.awssdk.services.autoscaling.model.AutoScalingException;
import software.amazon.awssdk.services.autoscaling.model.AutoScalingGroup;
import
 software.amazon.awssdk.services.autoscaling.model.DescribeAutoScalingGroupsResponse;
import
 software.amazon.awssdk.services.autoscaling.model.DescribeAutoScalingGroupsRequest;
import software.amazon.awssdk.services.autoscaling.model.Instance;
import java.util.List;

/**
 * Before running this SDK for Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeAutoScalingInstances {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <groupName>

                Where:
                    groupName - The name of the Auto Scaling group.
                """;

        if (args.length != 1) {
```

```
            System.out.println(usage);
            System.exit(1);
        }

        String groupName = args[0];
        AutoScalingClient autoScalingClient = AutoScalingClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String instanceId = getAutoScaling(autoScalingClient, groupName);
        System.out.println(instanceId);
        autoScalingClient.close();
    }

    public static String getAutoScaling(AutoScalingClient autoScalingClient,
String groupName) {
        try {
            String instanceId = "";
            DescribeAutoScalingGroupsRequest scalingGroupsRequest =
DescribeAutoScalingGroupsRequest.builder()
                    .autoScalingGroupNames(groupName)
                    .build();

            DescribeAutoScalingGroupsResponse response = autoScalingClient
                    .describeAutoScalingGroups(scalingGroupsRequest);
            List<AutoScalingGroup> groups = response.autoScalingGroups();
            for (AutoScalingGroup group : groups) {
                System.out.println("The group name is " +
group.autoScalingGroupName());
                System.out.println("The group ARN is " +
group.autoScalingGroupARN());

                List<Instance> instances = group.instances();
                for (Instance instance : instances) {
                    instanceId = instance.instanceId();
                }
            }
            return instanceId;
        } catch (AutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

```
}
```

- Einzelheiten zur API finden Sie DescribeAutoScalingGroupsin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun getAutoScalingGroups(groupName: String) {
    val scalingGroupsRequest =
        DescribeAutoScalingGroupsRequest {
            autoScalingGroupNames = listOf(groupName)
        }

    AutoScalingClient { region = "us-east-1" }.use { autoScalingClient ->
        val response =
 autoScalingClient.describeAutoScalingGroups(scalingGroupsRequest)
        response.autoScalingGroups?.forEach { group ->
            println("The group name is ${group.autoScalingGroupName}")
            println("The group ARN is ${group.autoScalingGroupArn}")
            group.instances?.forEach { instance ->
                println("The instance id is ${instance.instanceId}")
                println("The lifecycle state is " + instance.lifecycleState)
            }
        }
    }
}
```

- Einzelheiten zur API finden Sie DescribeAutoScalingGroupsin der API-Referenz zum AWS SDK für Kotlin.

PHP

SDK für PHP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public function describeAutoScalingGroups($autoScalingGroupNames)
    {
        return $this->autoScalingClient->describeAutoScalingGroups([
            'AutoScalingGroupNames' => $autoScalingGroupNames
        ]);
    }
```

- Einzelheiten zur API finden Sie DescribeAutoScalingGroupsin der AWS SDK for PHP API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Namen Ihrer Auto Scaling Scaling-Gruppen aufgeführt.

```
Get-ASAutoScalingGroup | format-table -property AutoScalingGroupName
```

Ausgabe:

```
AutoScalingGroupName
--------------------
my-asg-1
my-asg-2
my-asg-3
my-asg-4
my-asg-5
my-asg-6
```

Beispiel 2: Dieses Beispiel beschreibt die angegebene Auto Scaling Scaling-Gruppe.

```
Get-ASAutoScalingGroup -AutoScalingGroupName my-asg-1
```

Ausgabe:

```
AutoScalingGroupARN     : arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-a11a-7b0acd480
                          f03:autoScalingGroupName/my-asg-1
AutoScalingGroupName    : my-asg-1
AvailabilityZones       : {us-west-2b, us-west-2a}
CreatedTime             : 3/1/2015 9:05:31 AM
DefaultCooldown         : 300
DesiredCapacity         : 2
EnabledMetrics          : {}
HealthCheckGracePeriod  : 300
HealthCheckType         : EC2
Instances               : {my-lc}
LaunchConfigurationName : my-lc
LoadBalancerNames       : {}
MaxSize                 : 0
MinSize                 : 0
PlacementGroup          :
Status                  :
SuspendedProcesses      : {}
Tags                    : {}
TerminationPolicies     : {Default}
VPCZoneIdentifier       : subnet-e4f33493,subnet-5264e837
```

Beispiel 3: Dieses Beispiel beschreibt die angegebenen zwei Auto Scaling Scaling-Gruppen.

```
Get-ASAutoScalingGroup -AutoScalingGroupName @("my-asg-1", "my-asg-2")
```

Beispiel 4: Dieses Beispiel beschreibt die Auto Scaling Scaling-Instances für die angegebene Auto Scaling Scaling-Gruppe.

```
(Get-ASAutoScalingGroup -AutoScalingGroupName my-asg-1).Instances
```

Beispiel 5: Dieses Beispiel beschreibt alle Ihre Auto Scaling Scaling-Gruppen.

```
Get-ASAutoScalingGroup
```

Beispiel 6: In diesem Beispiel werden alle Ihre Auto Scaling Scaling-Gruppen in Batches von 10 beschrieben.

```
$nextToken = $null
do {
  Get-ASAutoScalingGroup -NextToken $nextToken -MaxRecord 10
  $nextToken = $AWSHistory.LastServiceResponse.NextToken
} while ($nextToken -ne $null)
```

Beispiel 7: Dieses LaunchTemplate Beispiel beschreibt die angegebene Auto Scaling Scaling-Gruppe. In diesem Beispiel wird davon ausgegangen, dass die Option „Instance-Kaufoptionen“ auf „An der Startvorlage festhalten“ gesetzt ist. Falls diese Option auf „Kaufoptionen und Instanztypen kombinieren“ gesetzt ist, LaunchTemplate kann über „MixedInstancesPolicy“ darauf zugegriffen werden. LaunchTemplate„Eigenschaft.

```
(Get-ASAutoScalingGroup -AutoScalingGroupName my-ag-1).LaunchTemplate
```

Ausgabe:

```
LaunchTemplateId     LaunchTemplateName   Version
----------------     ------------------   -------
lt-06095fd619cb40371 test-launch-template $Default
```

- Einzelheiten zur API finden Sie unter DescribeAutoScalingGroups AWS Tools for PowerShellCmdlet-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class AutoScalingWrapper:
    """Encapsulates Amazon EC2 Auto Scaling actions."""
```

```
    def __init__(self, autoscaling_client):
        """
        :param autoscaling_client: A Boto3 Amazon EC2 Auto Scaling client.
        """
        self.autoscaling_client = autoscaling_client


    def describe_group(self, group_name):
        """
        Gets information about an Auto Scaling group.

        :param group_name: The name of the group to look up.
        :return: Information about the group, if found.
        """
        try:
            response = self.autoscaling_client.describe_auto_scaling_groups(
                AutoScalingGroupNames=[group_name]
            )
        except ClientError as err:
            logger.error(
                "Couldn't describe group %s. Here's why: %s: %s",
                group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            groups = response.get("AutoScalingGroups", [])
            return groups[0] if len(groups) > 0 else None
```

- Einzelheiten zur API finden Sie DescribeAutoScalingGroupsin AWS SDK for Python (Boto3) API Reference.

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
async fn list_groups(client: &Client) -> Result<(), Error> {
    let resp = client.describe_auto_scaling_groups().send().await?;

    println!("Groups:");

    let groups = resp.auto_scaling_groups();

    for group in groups {
        println!(
            "Name:  {}",
            group.auto_scaling_group_name().unwrap_or("Unknown")
        );
        println!(
            "Arn:   {}",
            group.auto_scaling_group_arn().unwrap_or("unknown"),
        );
        println!("Zones: {:?}", group.availability_zones(),);
        println!();
    }

    println!("Found {} group(s)", groups.len());

    Ok(())
}
```

- Einzelheiten zur API finden Sie DescribeAutoScalingGroupsin der API-Referenz zum AWS SDK für Rust.

Löschen Sie eine Auto Scaling Scaling-Gruppe mithilfe eines AWS SDK

Die folgenden Codebeispiele zeigen, wie man es benutzt`DeleteAutoScalingGroup`.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Aktualisieren Sie die Mindestgröße einer Auto-Scaling-Gruppe auf Null, beenden Sie alle Instances in der Gruppe und löschen Sie die Gruppe.

```
    /// <summary>
    /// Try to terminate an instance by its Id.
    /// </summary>
    /// <param name="instanceId">The Id of the instance to terminate.</param>
    /// <returns>Async task.</returns>
    public async Task TryTerminateInstanceById(string instanceId)
    {
        var stopping = false;
        Console.WriteLine($"Stopping {instanceId}...");
        while (!stopping)
        {
            try
            {
                await
_amazonAutoScaling.TerminateInstanceInAutoScalingGroupAsync(
                    new TerminateInstanceInAutoScalingGroupRequest()
                    {
                        InstanceId = instanceId,
                        ShouldDecrementDesiredCapacity = false
                    });
                stopping = true;
            }
            catch (ScalingActivityInProgressException)
            {
```

```
                Console.WriteLine($"Scaling activity in progress for
{instanceId}. Waiting...");
                Thread.Sleep(10000);
            }
        }
    }

    /// <summary>
    /// Tries to delete the EC2 Auto Scaling group. If the group is in use or in
progress,
    /// waits and retries until the group is successfully deleted.
    /// </summary>
    /// <param name="groupName">The name of the group to try to delete.</param>
    /// <returns>Async task.</returns>
    public async Task TryDeleteGroupByName(string groupName)
    {
        var stopped = false;
        while (!stopped)
        {
            try
            {
                await _amazonAutoScaling.DeleteAutoScalingGroupAsync(
                    new DeleteAutoScalingGroupRequest()
                    {
                        AutoScalingGroupName = groupName
                    });
                stopped = true;
            }
            catch (Exception e)
                when ((e is ScalingActivityInProgressException)
                      || (e is Amazon.AutoScaling.Model.ResourceInUseException))
            {
                Console.WriteLine($"Some instances are still running.
Waiting...");
                Thread.Sleep(10000);
            }
        }
    }

    /// <summary>
    /// Terminate instances and delete the Auto Scaling group by name.
    /// </summary>
    /// <param name="groupName">The name of the group to delete.</param>
    /// <returns>Async task.</returns>
```

```
    public async Task TerminateAndDeleteAutoScalingGroupWithName(string
 groupName)
    {
        var describeGroupsResponse = await
 _amazonAutoScaling.DescribeAutoScalingGroupsAsync(
            new DescribeAutoScalingGroupsRequest()
            {
                AutoScalingGroupNames = new List<string>() { groupName }
            });
        if (describeGroupsResponse.AutoScalingGroups.Any())
        {
            // Update the size to 0.
            await _amazonAutoScaling.UpdateAutoScalingGroupAsync(
                new UpdateAutoScalingGroupRequest()
                {
                    AutoScalingGroupName = groupName,
                    MinSize = 0
                });
            var group = describeGroupsResponse.AutoScalingGroups[0];
            foreach (var instance in group.Instances)
            {
                await TryTerminateInstanceById(instance.InstanceId);
            }

            await TryDeleteGroupByName(groupName);
        }
        else
        {
            Console.WriteLine($"No groups found with name {groupName}.");
        }
    }
```

```
    /// <summary>
    /// Delete an Auto Scaling group.
    /// </summary>
    /// <param name="groupName">The name of the Amazon EC2 Auto Scaling group.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteAutoScalingGroupAsync(
        string groupName)
    {
```

```
        var deleteAutoScalingGroupRequest = new DeleteAutoScalingGroupRequest
        {
            AutoScalingGroupName = groupName,
            ForceDelete = true,
        };

        var response = await
_amazonAutoScaling.DeleteAutoScalingGroupAsync(deleteAutoScalingGroupRequest);
        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"You successfully deleted {groupName}");
            return true;
        }

        Console.WriteLine($"Couldn't delete {groupName}.");
        return false;
    }
```

- Einzelheiten zur API finden Sie DeleteAutoScalingGroupin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::AutoScaling::AutoScalingClient autoScalingClient(clientConfig);

            Aws::AutoScaling::Model::DeleteAutoScalingGroupRequest request;
            request.SetAutoScalingGroupName(groupName);
```

```
            Aws::AutoScaling::Model::DeleteAutoScalingGroupOutcome outcome =
                    autoScalingClient.DeleteAutoScalingGroup(request);

            if (outcome.IsSuccess()) {
                std::cout << "Auto Scaling group '" << groupName << "' was
 deleted."
                          << std::endl;
            }
            else {
                std::cerr << "Error with AutoScaling::DeleteAutoScalingGroup. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                result = false;
            }
        }
```

- Einzelheiten zur API finden Sie DeleteAutoScalingGroupin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Beispiel 1: Um die angegebene Auto Scaling Scaling-Gruppe zu löschen

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe gelöscht.

```
aws autoscaling delete-auto-scaling-group \
    --auto-scaling-group-name my-asg
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Löschen Ihrer Auto Scaling Scaling-Infrastruktur im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: So erzwingen Sie das Löschen der angegebenen Auto Scaling Scaling-Gruppe

Verwenden Sie die `--force-delete` Option, um die Auto Scaling Scaling-Gruppe zu löschen, ohne darauf zu warten, dass die Instances in der Gruppe beendet werden.

```
aws autoscaling delete-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --force-delete
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Löschen Ihrer Auto Scaling Scaling-Infrastruktur im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DeleteAutoScalingGroup AWS CLIBefehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.autoscaling.AutoScalingClient;
import software.amazon.awssdk.services.autoscaling.model.AutoScalingException;
import
 software.amazon.awssdk.services.autoscaling.model.DeleteAutoScalingGroupRequest;

/**
 * Before running this SDK for Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteAutoScalingGroup {
    public static void main(String[] args) {
        final String usage = """

                Usage:
```

```
                    <groupName>

                Where:
                    groupName - The name of the Auto Scaling group.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String groupName = args[0];
        AutoScalingClient autoScalingClient = AutoScalingClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deleteAutoScalingGroup(autoScalingClient, groupName);
        autoScalingClient.close();
    }

    public static void deleteAutoScalingGroup(AutoScalingClient
 autoScalingClient, String groupName) {
        try {
            DeleteAutoScalingGroupRequest deleteAutoScalingGroupRequest =
 DeleteAutoScalingGroupRequest.builder()
                    .autoScalingGroupName(groupName)
                    .forceDelete(true)
                    .build();


 autoScalingClient.deleteAutoScalingGroup(deleteAutoScalingGroupRequest);
            System.out.println("You successfully deleted " + groupName);

        } catch (AutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie DeleteAutoScalingGroupin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun deleteSpecificAutoScalingGroup(groupName: String) {
    val deleteAutoScalingGroupRequest =
        DeleteAutoScalingGroupRequest {
            autoScalingGroupName = groupName
            forceDelete = true
        }

    AutoScalingClient { region = "us-east-1" }.use { autoScalingClient ->
        autoScalingClient.deleteAutoScalingGroup(deleteAutoScalingGroupRequest)
        println("You successfully deleted $groupName")
    }
}
```

- Einzelheiten zur API finden Sie DeleteAutoScalingGroupin der API-Referenz zum AWS SDK für Kotlin.

PHP

SDK für PHP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public function deleteAutoScalingGroup($autoScalingGroupName)
    {
        return $this->autoScalingClient->deleteAutoScalingGroup([
```

```
            'AutoScalingGroupName' => $autoScalingGroupName,
            'ForceDelete' => true,
        ]);
    }
```

- Einzelheiten zur API finden Sie DeleteAutoScalingGroupin der AWS SDK for PHP API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe gelöscht, wenn sie keine laufenden Instances hat. Sie werden zur Bestätigung aufgefordert, bevor der Vorgang fortgesetzt wird.

```
Remove-ASAutoScalingGroup -AutoScalingGroupName my-asg
```

Ausgabe:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-ASAutoScalingGroup (DeleteAutoScalingGroup)" on
 Target "my-asg".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"):
```

Beispiel 2: Wenn Sie den Force-Parameter angeben, werden Sie nicht zur Bestätigung aufgefordert, bevor der Vorgang fortgesetzt wird.

```
Remove-ASAutoScalingGroup -AutoScalingGroupName my-asg -Force
```

Beispiel 3: In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe gelöscht und alle laufenden Instances, die sie enthält, beendet.

```
Remove-ASAutoScalingGroup -AutoScalingGroupName my-asg -ForceDelete $true -Force
```

- Einzelheiten zur API finden Sie unter DeleteAutoScalingGroup AWS Tools for PowerShellCmdlet-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Aktualisieren Sie die Mindestgröße einer Auto-Scaling-Gruppe auf Null, beenden Sie alle Instances in der Gruppe und löschen Sie die Gruppe.

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
 created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
 that is
                          created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
        self.ami_param = ami_param
```

```
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        self.launch_template_name = f"{resource_prefix}-template"
        self.group_name = f"{resource_prefix}-group"
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
        self.instance_profile_name = f"{resource_prefix}-prof"
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
        self.bad_creds_role_name = f"{resource_prefix}-bc-role"
        self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
        self.key_pair_name = f"{resource_prefix}-key-pair"


    def _try_terminate_instance(self, inst_id):
        stopping = False
        log.info(f"Stopping {inst_id}.")
        while not stopping:
            try:
                self.autoscaling_client.terminate_instance_in_auto_scaling_group(
                    InstanceId=inst_id, ShouldDecrementDesiredCapacity=True
                )
                stopping = True
            except ClientError as err:
                if err.response["Error"]["Code"] == "ScalingActivityInProgress":
                    log.info("Scaling activity in progress for %s. Waiting...",
inst_id)
                    time.sleep(10)
                else:
                    raise AutoScalerError(f"Couldn't stop instance {inst_id}:
{err}.")

    def _try_delete_group(self):
        """
        Tries to delete the EC2 Auto Scaling group. If the group is in use or in
progress,
        the function waits and retries until the group is successfully deleted.
        """
        stopped = False
        while not stopped:
            try:
                self.autoscaling_client.delete_auto_scaling_group(
                    AutoScalingGroupName=self.group_name
```

```
                )
                stopped = True
                log.info("Deleted EC2 Auto Scaling group %s.", self.group_name)
            except ClientError as err:
                if (
                    err.response["Error"]["Code"] == "ResourceInUse"
                    or err.response["Error"]["Code"] ==
 "ScalingActivityInProgress"
                ):
                    log.info(
                        "Some instances are still running. Waiting for them to
 stop..."
                    )
                    time.sleep(10)
                else:
                    raise AutoScalerError(
                        f"Couldn't delete group {self.group_name}: {err}."
                    )

    def delete_group(self):
        """
        Terminates all instances in the group, deletes the EC2 Auto Scaling
 group.
        """
        try:
            response = self.autoscaling_client.describe_auto_scaling_groups(
                AutoScalingGroupNames=[self.group_name]
            )
            groups = response.get("AutoScalingGroups", [])
            if len(groups) > 0:
                self.autoscaling_client.update_auto_scaling_group(
                    AutoScalingGroupName=self.group_name, MinSize=0
                )
                instance_ids = [inst["InstanceId"] for inst in groups[0]
["Instances"]]
                for inst_id in instance_ids:
                    self._try_terminate_instance(inst_id)
                self._try_delete_group()
            else:
                log.info("No groups found named %s, nothing to do.",
 self.group_name)
        except ClientError as err:
            raise AutoScalerError(f"Couldn't delete group {self.group_name}:
 {err}.")
```

- Einzelheiten zur API finden Sie DeleteAutoScalingGroupin AWS SDK for Python (Boto3) API Reference.

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
async fn delete_group(client: &Client, name: &str, force: bool) -> Result<(),
 Error> {
    client
        .delete_auto_scaling_group()
        .auto_scaling_group_name(name)
        .set_force_delete(if force { Some(true) } else { None })
        .send()
        .await?;

    println!("Deleted Auto Scaling group");

    Ok(())
}
```

- Einzelheiten zur API finden Sie DeleteAutoScalingGroupin der API-Referenz zum AWS SDK für Rust.

Recyceln der Instances in Ihrer Auto-Scaling-Gruppe

Amazon EC2 Auto Scaling bietet Funktionen, mit denen Sie die Amazon EC2 EC2-Instances in Ihrer Auto Scaling Scaling-Gruppe ersetzen können, nachdem Sie Aktualisierungen vorgenommen haben, wie z. B. das Hinzufügen einer neuen Startvorlage durch ein neues Amazon Machine Image (AMI) oder das Hinzufügen neuer Instance-Typen. Es hilft Ihnen auch dabei, Updates zu optimieren, indem es Ihnen die Möglichkeit gibt, sie in denselben Vorgang einzubeziehen, der die Instances ersetzt.

Dieser Abschnitt enthält Informationen, die Sie bei folgenden Aktionen unterstützen:

- Starten einer Instance-Aktualisierung, um Instances in Ihrer Auto-Scaling-Gruppe zu ersetzen.
- Deklarieren bestimmter Updates, die eine gewünschte Konfiguration beschreiben, und aktualisieren Sie die Auto-Scaling-Gruppe auf die gewünschte Konfiguration.
- Überspringen des Ersetzens bereits aktualisierter Instances.
- Verwenden Sie Checkpoints, um Instances phasenweise zu aktualisieren und Ihre Instances an bestimmten Punkten zu überprüfen.
- Erhalten von Benachrichtigungen per E-Mail, wenn ein Checkpoint erreicht ist.
- Verwenden Sie ein Rollback, um die zuvor verwendete Konfiguration der Auto-Scaling-Gruppe wiederherzustellen.
- Automatisches Rollback, wenn die Instance-Aktualisierung aus irgendeinem Grund fehlschlägt oder wenn von Ihnen angegebene CloudWatch Amazon-Alarme in den `ALARM` Status wechseln.
- Begrenzen Sie die Lebensdauer von Instances, um konsistente Softwareversionen und Instance-Konfigurationen in der gesamten Auto-Scaling-Gruppe bereitzustellen.

Inhalt

Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren

Sie können eine Instance-Aktualisierung verwenden, um die Instances in Ihrer Auto Scaling Scaling-Gruppe zu aktualisieren. Diese Funktion kann nützlich sein, wenn Sie aufgrund einer

Konfigurationsänderung Instances ersetzen müssen, insbesondere wenn Ihre Auto Scaling Scaling-Gruppe eine große Anzahl von Instances enthält.

Zu den Situationen, in denen eine Instanzaktualisierung hilfreich sein kann, gehören:

- Bereitstellung eines neuen Amazon Machine Image (AMI) oder Benutzerdatenskripts in Ihrer Auto Scaling Scaling-Gruppe. Sie können eine neue Startvorlage mit den Änderungen erstellen und dann eine Instance-Aktualisierung verwenden, um die Updates sofort bereitzustellen.
- Migrieren Sie Ihre Instances auf neue Instance-Typen, um von den neuesten Verbesserungen und Optimierungen zu profitieren.
- Umstellung Ihrer Auto Scaling Scaling-Gruppen von der Verwendung einer Startkonfiguration auf die Verwendung einer Startvorlage. Sie können Ihre Startkonfigurationen in Startvorlagen kopieren und dann eine Instance-Aktualisierung verwenden, um Ihre Instances auf die neuen Vorlagen zu aktualisieren. Weitere Informationen zur Migration zu Startvorlagen finden Sie unter Migrieren Sie Ihre Auto Scaling Scaling-Gruppen, um Vorlagen zu starten.

Inhalt

Wie funktioniert eine Instanzaktualisierung

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

Inhalt

- Frist der Zustandsprüfung
- Kompatibilität von Instance-Typen
- Einschränkungen

Funktionsweise

Um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren, können Sie eine neue Konfiguration definieren, die die neueste Version Ihrer Anwendung und alle anderen Updates, die Sie vornehmen möchten, enthält. Starten Sie dann eine Instanzaktualisierung, um bestehende Instances auf der Grundlage dieser Konfiguration durch neue zu ersetzen.

So führen Sie eine Instanzaktualisierung durch:

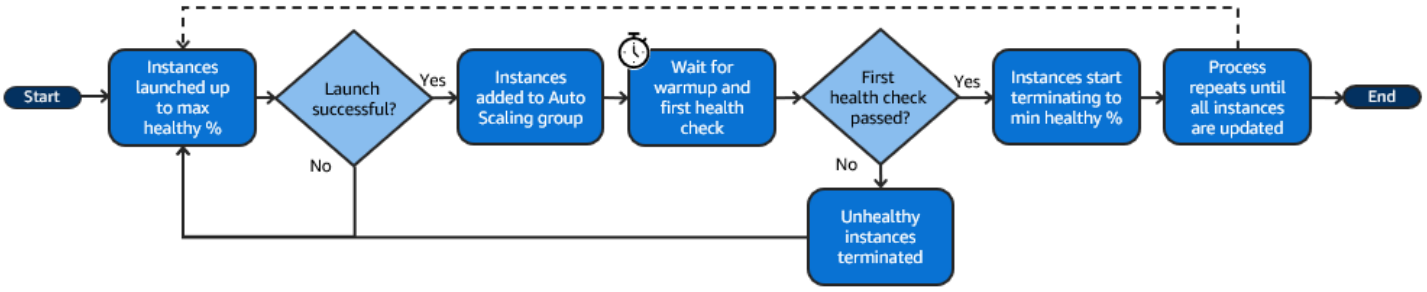
1. Erstellen Sie eine neue Startvorlage oder aktualisieren Sie die bestehende Vorlage mit den gewünschten Konfigurationsänderungen, z. B. einem neuen Amazon Machine Image (AMI). Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.
2. Starten Sie die Instance-Aktualisierung mit der Amazon EC2 Auto Scaling Scaling-Konsole oder dem SDK: AWS CLI
   - Geben Sie die neue Startvorlage oder die Version der Startvorlage an, die Sie erstellt haben. Dies wird verwendet, um neue Instances zu starten.
   - Legen Sie den bevorzugten Mindest- und Höchstwert für gesunde Werte fest. Dadurch wird gesteuert, wie viele Instances gleichzeitig ersetzt werden und ob neue Instances gestartet werden, bevor alte beendet werden.
   - Konfigurieren Sie alle optionalen Einstellungen, wie z. B.:
     - Checkpoints — Unterbrechen Sie die Aktualisierung der Instanz nach einem bestimmten Prozentsatz an Ersetzungen, um den Fortschritt zu überprüfen.
     - Zuordnung überspringen — Vergleichen Sie alte Instances mit der neuen Konfiguration und ersetzen Sie nur diejenigen, die nicht übereinstimmen. Wenn Sie eine Instanzaktualisierung von der Konsole aus starten, ist „Abgleich überspringen“ standardmäßig aktiviert.
     - Mehrere Instanztypen — Wenden Sie eine neue oder aktualisierte Richtlinie für gemischte Instanzen als Teil der gewünschten Konfiguration an.

Wenn die Instance-Aktualisierung gestartet wurde, wird Amazon EC2 Auto Scaling:

- Ersetzt Instances stapelweise auf der Grundlage der minimalen und maximalen fehlerfreien Werte.

- Starten Sie zuerst die neuen Instances, bevor Sie die alten beenden, wenn der Mindestprozentsatz für fehlerfreie Instances auf 100 Prozent festgelegt ist. Dadurch wird sichergestellt, dass Ihre gewünschte Kapazität jederzeit beibehalten wird.
- Überprüfen Sie den Integritätsstatus der Instances und geben Sie ihnen Zeit, sich aufzuwärmen, bevor weitere Instanzen ersetzt werden.
- Beenden und ersetzen Sie Instances, die sich als fehlerhaft erwiesen haben.
- Aktualisieren Sie die Auto Scaling Scaling-Gruppeneinstellungen automatisch mit den neuen Konfigurationsänderungen, nachdem die Instanzaktualisierung erfolgreich war.
- Ersetzen Sie `InService` Instanzen vor Instanzen, die sich in einem warmen Pool befinden.

Das folgende Flussdiagramm veranschaulicht das Verhalten beim Starten vor dem Beenden, wenn Sie den fehlerfreien Mindestwert auf 100 Prozent festlegen.



> **Note**
>
> Die Mindest- und Höchstwerte für einen fehlerfreien Zustand bei einer Instanzaktualisierung müssen nur angegeben werden, wenn Sie keine Instanzwartungsrichtlinie festgelegt haben oder wenn Sie die bestehende Richtlinie überschreiben müssen. Weitere Informationen finden Sie unter Wartungsrichtlinien für Instances.
> Ebenso müssen Sie den Instanz-Aufwärmzeitraum für eine Instanzaktualisierung nur angeben, wenn Sie den Standard-Warmup nicht aktiviert haben oder wenn Sie den Standard überschreiben müssen. Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

Schlüsselkonzepte

Bevor Sie beginnen, sollten Sie sich mit folgenden zentralen Konzepten der Instance-Aktualisierungen vertraut machen:

Minimaler fehlerfreier Prozentsatz

Der minimale fehlerfreie Prozentsatz ist der Prozentsatz der gewünschten Kapazität, die während einer Instanzaktualisierung in Betrieb, fehlerfrei und einsatzbereit bleiben soll, sodass die Aktualisierung fortgesetzt werden kann. Wenn der minimale fehlerfreie Prozentsatz beispielsweise 90 Prozent beträgt, und der maximale fehlerfreie Prozentsatz 100 Prozent beträgt, dann werden jeweils 10 Prozent der Kapazität ersetzt. Wenn die neuen Instances ihre Zustandsprüfungen nicht bestehen, beendet Amazon EC2 Auto Scaling sie und ersetzt diese. Wenn die Instance-Aktualisierung keine fehlerfreien Instances starten kann, wird sie schließlich fehlschlagen, und die anderen 90 Prozent der Gruppe bleiben unberührt. Wenn die neuen Instances fehlerfrei bleiben und ihre Aufwärmphase abgeschlossen haben, kann Amazon EC2 Auto Scaling weiterhin andere Instances ersetzen.

Eine Instance-Aktualisierung kann eine einzelne Instance, mehrere Instances auf einmal oder alle auf einmal ersetzen. Um jeweils nur eine Instance zu ersetzen, legen Sie einen fehlerfreien minimalen und maximalen Prozentsatz von 100 Prozent fest. Dadurch wird das Verhalten einer Instance-Aktualisierung dahingehend geändert, dass sie vor der Beendigung gestartet wird, wodurch verhindert wird, dass die Kapazität der Gruppe unter 100 Prozent der gewünschten Kapazität fällt. Um alle Instances auf einmal zu ersetzen, legen Sie einen fehlerfreien Mindestprozentsatz von 0 Prozent fest.

Maximaler fehlerfreier Prozentsatz

Der maximale fehlerfreie Prozentsatz ist der Prozentsatz der gewünschten Kapazität, auf den Ihre Auto-Scaling-Gruppe beim Austausch von Instances erhöhen kann. Die Differenz zwischen Minimum und Maximum darf 100 nicht überschreiten. Ein größerer Bereich erhöht die Anzahl der Instances, die gleichzeitig ausgetauscht werden können.

Instance-Aufwärmphase

Der Instance-Warmup ist die Zeitspanne zwischen dem Zeitpunkt, an dem sich der Zustand einer neuen Instance zu `InService` ändert, und dem Zeitpunkt, an dem davon ausgegangen wird, dass sie ihre Initialisierung abgeschlossen hat. Wenn die Instances während einer Instance-Aktualisierung ihre Zustandsprüfungen bestehen, fährt Amazon EC2 Auto Scaling nicht sofort mit dem Ersetzen der nächsten Instance fort, nachdem festgestellt wurde, dass eine neu gestartete Instance fehlerfrei ist. Es wartet die Aufwärmphase ab, bevor es mit dem Ersetzen der nächsten Instance fortfährt. Dies kann hilfreich sein, wenn Ihre Anwendung noch eine gewisse Initialisierungszeit benötigt, bevor sie auf Anfragen reagiert.

Die Aufwärmphase der Instance funktioniert genauso wie die standardmäßige Aufwärmphase der Instance. Daher gelten dieselben Überlegungen zur Skalierung. Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

Gewünschte Konfiguration

Die gewünschte Konfiguration ist die neue Konfiguration, die Amazon EC2 Auto Scaling in Ihrer Auto-Scaling-Gruppe bereitstellen soll. Sie können beispielsweise eine neue Startvorlage und neue Instance-Typen für Ihre Instances angeben. Während einer Instance-Aktualisierung aktualisiert Amazon EC2 Auto Scaling die Auto-Scaling-Gruppe auf die gewünschte Konfiguration. Wenn während einer Instance-Aktualisierung ein Scale-Out-Ereignis auftritt, startet Amazon EC2 Auto Scaling neue Instances mit der gewünschten Konfiguration anstelle der aktuellen Einstellungen der Gruppe. Nach erfolgreicher Instance-Aktualisierung aktualisiert Amazon EC2 Auto Scaling die Auto-Scaling-Gruppeneinstellungen, um die neue gewünschte Konfiguration wiederzugeben, die Sie als Teil der Instance-Aktualisierung angegeben haben.

Überspringen

Das Überspringen des Abgleichs weist Amazon EC2 Auto Scaling an, Instances zu ignorieren, die bereits über Ihre neuesten Aktualisierungen verfügen. Auf diese Weise ersetzen Sie nicht mehr Instances als Sie benötigen. Dies ist hilfreich, wenn Sie sicherstellen möchten, dass Ihre Auto-Scaling-Gruppe eine bestimmte Version Ihrer Startvorlage verwendet und nur die Instances ersetzt, die eine andere Version verwenden.

Prüfpunkte

Ein Checkpoint ist ein Zeitpunkt, an dem die Instance-Aktualisierung für eine bestimmte Zeit angehalten wird. Eine Instance-Aktualisierung kann mehrere Checkpoints enthalten. Amazon EC2 Auto Scaling gibt Ereignisse für jeden Checkpoint aus. Daher können Sie eine EventBridge Regel hinzufügen, um die Ereignisse an ein Ziel wie Amazon SNS zu senden, um benachrichtigt zu werden, wenn ein Checkpoint erreicht wird. Nachdem ein Prüfpunkt erreicht wurde, haben Sie die Möglichkeit, Ihre Bereitstellung zu überprüfen. Wenn Probleme festgestellt werden, können Sie die Instance -Aktualisierung abbrechen oder zurücksetzen. Die Möglichkeit, Updates in Phasen bereitzustellen, ist ein wesentlicher Vorteil von Checkpoints. Wenn Sie keine Checkpoints verwenden, werden fortlaufend rollende Ersetzungen durchgeführt.

Weitere Informationen zu allen Standardeinstellungen, die Sie beim Starten einer Instance-Aktualisierung konfigurieren können, finden Sie unterDie Standardwerte für eine Instance-Aktualisierung verstehen.

Frist der Zustandsprüfung

Amazon EC2 Auto Scaling bestimmt anhand des Status der von Ihrer Auto-Scaling-Gruppe verwendeten Zustandsprüfungen, ob eine Instance fehlerfrei ist. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Um sicherzustellen, dass diese Zustandsprüfungen so schnell wie möglich beginnen, sollten Sie die Karenzzeit für die Zustandsprüfung der Gruppe nicht zu hoch ansetzen, nur hoch genug, damit Ihre Elastic Load Balancing-Zustandsprüfungen feststellen können, ob ein Ziel zur Bearbeitung von Anfragen verfügbar ist. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.

Kompatibilität von Instance-Typen

Bevor Sie Ihren Instance-Typ ändern, sollten Sie überprüfen, ob er mit Ihrer Startvorlage kompatibel ist. Dadurch wird die Kompatibilität mit dem von Ihnen angegebenen AMI bestätigt. Angenommen, Sie haben Ihre ursprünglichen Instances von einem paravirtuellen (PV) AMI gestartet, möchten aber zu einem Instance-Typ der aktuellen Generation wechseln, der nur von einem Hardware Virtual Machine (HVM) AMI unterstützt wird. In diesem Fall müssen Sie in Ihrer Startvorlage ein HVM-AMI verwenden.

Um die Kompatibilität des Instance-Typs zu bestätigen, ohne Instances zu starten, verwenden Sie den Befehl run-instances mit der Option `--dry-run`, wie im folgenden Beispiel gezeigt.

```
aws ec2 run-instances --launch-template LaunchTemplateName=my-template,Version='1' --
dry-run
```

Informationen darüber, wie die Kompatibilität bestimmt wird, finden Sie unter Kompatibilität bei der Änderung des Instance-Typs im Amazon EC2 EC2-Benutzerhandbuch.

Einschränkungen

- Gesamtdauer: Die maximale Zeitspanne, die eine Instance-Aktualisierung aktiv Instances ersetzen kann, beträgt 14 Tage.
- Unterschied im Verhalten gewichteter Gruppen: Wenn eine gemischte Instances-Gruppe mit einer Instance-Gewichtung konfiguriert ist, die größer oder gleich der gewünschten Kapazität der Gruppe ist, ersetzt Amazon EC2 Auto Scaling möglicherweise alle `InService`-Instances auf einmal. Um diese Situation zu vermeiden, folgen Sie der Empfehlung im Konfigurieren Sie eine Auto Scaling Scaling-Gruppe für die Verwendung von Instanzgewichten-Thema. Geben Sie eine gewünschte

Kapazität an, die größer ist als Ihre größte Gewichtung, wenn Sie Gewichtungen mit Ihrer Auto-Scaling-Gruppe verwenden.

- Zeitüberschreitung nach einer Stunde: Wenn eine Instance-Aktualisierung keine weiteren Ersetzungen vornehmen kann, weil sie darauf wartet, Instances im Standby-Modus oder vor Abskalierung geschützte Instances zu ersetzen, oder weil die neuen Instances ihre Zustandsprüfungen nicht bestehen, wiederholt Amazon EC2 Auto Scaling den Versuch eine Stunde lang. Es wird auch eine Statusmeldung angezeigt, mit der Sie das Problem beheben können. Wenn das Problem nach einer Stunde weiterhin besteht, schlägt der Vorgang fehl. Die Absicht besteht darin, ihm im Falle eines vorübergehenden Problems Zeit zur Wiederherstellung zu geben.
- Code mithilfe von Benutzerdaten bereitstellen: Bei Skip Matching wird nicht nach Codeänderungen gesucht, die über ein Benutzerdatenskript bereitgestellt werden. Wenn Sie Benutzerdaten verwenden, um neuen Code abzurufen und diese Updates auf neuen Instances zu installieren, empfehlen wir Ihnen, den Skip-Abgleich zu deaktivieren, um sicherzustellen, dass alle Instanzen Ihren neuesten Code erhalten, auch ohne ein Versionsupdate für die Startvorlage.
- Aktualisierungseinschränkung: Wenn Sie versuchen, die Startvorlage, die Startkonfiguration oder die Richtlinie für gemischte Instanzen einer Auto Scaling Scaling-Gruppe zu aktualisieren, während eine Instance-Aktualisierung mit der gewünschten Konfiguration aktiv ist, schlägt die Anfrage mit dem folgenden Validierungsfehler fehl: `An active instance refresh with a desired configuration exists. All configuration options derived from the desired configuration are not available for update while the instance refresh is active.`

Die Standardwerte für eine Instance-Aktualisierung verstehen

Bevor Sie eine Instance-Aktualisierung starten, können Sie verschiedene Voreinstellungen anpassen, die sich auf die Instance-Aktualisierung auswirken. Einige Voreinstellungen sind unterschiedlich, je nachdem, ob Sie die Konsole oder die Befehlszeile (AWS CLI oder das AWS SDK) verwenden.

In der folgenden Tabelle sind die Standardwerte für Einstellungen der Instance-Aktualisierung aufgeführt.

Einstellung	AWS CLI oder SDK AWS	Die Konsole von Amazon EC2 Auto Scaling
CloudWatch Alarm	Deaktiviert (null)	Disabled

Einstellung	AWS CLI oder SDK AWS	Die Konsole von Amazon EC2 Auto Scaling
Automatisches Zurücksetzen	Deaktiviert (`false`)	Disabled
Prüfpunkte	Deaktiviert (`false`)	Disabled
Checkpoint-Verzögerung	1 Stunde (`3600` Sekunden)	1 Stunde
Instance-Aufwärmphase	Die standardmäßige Instance-Aufwärmphase, falls definiert , oder andernfalls die Frist für die Zustandsprüfung.	Die standardmäßige Instance-Aufwärmphase, falls definiert , oder andernfalls die Frist für die Zustandsprüfung.
Maximaler fehlerfreier Prozentsatz	Variiert je nach Ihrer Instance-Wartungsrichtlinie. Wenn es keine Instanz-Wartungsrichtlinie gibt, ist diese standardmäßig auf 100 Prozent (Null) eingestellt.	Variiert je nach Ihrer Instance-Wartungsrichtlinie. Wenn es keine Instanz-Wartungsrichtlinie gibt, ist diese standardmäßig auf 100 Prozent (Null) eingestellt.
Minimaler fehlerfreier Prozentsatz	Variiert je nach Ihrer Instance-Wartungsrichtlinie. Wenn keine Instance-Wartungsrichtlinie vorhanden ist, wird dieser Wert standardmäßig auf 90 Prozent gesetzt.	Variiert je nach Ihrer Instance-Wartungsrichtlinie. Wenn keine Instance-Wartungsrichtlinie vorhanden ist, wird dieser Wert standardmäßig auf 90 Prozent gesetzt.
Vor Abskalierung geschützte Instances	Wait	Ignore
Überspringen	Deaktiviert (`false`)	Aktiviert
Standby-Instances	Wait	Ignore

Es folgt eine Beschreibung der einzelnen Einstellungen:

CloudWatch Alarmanlage (**`AlarmSpecification`**)

Die CloudWatch Alarmspezifikation. CloudWatch Alarme können verwendet werden, um Probleme zu identifizieren und den Vorgang fehlschlagen zu lassen, wenn ein Alarm in den `ALARM` Status wechselt. Weitere Informationen finden Sie unter Starten einer Instance-Aktualisierung mit automatischem Rollback.

Automatisches Zurücksetzen (**`AutoRollback`**)

Legt fest, ob Amazon EC2 Auto Scaling die Auto-Scaling-Gruppe auf ihre vorherige Konfiguration zurücksetzt, wenn die Aktualisierung der Instance fehlschlägt. Weitere Informationen finden Sie unter Änderungen mit einem Rollback rückgängig machen.

Checkpoints (**`CheckpointPercentages`**)

Steuert, ob Amazon EC2 Auto Scaling Instances phasenweise ersetzt. Dies ist nützlich, wenn Sie Ihre Instances überprüfen müssen, bevor Sie alle Instances austauschen. Weitere Informationen finden Sie unter Prüfpunkte zu einer Instance-Aktualisierung hinzufügen.

Checkpoint-Verzögerung (**`CheckpointDelay`**)

Die Zeit in Sekunden, die nach einem Checkpoint gewartet werden muss, bevor fortgefahren wird. Weitere Informationen finden Sie unter Prüfpunkte zu einer Instance-Aktualisierung hinzufügen.

Instance-Aufwärmphase (**`InstanceWarmup`**)

Eine Zeitspanne in Sekunden, in der Amazon EC2 Auto Scaling wartet, bis die Initialisierung einer neuen Instance als abgeschlossen gilt, bevor die nächste Instance ersetzt wird. Wenn Sie bereits eine standardmäßige Aufwärmphase für die Instance der Auto-Scaling-Gruppe definiert haben, müssen Sie die Aufwärmphase für die Instance nicht ändern (es sei denn, Sie möchten den Standard überschreiben). Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

Maximaler fehlerfreier Prozentsatz (**`MaxHealthyPercentage`**)

Der Prozentsatz der gewünschten Kapazität der Auto-Scaling-Gruppe, auf den sich Ihre Gruppe beim Ersetzen von Instances erhöhen kann.

Minimaler fehlerfreier Prozentsatz (**`MinHealthyPercentage`**)

Der Prozentsatz der gewünschten Kapazität der Auto-Scaling-Gruppe, der betriebsbereit, fehlerfrei und einsatzbereit sein muss, bevor der Vorgang fortgesetzt werden kann.

Vor Abskalierung geschützte Instances (**`ScaleInProtectedInstances`**)

Steuert, was Amazon EC2 Auto Scaling macht, wenn Instances gefunden werden, die vor dem Abskalieren geschützt sind. Weitere Informationen zu diesen Instances finden Sie unter Instance-Abskalierungsschutz verwenden.

Amazon EC2 Auto Scaling bietet die folgenden Optionen an:

- Ersetzen (`Refresh`) — Ersetzt Instanzen, die vor dem Skalieren geschützt sind.
- Ignorieren (`Ignore`) — Ignoriert Instances, die vor einer Skalierung geschützt sind, und ersetzt weiterhin Instances, die nicht geschützt sind.
- Warten (`Wait`) — Wartet eine Stunde, bis Sie den Scale-In-Schutz entfernt haben. Wenn Sie dies nicht tun, schlägt die Instance-Aktualisierung fehl.

Überspringen des Abgleichs (**`SkipMatching`**)

Steuert, ob Amazon EC2 Auto Scaling das Ersetzen von Instances, die der gewünschten Konfiguration entsprechen, überspringt. Wenn keine gewünschte Konfiguration angegeben wird, werden Instances mit der gleichen Startvorlage und den gleichen Instance-Typen, die die Auto-Scaling-Gruppe vor dem Start der Instace-Aktualisierung verwendet hat, nicht ersetzt. Weitere Informationen finden Sie unter Verwenden einer Instance-Aktualisierung mit Funktion zum Überspringen des Abgleichs.

Standby-Instances (**`StandbyInstances`**)

Steuert, was Amazon EC2 Auto Scaling macht, wenn sich Instances im Zustand `Standby` befinden. Weitere Informationen zu diesen Instances finden Sie unter Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe.

Amazon EC2 Auto Scaling bietet die folgenden Optionen an:

- Terminate (`Terminate`) — Beendet Instances, die aktiv sind. `Standby`
- Ignorieren (`Ignore`) — Ignoriert Instanzen, die sich im Status befinden, `Standby` und ersetzt weiterhin Instanzen, die sich im Status befinden. `InService`
- Warten (`Wait`) — Wartet eine Stunde, bis Sie die Instances wieder in Betrieb nehmen. Wenn Sie dies nicht tun, schlägt die Instance-Aktualisierung fehl.

Starten einer Instance-Aktualisierung

> **Important**
>
> Sie können eine Instance-Aktualisierung, die gerade ausgeführt wird, zurücksetzen, um alle Änderungen rückgängig zu machen. Damit dies funktioniert, muss die Auto-Scaling-Gruppe die Voraussetzungen für die Verwendung von Rollbacks erfüllen, bevor die Instance-Aktualisierung gestartet wird. Weitere Informationen finden Sie unter Änderungen mit einem Rollback rückgängig machen.

Die folgenden Verfahren helfen Ihnen, eine Instanzaktualisierung mit dem AWS Management Console oder AWS CLI zu starten.

Starten einer Instance-Aktualisierung (Konsole)

Durch das erste Starten einer Instance-Aktualisierung mit der Konsole werden Sie die verfügbaren Funktionen und Optionen besser verstehen.

Starten einer Instance-Aktualisierung in der Konsole (grundlegendes Verfahren)

Gehen Sie wie folgt vor, wenn Sie zuvor keine mixed instances policy (Richtlinie für gemischte Instances) für Ihre Auto-Scaling-Gruppe definiert haben. Wenn Sie zuvor eine Richtlinie für gemischte Instances definiert haben, lesen Sie Starten einer Instance-Aktualisierung in der Konsole (Gruppe mit gemischten Instances) zum Starten einer Instance-Aktualisierung.

So starten Sie eine Instance-Aktualisierung

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Unten auf der Seite Auto-Scaling-Gruppen wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Instance refresh (Instance-Aktualisierung) unter Active instance refresh (Aktive Instance-Aktualisierung) die Option Start instance refresh (Instance-Aktualisierung starten) aus.
4. Für die Verfügbarkeitseinstellungen gehen Sie wie folgt vor:

   a. Für die Methode der Instance-Ersetzung:

- Wenn Sie keine Instance-Wartungsrichtlinie für die Auto-Scaling-Gruppe festgelegt haben, lautet die Standardeinstellung für die Methode zum Ersetzen von Instances Beenden und starten. Dies ist das alte Standardverhalten für eine Instance-Aktualisierung.
- Wenn Sie eine Instance-Wartungsrichtlinie für die Auto-Scaling-Gruppe festlegen, bietet diese Standardwerte für die Instance-Ersatzmethode. Um die Instance-Wartungsrichtlinie zu überschreiben, wählen Sie Override. Überschreiben gilt nur für die aktuelle Instance-Aktualisierung. Wenn Sie das nächste Mal eine Instance-Aktualisierung starten, werden diese Werte auf die Standardwerte der Instance-Wartungsrichtlinie zurückgesetzt.

Im folgenden Verfahren wird erläutert, wie die Instance-Ersatzmethode aktualisiert werden kann.

i. Wählen Sie eine der folgenden Instance-Ersatzmethoden:

   - Vor dem Beenden starten: Eine neue Instance muss zuerst bereitgestellt werden, bevor eine bestehende Instance beendet werden kann. Dies ist eine gute Wahl für Anwendungen, bei denen Verfügbarkeit wichtiger ist als Kosteneinsparungen.
   - Beenden und starten: Neue Instances werden zur gleichen Zeit bereitgestellt, wie Ihre bestehenden Instances beendet werden. Dies ist eine gute Wahl für Anwendungen, bei denen Kosteneinsparungen Vorrang vor der Verfügbarkeit haben. Es ist auch eine gute Wahl für Anwendungen, die nicht mehr Kapazität benötigen, als derzeit verfügbar ist.
   - Benutzerdefiniertes Verhalten: Mit dieser Option können Sie einen benutzerdefinierten Mindest- und Höchstbereich für die Kapazität einrichten, die beim Austausch von Instances verfügbar sein soll. Dies kann Ihnen helfen, das richtige Gleichgewicht zwischen Kosten und Verfügbarkeit zu finden.

ii. Geben Sie unter Fehlerfreien Prozentsatz festlegen Werte für eines oder beide der folgenden Felder ein. Die Aktivierungsfelder variieren je nach gewählter Option für die Instance-Ersatzmethode.

   - Min.: Legt den fehlerfreien Mindestprozentsatz fest, der erforderlich ist, um mit der Instance-Aktualisierung fortzufahren.
   - Max.: Legt den maximalen fehlerfreien Prozentsatz fest, der während der Instance-Aktualisierung möglich ist.

iii. Erweitern Sie den Abschnitt Geschätzte temporäre Kapazität bei Austauscharbeiten auf der Grundlage der aktuellen Gruppengröße anzeigen, um zu überprüfen, ob die Werte für Min. und Max für Ihre Gruppe gelten. Welche genauen Werte verwendet werden, hängt vom gewünschten Kapazitätswert ab, der sich ändert, wenn die Gruppe skaliert wird.

iv. Erweitern Sie den Abschnitt Fallback-Verhalten für ungültige Ersatzgrößen festlegen und wählen Sie dann aus, ob Sie gegen den maximalen fehlerfreien Prozentsatz verstoßen möchten, um der Verfügbarkeit Priorität einzuräumen, oder ob Sie gegen den minimalen fehlerfreien Prozentsatz verstoßen möchten.

Es wird für sehr kleine Gruppen nicht empfohlen, die Standardoption Minimaler fehlerfreie Prozentsatz beizubehalten. Wenn sich nur eine Instance in der Auto-Scaling-Gruppe befindet, kann das Starten einer Instance-Aktualisierung zu einem Ausfall führen.

Dieser Schritt konfiguriert das Fallback-Verhalten, wenn Sie eine Auto-Scaling-Gruppe verwenden, die noch keine Instance-Wartungsrichtlinie hat. Diese Option ist nicht verfügbar und wird nicht angezeigt, wenn Ihre Gruppe über eine Instance-Wartungsrichtlinie verfügt. Diese Option ist auch nur für die Ersatzmethode Beenden und Starten verfügbar. Bei anderen Ersatzmethoden wird die maximale Fehlerquote überschritten, um der Verfügbarkeit Priorität einzuräumen.

b. Geben Sie für Instance-Aufwärmphase die Anzahl der Sekunden ein, ab der sich der Status einer neuen Instance in `InService` ändert, wenn sie ihre Initialisierung abgeschlossen hat. Amazon EC2 Auto Scaling wartet diese Zeit ab, bevor es mit dem Ersetzen der nächsten Instance fortfährt.

Während der Aufwärmphase wird eine neu gestartete Instance nicht zu den aggregierten Metriken der Auto-Scaling-Gruppe (z. B. `CPUUtilization`, `NetworkIn` und `NetworkOut`) gezählt. Wenn Sie der Auto-Scaling-Gruppe Skalierungsrichtlinien hinzugefügt haben, werden die Skalierungsaktivitäten parallel ausgeführt. Wenn Sie ein langes Intervall für die Aufwärmphase der Instance-Aktualisierung festlegen, dauert es länger, bis neu gestartete Instances in den Metriken angezeigt werden. Daher verhindert eine angemessene Aufwärmphase, dass Amazon EC2 Auto Scaling auf veraltete Metrikdaten skaliert.

Wenn Sie bereits eine standardmäßige Aufwärmphase für die Instance der Auto-Scaling-Gruppe definiert haben, müssen Sie die Aufwärmphase für die Instance nicht ändern. Wenn Sie die Standardeinstellung jedoch überschreiben möchten, können Sie einen Wert

für diese Option festlegen. Weitere Informationen zum Festlegen der standardmäßigen Aufwärmphase der Instance finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

5. Für Einstellungen aktualisieren nehmen Sie Folgendes vor:

   a. (Optional) Für Checkpoints (Prüfpunkte) wählen Sie Enable Checkpoints (Prüfpunkte aktivieren) aus, um Instances mit einem inkrementellen oder schrittweisen Ansatz für eine Instance-Aktualisierung zu ersetzen. Dies bietet zusätzliche Zeit für die Verifizierung verschiedener Ersatzvorgänge. Wenn Sie sich dafür entscheiden, Prüfpunkte nicht zu aktivieren, werden die Instances in einem fast kontinuierlichen Vorgang ersetzt.

   Wenn Sie Prüfpunkte aktivieren, finden Sie unter Aktivieren von Prüfpunkten (Konsole) zusätzliche Schritte.

   b. Aktivieren oder Deaktivieren von Skip Matching (Abgleich überspringen):

   - Um das Ersetzen von Instances zu überspringen, die bereits mit Ihrer Startvorlage übereinstimmen, lassen Sie das Kontrollkästchen Überspringen des Abgleichs aktivieren aktiviert.
   - Wenn Sie das Überspringen des Abgleichs deaktivieren, indem Sie dieses Kontrollkästchen deaktivieren, können alle Instances ersetzt werden.

   Wenn Sie das Überspringen des Abgleichs aktivieren, können Sie eine neue Startvorlage oder eine neue Version der Startvorlage festlegen, anstatt die vorhandene zu verwenden. Sie können dies im Abschnitt Gewünschte Konfiguration auf der Seite Instance-Aktualisierung starten tun.

   > **Note**
   >
   > Um die Funktion „Abgleich überspringen“ zur Aktualisierung einer Auto-Scaling-Gruppe zu verwenden, die derzeit eine Startkonfiguration verwendet, müssen Sie eine Startvorlage in Desired configuration (Gewünschte Konfiguration) auswählen. Die Verwendung von „Abgleich überspringen“ mit einer Startkonfiguration wird nicht unterstützt.

   c. Wählen Sie für Standby-Instances die Option Ignorieren, Beenden oder Warten aus. Dies legt fest, was passiert, wenn Instances im `Standby`-Status erkannt werden. Weitere

Informationen finden Sie unter Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe.

Wenn Sie Warten auswählen, müssen Sie zusätzliche Schritte unternehmen, um diese Instances wieder in Betrieb zu nehmen. Wenn Sie dies nicht tun, ersetzt die Instance-Aktualisierung alle `InService`-Instances und wartet eine Stunde. Wenn dann noch `Standby`-Instances übrig bleiben, schlägt die Instance-Aktualisierung fehl. Um diese Situation zu vermeiden, wählen Sie für diese Instances stattdessen Ignorieren oder Beenden aus.

d. Wählen Sie für Vor Abskalierung geschützte Instances die Option Ignorieren, Ersetzen oder Warten aus. Dies legt fest, was passiert, wenn vor Abskalierung geschützte Instances erkannt werden. Weitere Informationen finden Sie unter Instance-Abskalierungsschutz verwenden.

Wenn Sie Warten auswählen, müssen Sie zusätzliche Schritte unternehmen, um den Abskalierungsschutz dieser Instances zu entfernen. Wenn Sie dies nicht tun, ersetzt die Instance-Aktualisierung alle ungeschützten Instances und wartet eine Stunde. Wenn dann vor Abskalierung geschützte Instances übrig bleiben, schlägt die Instance-Aktualisierung fehl. Um diese Situation zu verhindern, wählen Sie stattdessen für diese Instances Ignorieren oder Ersetzen aus.

6. (Optional) Wählen Sie für CloudWatch Alarm die Option CloudWatch Alarme aktivieren und wählen Sie dann einen oder mehrere Alarme aus. CloudWatch Alarme können verwendet werden, um Probleme zu identifizieren und den Vorgang fehlschlagen zu lassen, wenn ein Alarm in den `ALARM` Status wechselt. Weitere Informationen finden Sie unter Starten einer Instance-Aktualisierung mit automatischem Rollback.
7. (Optional) Erweitern Sie den Abschnitt Gewünschte Konfiguration, um Aktualisierungen anzugeben, die Sie an Ihrer Auto-Scaling-Gruppe vornehmen möchten.

   Für diesen Schritt können Sie die JSON- oder YAML-Syntax verwenden, um Parameterwerte zu bearbeiten, anstatt eine Auswahl in der Konsolenschnittstelle zu treffen. Wählen Sie hierfür Use code editor (Code-Editor verwenden) anstelle von Use console interface (Konsolenschnittstelle verwenden) aus. Im folgenden Verfahren wird erläutert, wie mit der Konsolenschnittstelle eine Auswahl getroffen werden kann.

   a. Für Update launch template (Startvorlage aktualisieren):

      - Wenn Sie keine neue Startvorlage oder neue Startvorlagenversion für Ihre Auto-Scaling-Gruppe erstellt haben, aktivieren Sie dieses Kontrollkästchen nicht.

- Wenn Sie eine neue Startvorlage oder eine neue Startvorlagen-Version erstellt haben, aktivieren Sie dieses Kontrollkästchen. Wenn Sie diese Option auswählen, zeigt Amazon EC2 Auto Scaling Ihnen die aktuelle Startvorlage und die aktuelle Version der Startvorlage an. Es listet auch alle anderen verfügbaren Versionen auf. Wählen Sie die Startvorlage und dann die Version aus.

  Nachdem Sie eine Version ausgewählt haben, werden Ihnen die Versionsinformationen angezeigt. Dies ist die Version der Startvorlage, die beim Ersetzen von Instances im Rahmen einer Instance-Aktualisierung verwendet wird. Wenn die Instance-Aktualisierung erfolgreich ist, wird diese Version der Startvorlage auch verwendet, wenn neue Instances gestartet werden, z. B. wenn die Gruppe skaliert wird.

b. Für Choose a set of instance types and purchase options to override the instance type in the launch template (Eine Reihe von Instance-Typen und Kaufoptionen auswählen, um den Instance-Typ in der Startvorlage außer Kraft zu setzen):

- Aktivieren Sie dieses Kontrollkästchen nicht, wenn Sie den Instance-Typ und die Kaufoption verwenden möchten, die Sie in Ihrer Startvorlage angegeben haben.
- Aktivieren Sie dieses Kontrollkästchen, wenn Sie den Instance-Typ in der Startvorlage überschreiben oder Spot-Instances ausführen möchten. Sie können jeden Instance-Typ entweder manuell hinzufügen oder einen primären Instance-Typ und eine Empfehlungsoption auswählen, die alle zusätzlichen passenden Instance-Typen für Sie abruft. Wenn Sie Spot-Instances starten möchten, empfehlen wir, einige unterschiedliche Instance-Typen hinzuzufügen. Dadurch kann Amazon EC2 Auto Scaling einen weiteren Instance-Typ starten, wenn in den ausgewählten Availability Zones nicht genügend Instance-Kapazität zur Verfügung steht. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.

  > **Warning**
  >
  > Verwenden Sie Spot-Instances nicht mit Anwendungen, die eine Spot-Instance-Unterbrechung nicht verarbeiten können. Unterbrechungen können auftreten, wenn der Amazon-EC2-Spot-Service Kapazität zurückfordern muss.

Wenn Sie dieses Kontrollkästchen auswählen, stellen Sie sicher, dass die Startvorlage nicht bereits Spot-Instances anfordert. Sie können keine Startvorlage verwenden, die Spot-

Instances zum Erstellen einer Auto-Scaling-Gruppe auffordert, die mehrere Instance-Typen verwendet und Spot- und On-Demand-Instances startet.

> **Note**
>
> Um diese Optionen in einer Auto-Scaling-Gruppe zu aktualisieren, die derzeit eine Startkonfiguration verwendet, müssen Sie eine Startvorlage in Update launch Vorlage (Startvorlage aktualisieren) auswählen. Das Überschreiben des Instance-Typs in Ihrer Startkonfiguration wird nicht unterstützt.

8. (Optional) Wählen Sie unter Rollback-Einstellungen die Option Automatisches Rollback aktivieren aus, um die Instance-Aktualisierung automatisch zurückzusetzen, falls sie fehlschlägt.

   Diese Einstellung kann nur aktiviert werden, wenn die Auto-Scaling-Gruppe die Voraussetzungen für die Verwendung von Rollbacks erfüllt.

   Weitere Informationen finden Sie unter Änderungen mit einem Rollback rückgängig machen.
9. Überprüfen Sie Ihre gesamte Auswahl, um zu bestätigen, dass alles korrekt eingerichtet ist.

   An dieser Stelle empfiehlt es sich sicherzustellen, dass sich die Unterschiede zwischen den aktuellen und den vorgeschlagenen Änderungen nicht auf unerwartete oder unerwünschte Weise auf Ihre Anwendung auswirken. Informationen zur Bestätigung, dass Ihr Instance-Typ mit Ihrer Startvorlage kompatibel ist, finden Sie unter Kompatibilität von Instance-Typen.
10. Wenn Sie mit der Auswahl für Ihre Instance-Aktualisierung zufrieden sind, klicken Sie auf Instance-Aktualisierung Starten.

Starten einer Instance-Aktualisierung in der Konsole (Gruppe mit gemischten Instances)

Gehen Sie wie folgt vor, wenn Sie eine Auto-Scaling-Gruppe mit einer Richtlinie für gemischte Instances erstellt haben. Wenn Sie zuvor keine Richtlinie für gemischte Instances für Ihre Gruppe definiert haben, lesen Sie Starten einer Instance-Aktualisierung in der Konsole (grundlegendes Verfahren) zum Starten einer Instance-Aktualisierung.

So starten Sie eine Instance-Aktualisierung

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

Unten auf der Seite Auto-Scaling-Gruppen wird ein geteilter Bereich geöffnet.

3. Wählen Sie auf der Registerkarte Instance refresh (Instance-Aktualisierung) unter Active instance refresh (Aktive Instance-Aktualisierung) die Option Start instance refresh (Instance-Aktualisierung starten) aus.
4. Für die Verfügbarkeitseinstellungen gehen Sie wie folgt vor:

   a. Für die Methode der Instance-Ersetzung:

      - Wenn Sie keine Instance-Wartungsrichtlinie für die Auto-Scaling-Gruppe festgelegt haben, lautet die Standardeinstellung für die Methode zum Ersetzen von Instances Beenden und starten. Dies ist das alte Standardverhalten für eine Instance-Aktualisierung.
      - Wenn Sie eine Instance-Wartungsrichtlinie für die Auto-Scaling-Gruppe festlegen, bietet diese Standardwerte für die Instance-Ersatzmethode. Um die Instance-Wartungsrichtlinie zu überschreiben, wählen Sie Override. Überschreiben gilt nur für die aktuelle Instance-Aktualisierung. Wenn Sie das nächste Mal eine Instance-Aktualisierung starten, werden diese Werte auf die Standardwerte der Instance-Wartungsrichtlinie zurückgesetzt.

      Im folgenden Verfahren wird erläutert, wie die Instance-Ersatzmethode aktualisiert werden kann.

      i. Wählen Sie eine der folgenden Instance-Ersatzmethoden:

         - Vor dem Beenden starten: Eine neue Instance muss zuerst bereitgestellt werden, bevor eine bestehende Instance beendet werden kann. Dies ist eine gute Wahl für Anwendungen, bei denen Verfügbarkeit wichtiger ist als Kosteneinsparungen.
         - Beenden und starten: Neue Instances werden zur gleichen Zeit bereitgestellt, wie Ihre bestehenden Instances beendet werden. Dies ist eine gute Wahl für Anwendungen, bei denen Kosteneinsparungen Vorrang vor der Verfügbarkeit haben. Es ist auch eine gute Wahl für Anwendungen, die nicht mehr Kapazität benötigen, als derzeit verfügbar ist.
         - Benutzerdefiniertes Verhalten: Mit dieser Option können Sie einen benutzerdefinierten Mindest- und Höchstbereich für die Kapazität einrichten, die beim Austausch von Instances verfügbar sein soll. Dies kann Ihnen helfen, das richtige Gleichgewicht zwischen Kosten und Verfügbarkeit zu finden.

ii. Geben Sie unter Fehlerfreien Prozentsatz festlegen Werte für eines oder beide der folgenden Felder ein. Die Aktivierungsfelder variieren je nach gewählter Option für die Instance-Ersatzmethode.

- Min.: Legt den fehlerfreien Mindestprozentsatz fest, der erforderlich ist, um mit der Instance-Aktualisierung fortzufahren.
- Max.: Legt den maximalen fehlerfreien Prozentsatz fest, der während der Instance-Aktualisierung möglich ist.

iii. Erweitern Sie den Abschnitt Geschätzte temporäre Kapazität bei Austauscharbeiten auf der Grundlage der aktuellen Gruppengröße anzeigen, um zu überprüfen, ob die Werte für Min. und Max für Ihre Gruppe gelten. Welche genauen Werte verwendet werden, hängt vom gewünschten Kapazitätswert ab, der sich ändert, wenn die Gruppe skaliert wird.

iv. Erweitern Sie den Abschnitt Fallback-Verhalten für ungültige Ersatzgrößen festlegen und wählen Sie dann aus, ob Sie gegen den maximalen fehlerfreien Prozentsatz verstoßen möchten, um der Verfügbarkeit Priorität einzuräumen, oder ob Sie gegen den minimalen fehlerfreien Prozentsatz verstoßen möchten.

Es wird für sehr kleine Gruppen nicht empfohlen, die Standardoption Minimaler fehlerfreie Prozentsatz beizubehalten. Wenn sich nur eine Instance in der Auto-Scaling-Gruppe befindet, kann das Starten einer Instance-Aktualisierung zu einem Ausfall führen.

Dieser Schritt konfiguriert das Fallback-Verhalten, wenn Sie eine Auto-Scaling-Gruppe verwenden, die noch keine Instance-Wartungsrichtlinie hat. Diese Option ist nicht verfügbar und wird nicht angezeigt, wenn Ihre Gruppe über eine Instance-Wartungsrichtlinie verfügt. Diese Option ist auch nur für die Ersatzmethode Beenden und Starten verfügbar. Bei anderen Ersatzmethoden wird die maximale Fehlerquote überschritten, um der Verfügbarkeit Priorität einzuräumen.

b. Geben Sie für Instance-Aufwärmphase die Anzahl der Sekunden ein, ab der sich der Status einer neuen Instance in `InService` ändert, wenn sie ihre Initialisierung abgeschlossen hat. Amazon EC2 Auto Scaling wartet diese Zeit ab, bevor es mit dem Ersetzen der nächsten Instance fortfährt.

Während der Aufwärmphase wird eine neu gestartete Instance nicht zu den aggregierten Metriken der Auto-Scaling-Gruppe (z. B. `CPUUtilization`, `NetworkIn` und `NetworkOut`) gezählt. Wenn Sie der Auto-Scaling-Gruppe Skalierungsrichtlinien hinzugefügt haben,

werden die Skalierungsaktivitäten parallel ausgeführt. Wenn Sie ein langes Intervall für die Aufwärmphase der Instanzaktualisierung festlegen, dauert es länger, bis neu gestartete Instances in den Metriken angezeigt werden. Daher verhindert eine angemessene Aufwärmphase, dass Amazon EC2 Auto Scaling auf veraltete Metrikdaten skaliert.

Wenn Sie bereits eine standardmäßige Aufwärmphase für die Instance der Auto-Scaling-Gruppe definiert haben, müssen Sie die Aufwärmphase für die Instance nicht ändern. Wenn Sie die Standardeinstellung jedoch überschreiben möchten, können Sie einen Wert für diese Option festlegen. Weitere Informationen zum Festlegen der standardmäßigen Aufwärmphase der Instance finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

5. Für Einstellungen aktualisieren nehmen Sie Folgendes vor:

   a. (Optional) Für Checkpoints (Prüfpunkte) wählen Sie Enable Checkpoints (Prüfpunkte aktivieren) aus, um Instances mit einem inkrementellen oder schrittweisen Ansatz für eine Instance-Aktualisierung zu ersetzen. Dies bietet zusätzliche Zeit für die Verifizierung verschiedener Ersatzvorgänge. Wenn Sie sich dafür entscheiden, Prüfpunkte nicht zu aktivieren, werden die Instances in einem fast kontinuierlichen Vorgang ersetzt.

      Wenn Sie Prüfpunkte aktivieren, finden Sie unter Aktivieren von Prüfpunkten (Konsole) zusätzliche Schritte.

   b. Aktivieren oder Deaktivieren von Skip Matching (Abgleich überspringen):

      - Um das Ersetzen von Instances zu überspringen, die bereits mit Ihrer Startvorlage und allen Instance-Typ-Überschreibungen übereinstimmen, lassen Sie das Kontrollkästchen Überspringen des Abgleichs aktivieren aktiviert.
      - Wenn Sie das Überspringen des Abgleichs deaktivieren, indem Sie dieses Kontrollkästchen deaktivieren, können alle Instances ersetzt werden.

      Wenn Sie das Überspringen des Abgleichs aktivieren, können Sie eine neue Startvorlage oder eine neue Version der Startvorlage festlegen, anstatt die vorhandene zu verwenden. Sie können dies im Abschnitt Gewünschte Konfiguration auf der Seite Instance-Aktualisierung starten tun. Sie können Ihre Instance-Typ-Überschreibungen auch in Gewünschte Konfiguration aktualisieren.

   c. Wählen Sie für Standby-Instances die Option Ignorieren, Beenden oder Warten aus. Dies legt fest, was passiert, wenn Instances im `Standby`-Status erkannt werden. Weitere

Informationen finden Sie unter Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe.

Wenn Sie Warten auswählen, müssen Sie zusätzliche Schritte unternehmen, um diese Instances wieder in Betrieb zu nehmen. Wenn Sie dies nicht tun, ersetzt die Instance-Aktualisierung alle `InService`-Instances und wartet eine Stunde. Wenn dann `Standby`-Instances übrig bleiben, schlägt die Instance-Aktualisierung fehl. Um diese Situation zu vermeiden, wählen Sie für diese Instances stattdessen Ignorieren oder Beenden aus.

d. Wählen Sie für Vor Abskalierung geschützte Instances die Option Ignorieren, Ersetzen oder Warten aus. Dies legt fest, was passiert, wenn vor Abskalierung geschützte Instances erkannt werden. Weitere Informationen finden Sie unter Instance-Abskalierungsschutz verwenden.

Wenn Sie Warten auswählen, müssen Sie zusätzliche Schritte unternehmen, um den Abskalierungsschutz dieser Instances zu entfernen. Wenn Sie dies nicht tun, ersetzt die Instance-Aktualisierung alle ungeschützten Instances und wartet eine Stunde. Wenn dann vor Abskalierung geschützte Instances übrig bleiben, schlägt die Instance-Aktualisierung fehl. Um diese Situation zu verhindern, wählen Sie stattdessen für diese Instances Ignorieren oder Ersetzen aus.

6. (Optional) Wählen Sie für CloudWatch Alarm die Option CloudWatch Alarme aktivieren und wählen Sie dann einen oder mehrere Alarme aus. CloudWatch Alarme können verwendet werden, um Probleme zu identifizieren und den Vorgang fehlschlagen zu lassen, wenn ein Alarm in den `ALARM` Status wechselt. Weitere Informationen finden Sie unter Starten einer Instance-Aktualisierung mit automatischem Rollback.
7. Führen Sie im Abschnitt Gewünschte Konfiguration Folgendes aus.

Für diesen Schritt können Sie die JSON- oder YAML-Syntax verwenden, um Parameterwerte zu bearbeiten, anstatt eine Auswahl in der Konsolenschnittstelle zu treffen. Wählen Sie hierfür Use code editor (Code-Editor verwenden) anstelle von Use console interface (Konsolenschnittstelle verwenden) aus. Im folgenden Verfahren wird erläutert, wie mit der Konsolenschnittstelle eine Auswahl getroffen werden kann.

a. Für Update launch template (Startvorlage aktualisieren):

- Wenn Sie keine neue Startvorlage oder neue Startvorlagenversion für Ihre Auto-Scaling-Gruppe erstellt haben, aktivieren Sie dieses Kontrollkästchen nicht.

- Wenn Sie eine neue Startvorlage oder eine neue Startvorlagen-Version erstellt haben, aktivieren Sie dieses Kontrollkästchen. Wenn Sie diese Option auswählen, zeigt Amazon EC2 Auto Scaling Ihnen die aktuelle Startvorlage und die aktuelle Version der Startvorlage an. Es listet auch alle anderen verfügbaren Versionen auf. Wählen Sie die Startvorlage und dann die Version aus.

  Nachdem Sie eine Version ausgewählt haben, werden Ihnen die Versionsinformationen angezeigt. Dies ist die Version der Startvorlage, die beim Ersetzen von Instances im Rahmen einer Instance-Aktualisierung verwendet wird. Wenn die Instance-Aktualisierung erfolgreich ist, wird diese Version der Startvorlage auch verwendet, wenn neue Instances gestartet werden, z. B. wenn die Gruppe skaliert wird.

b. Für Use these settings to override the instance type and purchase option defined in the launch template (Verwenden Sie diese Einstellungen, um den Instance-Typ und die Kaufoption außer Kraft zu setzen, die in der Startvorlage definiert sind):

   In der Standardeinstellung ist dieses Kontrollkästchen aktiviert. Amazon EC2 Auto Scaling füllt die einzelnen Parameter mit dem Wert, der derzeit in der Richtlinie für gemischte Instances für die Auto-Scaling-Gruppe festgelegt ist. Aktualisieren Sie nur die Werte für die Parameter, die Sie ändern möchten. Hinweise zu diesen Einstellungen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.

   > **Warning**
   >
   > Es wird empfohlen, dieses Kontrollkästchen nicht zu deaktivieren. Deaktivieren Sie es nur, wenn Sie die Verwendung einer Richtlinie für gemischte Instances beenden möchten. Nach einer erfolgreichen Instance-Aktualisierung aktualisiert Amazon EC2 Auto Scaling Ihre Gruppe, damit Sie mit der gewünschten Konfiguration übereinstimmt. Wenn es keine Richtlinie für gemischte Instances mehr enthält, beendet Amazon EC2 Auto Scaling schrittweise alle derzeit ausgeführten Spot-Instances und ersetzt sie durch On-Demand-Instances. Oder, wenn Ihre Startvorlage Spot-Instances anfordert, beendet Amazon EC2 Auto Scaling schrittweise alle derzeit ausgeführten On-Demand-Instances und ersetzt sie durch Spot-Instances.

8. (Optional) Wählen Sie unter Rollback-Einstellungen die Option Automatisches Rollback aktivieren aus, um die Instance-Aktualisierung automatisch zurückzusetzen, falls sie fehlschlägt.

Diese Einstellung kann nur aktiviert werden, wenn die Auto-Scaling-Gruppe die Voraussetzungen für die Verwendung von Rollbacks erfüllt.

Weitere Informationen finden Sie unter Änderungen mit einem Rollback rückgängig machen.

9. Überprüfen Sie Ihre gesamte Auswahl, um zu bestätigen, dass alles korrekt eingerichtet ist.

   An dieser Stelle empfiehlt es sich sicherzustellen, dass sich die Unterschiede zwischen den aktuellen und den vorgeschlagenen Änderungen nicht auf unerwartete oder unerwünschte Weise auf Ihre Anwendung auswirken. Informationen zur Bestätigung, dass Ihr Instance-Typ mit Ihrer Startvorlage kompatibel ist, finden Sie unter Kompatibilität von Instance-Typen.

   Wenn Sie mit der Auswahl für Ihre Instance-Aktualisierung zufrieden sind, klicken Sie auf Instance-Aktualisierung Starten.

Starten einer Instance-Aktualisierung (AWS CLI)

So starten Sie eine Instance-Aktualisierung

Verwenden Sie den Befehl start-instance-refresh, um eine Instance-Aktualisierung über die AWS CLI zu starten. Sie können alle Voreinstellungen angeben, die Sie in einer JSON-Konfigurationsdatei ändern möchten. Wenn Sie auf die Konfigurationsdatei verweisen, geben Sie den Dateipfad und -namen an, wie im folgenden Beispiel gezeigt.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Inhalt von `config.json`:

```
{
    "AutoScalingGroupName": "my-asg",
    "Preferences": {
      "InstanceWarmup": 60,
      "MinHealthyPercentage": 50,
      "AutoRollback": true,
      "ScaleInProtectedInstances": Ignore,
      "StandbyInstances": Terminate
    }
}
```

Wenn keine Voreinstellungen angegeben werden, werden die Standardwerte verwendet. Weitere Informationen finden Sie unter Die Standardwerte für eine Instance-Aktualisierung verstehen.

Beispielausgabe:

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Überwachen Sie die Aktualisierung einer Instanz

Sie können eine laufende Instanzaktualisierung überwachen oder den Status vergangener Instanzaktualisierungen der letzten sechs Wochen mit dem oder nachschlagen. AWS Management Console AWS CLI

Überwachen und überprüfen Sie den Status einer Instanzaktualisierung

Verwenden Sie eine der folgenden Methoden, um den Status einer Instanzaktualisierung zu überwachen und zu überprüfen:

Console

> **Tip**
>
> In diesem Verfahren sollten die benannten Spalten bereits angezeigt werden. Um ausgeblendete Spalten anzuzeigen oder die Anzahl der angezeigten Zeilen zu ändern, wählen Sie das Zahnradsymbol in der oberen rechten Ecke des Abschnitts, um das Einstellungsfenster zu öffnen. Aktualisieren Sie die Einstellungen nach Bedarf und wählen Sie Bestätigen aus.

Um den Status einer Instanzaktualisierung zu überwachen und zu überprüfen (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Auf der Registerkarte Instance refresh (Instance-Aktualisierung) unter Instance refresh history (Instance-Aktualisierungsverlauf) können Sie den Status Ihrer Anforderung bestimmen, indem

Sie sich die Spalte Status ansehen. Der Vorgang wechselt während der Initialisierung in den `Pending` Status. Der Status sollte sich dann schnell in `InProgress` ändern. Wenn alle Instances aktualisiert sind, ändert sich der Status in `Successful`.

4. Sie können den Erfolg oder Misserfolg laufender Aktivitäten weiter überwachen, indem Sie sich die Skalierungsaktivitäten der Gruppe ansehen. Auf der Registerkarte Activity (Aktivität) unter Activity history (Aktivitätsverlauf) werden beim Start der Instance-Aktualisierung Einträge angezeigt, wenn Instances beendet werden. Beim Starten von Instances werden weitere Einträge angezeigt. Wenn Sie zahlreiche Skalierungsaktivitäten haben, können Sie sich mehr davon anzeigen lassen, indem Sie oben im Aktivitätsverlauf auf das Symbol **>** klicken. Informationen zur Behebung von Problemen, die zum Fehlschlagen von Aktivitäten führen können, finden Sie unterFehlersuche bei Amazon EC2 Auto Scaling.
5. (Optional) Auf der Registerkarte Instanzverwaltung unter Instances können Sie bei Bedarf den Fortschritt bestimmter Instances überprüfen.

AWS CLI

Um den Status einer Instanzaktualisierung zu überwachen und zu überprüfen (AWS CLI)

Verwenden Sie den folgenden Befehl describe-instance-refreshes.

```
aws autoscaling describe-instance-refreshes --auto-scaling-group-name my-asg
```

Es folgt eine Beispielausgabe.

Instanzaktualisierungen werden nach der Startzeit sortiert. Instanzaktualisierungen, die noch im Gange sind, werden zuerst beschrieben.

```
{
  "InstanceRefreshes":[
    {
      "InstanceRefreshId":"08b91cf7-8fa6-48af-b6a6-d227f40f1b9b",
      "AutoScalingGroupName":"my-asg",
      "Status":"InProgress",
      "StatusReason":"Waiting for instances to warm up before continuing. For
 example: i-0645704820a8e83ff is warming up.",
      "StartTime":"2023-11-24T16:46:52+00:00",
      "PercentageComplete":50,
      "InstancesToUpdate":0,
      "Preferences":{
        "MaxHealthyPercentage":120,
```

```
        "MinHealthyPercentage":90,
        "InstanceWarmup":60,
        "SkipMatching":false,
        "AutoRollback":true,
        "ScaleInProtectedInstances":"Ignore",
        "StandbyInstances":"Ignore"
      }
    },
    {
      "InstanceRefreshId":"0e151305-1e57-4a32-a256-1fd14157c5ec",
      "AutoScalingGroupName":"my-asg",
      "Status":"Successful",
      "StartTime":"2023-11-22T13:53:37+00:00",
      "EndTime":"2023-11-22T13:59:45+00:00",
      "PercentageComplete":100,
      "InstancesToUpdate":0,
      "Preferences":{
        "MaxHealthyPercentage":120,
        "MinHealthyPercentage":90,
        "InstanceWarmup":60,
        "SkipMatching":false,
        "AutoRollback":true,
        "ScaleInProtectedInstances":"Ignore",
        "StandbyInstances":"Ignore"
      }
    }
  ]
}
```


Sie können den Erfolg oder Misserfolg laufender Aktivitäten weiter überwachen, indem Sie sich die Skalierungsaktivitäten der Gruppe ansehen. Die Skalierungsaktivitäten helfen Ihnen auch dabei, weitere Details zu finden, um Probleme bei einer Instance-Aktualisierung zu beheben. Weitere Informationen finden Sie unter Fehlersuche bei Amazon EC2 Auto Scaling.

Status von Instance-Aktualisierungen

Wenn Sie eine Instance-Aktualisierung starten, wechselt diese in den Status Ausstehend. Sie wechselt von „Ausstehend“ InProgresszu „Erfolgreich“, „Fehlgeschlagen“, „Storniert“ oder RollbackFailed. RollbackSuccessful

Eine Instance-Aktualisierung kann die folgenden Status haben:

Status	Description
Ausstehend	Die Anfrage wurde erstellt, aber die Instance-Aktualisierung wurde nicht gestartet.
InProgress	Eine Instance-Aktualisierung ist in Bearbeitung.
Erfolgreich	Eine Instance-Aktualisierung wurde erfolgreich abgeschlossen.
Fehlgeschlagen	Eine Instance-Aktualisierung konnte nicht abgeschlossen werden. Sie können Probleme mit dem Statusgrund und den Skalierungsaktivitäten beheben.
Abbrechen	Eine laufende Instance-Aktualisierung wird abgebrochen.
Abgebrochen	Die Instance-Aktualisierung wird abgebrochen.
RollbackInFortschritt	Eine Instance-Aktualisierung wird rückgängig gemacht.
RollbackFailed	Das Rollback konnte nicht abgeschlossen werden. Sie können Probleme mit dem Statusgrund und den Skalierungsaktivitäten beheben.
RollbackSuccessful	Das Rollback wurde erfolgreich abgeschlossen.

Abbrechen einer Instance-Aktualisierung

Sie können eine Instance-Aktualisierung abbrechen, die noch ausgeführt wird. Sie können den Vorgang nicht mehr abbrechen, nachdem er beendet ist.

Das Abbrechen einer Instance-Aktualisierung setzt keine Instances zurück, die bereits ersetzt wurden. Führen Sie stattdessen ein Rollback durch, um die Änderungen an Ihren Instances rückgängig zu machen. Weitere Informationen finden Sie unter Änderungen mit einem Rollback rückgängig machen.

Themen

- Abbrechen einer Instance-Aktualisierung (Konsole)
- Abbrechen einer Instance-Aktualisierung (AWS CLI)

Abbrechen einer Instance-Aktualisierung (Konsole)

So brechen Sie eine Instance-Aktualisierung ab

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.
3. Wählen Sie auf der Registerkarte Instance-Aktualisierung unter Aktive Instance-Aktualisierung die Option Aktionen und dann Abbrechen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Confirm (Bestätigen).

Der Status der Instance-Aktualisierung ist auf Abbrechen gesetzt. Nach Abschluss des Abbruchs wird der Status der Instance-Aktualisierung auf Abgebrochen gesetzt.

Abbrechen einer Instance-Aktualisierung (AWS CLI)

So brechen Sie eine Instance-Aktualisierung ab

Verwenden Sie den Befehl cancel-instance-refresh von AWS CLI und geben Sie den Namen der Auto Scaling Scaling-Gruppe an.

```
aws autoscaling cancel-instance-refresh --auto-scaling-group-name my-asg
```

Beispielausgabe:

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Änderungen mit einem Rollback rückgängig machen

Sie können eine Instance-Aktualisierung, die noch ausgeführt wird, rückgängig machen. Sie können den Vorgang nicht mehr rückgängig machen, nachdem er beendet ist. Sie können Ihre Auto-Scaling-Gruppe jedoch erneut aktualisieren, indem Sie eine neue Instance-Aktualisierung starten.

Beim Rollback ersetzt Amazon EC2 Auto Scaling die Instances, die bisher bereitgestellt wurden. Die neuen Instances entsprechen der letzten Konfiguration, die Sie in der Auto-Scaling-Gruppe gespeichert haben, bevor Sie mit der Instance-Aktualisierung begonnen haben.

Amazon EC2 Auto Scaling bietet die folgenden Möglichkeiten ein Rollback durchzuführen:

- Manuelles Rollback: Sie starten ein Rollback manuell, um das, was bis zum Rollback-Punkt bereitgestellt wurde, rückgängig zu machen.
- Automatisches Rollback: Amazon EC2 Auto Scaling macht automatisch rückgängig, was bereitgestellt wurde, wenn die Instance-Aktualisierung aus irgendeinem Grund fehlschlägt oder wenn von Ihnen angegebene CloudWatch Alarme in den Status wechseln. `ALARM`

Inhalt

Überlegungen

Die folgenden Überlegungen gelten für die Verwendung eines Rollbacks:

- Die Rollback-Option ist nur verfügbar, wenn Sie beim Starten einer Instance-Aktualisierung eine gewünschte Konfiguration angeben.
- Sie können nur dann zu einer früheren Version einer Startvorlage zurückkehren, wenn es sich bei der Version um eine bestimmte nummerierte Version handelt. Die Rollback-Option ist nicht verfügbar, wenn die Auto-Scaling-Gruppe so konfiguriert ist, dass sie die Startvorlagenversion `$Latest` oder `$Default` verwendet.
- Sie können auch nicht zu einer Startvorlage zurückkehren, die für die Verwendung eines AMI-Alias aus dem AWS Systems Manager Parameterspeicher konfiguriert ist.
- Die Konfiguration, die Sie zuletzt in der Auto-Scaling-Gruppe gespeichert haben, muss sich in einem stabilen Zustand befinden. Wenn er sich nicht in einem stabilen Zustand befindet, wird der Rollback-Workflow trotzdem ausgeführt, aber er wird letztendlich fehlschlagen. Bis Sie das Problem behoben haben, befindet sich die Auto-Scaling-Gruppe möglicherweise in einem fehlerhaften Status, in dem Instances nicht mehr erfolgreich gestartet werden können. Dies kann die Verfügbarkeit des Services oder der Anwendung beeinträchtigen.

Manuelles Starten eines Rollbacks

Console

So starten Sie manuell ein Rollback einer Instance-Aktualisierung (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.
3. Wählen Sie auf der Registerkarte Instance-Aktualisierung unter Aktive Instance-Aktualisierung die Optionen Aktionen und dann Rollback starten.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Confirm (Bestätigen).

AWS CLI

So starten Sie manuell ein Rollback einer Instance-Aktualisierung (AWS CLI)

Verwenden Sie den Befehl rollback-instance-refresh von der AWS CLI und geben Sie den Namen der Auto-Scaling-Gruppe an.

```
aws autoscaling rollback-instance-refresh --auto-scaling-group-name my-asg
```

Beispielausgabe:

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

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

Starten einer Instance-Aktualisierung mit automatischem Rollback

Mithilfe der auto Rollback-Funktion können Sie die Instance-Aktualisierung automatisch rückgängig machen, wenn sie fehlschlägt, z. B. wenn Fehler auftreten oder ein bestimmter CloudWatch Amazon-Alarm in den `ALARM` Status wechselt.

Wenn Sie das automatische Rollback aktivieren und beim Ersetzen von Instances Fehler auftreten, versucht die Instance-Aktualisierung eine Stunde lang, alle Ersetzungen abzuschließen, bevor sie fehlschlägt und ein Rollback erfolgt. Diese Fehler werden in der Regel dadurch verursacht, dass EC2-Startfehler, falsch konfigurierte Integritätsprüfungen oder das Nichtignorieren oder Beenden von Instances, die sich im Status `Standby` befinden oder vor dem Abskalieren geschützt sind, nicht ignoriert oder zugelassen werden.

Die Angabe von CloudWatch Alarmen ist optional. Um einen Alarm anzugeben, müssen Sie ihn zunächst erstellen. Sie können Metrikalarme und zusammengesetzte Alarme angeben. Informationen zum Erstellen des Alarms finden Sie im CloudWatch Amazon-Benutzerhandbuch. Wenn Sie beispielsweise Elastic Load Balancing-Metriken verwenden und einen Application Load Balancer verwenden, könnten Sie die Metriken `HTTPCode_ELB_5XX_Count` und `HTTPCode_ELB_4XX_Count` verwenden.

Überlegungen

- Wenn Sie einen CloudWatch Alarm angeben, aber kein auto Rollback aktivieren und der Alarmstatus auf wechselt`ALARM`, schlägt die Instanzaktualisierung ohne Rollback fehl.
- Sie können maximal 10 Alarme auswählen, wenn Sie eine Instance-Aktualisierung starten.
- Bei der Auswahl eines CloudWatch Alarms muss sich der Alarm in einem kompatiblen Zustand befinden. Wenn der Alarmstatus `INSUFFICIENT_DATA` oder `ALARM` ist, erhalten Sie eine Fehlermeldung, wenn Sie versuchen, die Instance-Aktualisierung zu starten.
- Wenn Sie einen Alarm für Amazon EC2 Auto Scaling erstellen, sollte der Alarm beinhalten, wie mit fehlenden Datenpunkten umzugehen ist. Wenn bei einer Metrik planmäßig häufig Datenpunkte fehlen, ist der Status des Alarms während dieser Zeiträume `INSUFFICIENT_DATA`. In diesem Fall kann Amazon EC2 Auto Scaling keine Instances ersetzen, bis neue Datenpunkte gefunden werden. Um den Alarm zu zwingen, den vorherigen Zustand `ALARM` oder `OK` beizubehalten, können Sie stattdessen fehlende Daten ignorieren. Weitere Informationen finden Sie unter Konfiguration der Behandlung fehlender Daten durch Alarme im CloudWatch Amazon-Benutzerhandbuch.

Console

So starten Sie eine Instance-Aktualisierung mit automatischem Rollback (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.
3. Wählen Sie auf der Registerkarte Instance refresh (Instance-Aktualisierung) unter Active instance refresh (Aktive Instance-Aktualisierung) die Option Start instance refresh (Instance-Aktualisierung starten) aus.
4. Befolgen Sie das Starten einer Instance-Aktualisierung (Konsole)-Verfahren und konfigurieren Sie die Einstellungen Ihrer Instance-Aktualisierungen nach Bedarf.
5. (Optional) Wählen Sie unter Einstellungen aktualisieren für CloudWatch Alarm die Option CloudWatch Alarme aktivieren und wählen Sie dann einen oder mehrere Alarme aus, um Probleme zu identifizieren und den Vorgang fehlschlagen zu lassen, wenn ein Alarm in den `ALARM` Status wechselt.
6. Wählen Sie unter Rollback-Einstellungen die Option Automatisches Rollback aus, um eine fehlgeschlagene Instance-Aktualisierung automatisch auf die Konfiguration zurückzusetzen, die Sie zuletzt in der Auto-Scaling-Gruppe gespeichert haben, bevor Sie mit der Instance-Aktualisierung begonnen haben.
7. Überprüfen Sie Ihre Auswahl und wählen Sie dann Instance-Aktualisierung starten.

AWS CLI

So starten Sie eine Instance-Aktualisierung mit automatischem Rollback (AWS CLI)

Verwenden Sie den Befehl start-instance-refresh und geben Sie `true` für die Option `AutoRollback` in `Preferences` an.

Das folgende Beispiel zeigt, wie eine Instance-Aktualisierung gestartet wird, die automatisch zurückgesetzt wird, wenn etwas fehlschlägt. Ersetzen Sie die *`italicized`* Parameterwerte durch eigene.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Inhalt von `config.json`.

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredConfiguration": {
      "LaunchTemplate": {
          "LaunchTemplateName": "my-launch-template",
          "Version": "1"
       }
    },
    "Preferences": {
      "AutoRollback": true
    }
}
```

Sie können auch ein automatisches Rollback durchführen, wenn die Instanzaktualisierung fehlschlägt oder wenn sich ein bestimmter CloudWatch Alarm im `ALARM` Status befindet, indem Sie die `AlarmSpecification` Option in der angeben `Preferences` und den Namen des Alarms angeben, wie im folgenden Beispiel. Ersetzen Sie die *`italicized`* Parameterwerte durch eigene.

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredConfiguration": {
      "LaunchTemplate": {
          "LaunchTemplateName": "my-launch-template",
          "Version": "1"
       }
    },
    "Preferences": {
      "AutoRollback": true,
      "AlarmSpecification": { "Alarms": [ "my-alarm" ] }
    }
}
```

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

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

Verwenden einer Instance-Aktualisierung mit Funktion zum Überspringen des Abgleichs

Das Überspringen des Abgleichs weist Amazon EC2 Auto Scaling an, Instances zu ignorieren, die bereits über Ihre neuesten Aktualisierungen verfügen. Auf diese Weise ersetzen Sie nicht mehr Instances als Sie benötigen. Dies ist hilfreich, wenn Sie sicherstellen möchten, dass Ihre Auto-Scaling-Gruppe eine bestimmte Version Ihrer Startvorlage verwendet und nur die Instances ersetzt, die eine andere Version verwenden.

Beachten Sie im Zusammenhang mit der Funktion zum Überspringen entsprechender Instances folgende Überlegungen:

- Wenn Sie eine Instance-Aktualisierung sowohl mit der Funktion zum Überspringen des Abgleichs als auch mit einer gewünschten Konfiguration starten, prüft Amazon EC2 Auto Scaling, ob Instances Ihrer gewünschten Konfiguration entsprechen. Anschließend werden nur die Instances ersetzt, die nicht Ihrer gewünschten Konfiguration entsprechen. Nach einer erfolgreichen Instance-Aktualisierung aktualisiert Amazon EC2 Auto Scaling die Gruppe, damit sie mit Ihrer gewünschten Konfiguration übereinstimmt.
- Wenn Sie eine Instance-Aktualisierung mit der Funktion zum Überspringen des Abgleichs starten, aber keine gewünschte Konfiguration angeben, prüft Amazon EC2 Auto Scaling, ob Instances mit der exakten Konfiguration übereinstimmen, die Sie zuletzt in der Auto-Scaling-Gruppe gespeichert haben. Anschließend ersetzt es nur die Instances, die nicht Ihrer zuletzt gespeicherten Konfiguration entsprechen.
- Sie können das Überspringen des Abgleichs mit einer neuen Startvorlage, einer neuen Version einer Startvorlage oder einem Satz von Instance-Typen verwenden. Wenn Sie das Überspringen des Abgleichs aktivieren, aber alles unverändert bleibt, ist die Instance-Aktualisierung sofort erfolgreich, ohne dass Instances ersetzt werden. Wenn Sie weitere Änderungen an Ihrer gewünschten Konfiguration vorgenommen haben (z. B. Änderungen an der Spot-Zuordnungsstrategie), wartet die Lösung Amazon EC2 Auto Scaling, bis die Instance-Aktualisierung erfolgreich ausgeführt wurde. Anschließend werden die Einstellungen der Auto-Scaling-Gruppe aktualisiert, sodass sie der neuen gewünschten Konfiguration entsprechen.

- Die Funktion zum Überspringen entsprechender Instances kann nicht mit einer neuen Startkonfiguration verwendet werden.
- Wenn Sie eine Instance-Aktualisierung starten und eine gewünschte Konfiguration angeben, stellt Amazon EC2 Auto Scaling sicher, dass alle Instances Ihre gewünschte Konfiguration verwenden. Wenn Sie also entweder `$Default` oder `$Latest` als gewünschte Version für Ihre Startvorlage angeben und dann während einer Instance-Aktualisierung eine neue Version der Startvorlage erstellen, werden alle Instances, die bereits ersetzt wurden, erneut ersetzt.
- Bei Skip Matching wird nicht ermittelt, ob ein Benutzerdatenskript in der Startvorlage aktualisierten Code abruft und ihn auf neuen Instances installiert. Aus diesem Grund überspringt Skip Matching möglicherweise das Ersetzen von Instanzen, auf denen veralteter Code installiert ist. In diesem Fall sollten Sie den Abgleich überspringen deaktivieren, um sicherzustellen, dass alle Instances Ihren neuesten Code erhalten, auch ohne ein Versionsupdate für die Startvorlage.

Dieser Abschnitt enthält AWS CLI Anweisungen zum Starten einer Instanzaktualisierung bei aktiviertem Skip-Matching. Weitere Informationen zur Verwendung der Konsole finden Sie unter Starten einer Instance-Aktualisierung (Konsole).

Überspringen des Abgleichs (einfaches Verfahren)

Folgen Sie den Schritten in diesem Abschnitt AWS CLI , um Folgendes zu tun:

- Erstellen einer Startvorlage, die Sie auf Ihre Instances anwenden möchten.
- Starten einer Instance-Aktualisierung, um Ihre Startvorlage auf Ihre Auto-Scaling-Gruppe anzuwenden. Wenn Sie das Überspringen des Abgleichs nicht aktivieren, werden alle Instances ersetzt. Dies gilt auch dann, wenn die Startvorlage, die zum Bereitstellen der Instance verwendet wird, dieselbe ist wie die, die Sie für Ihre gewünschte Konfiguration angegeben haben.

So verwenden Sie die Funktion zum Überspringen des Abgleichs mit einer neuen Startvorlage

1. Verwenden Sie den Befehl create-launch-template, um eine neue Startvorlage für Ihre Auto-Scaling-Gruppe zu erstellen. Schließen Sie die `--launch-template-data`-Option und die JSON-Eingabe ein, die die Details der Instances definiert, die für Ihre Auto-Scaling-Gruppe erstellt werden.

   Verwenden Sie beispielsweise den folgenden Befehl, um eine grundlegende Startvorlage mit der AMI-ID *ami-0123456789abcdef0* und dem Instance-Typ *t2.micro* zu erstellen.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling
 --version-description version1 \
  --launch-template-data
 '{"ImageId":"ami-0123456789abcdef0","InstanceType":"t2.micro"}'
```

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
   "LaunchTemplate": {
     "LaunchTemplateId": "lt-068f72b729example",
     "LaunchTemplateName": "my-template-for-auto-scaling",
     "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
     "CreateTime": "2023-01-30T18:16:06.000Z",
     "DefaultVersionNumber": 1,
     "LatestVersionNumber": 1
  }
}
```

Weitere Informationen finden Sie unter Beispiele für die Erstellung und Verwaltung von Startvorlagen mit dem AWS CLI.

2. Verwenden Sie den Befehl start-instance-refresh, um den Workflow zum Ersetzen einer Instance zu starten und Ihre neue Startvorlage mit der ID *lt-068f72b729example* anzuwenden. Da die Startvorlage neu ist, verfügt sie nur über eine Version. Dies bedeutet, dass die Version 1 der Startvorlage das Ziel dieser Instance-Aktualisierung ist. Falls während der Instance-Aktualisierung ein Aufskalierungsereignis auftritt und Amazon EC2 Auto Scaling neue Instances mit der Version 1 dieser Startvorlage bereitstellt, werden diese nicht ersetzt. Nach erfolgreichem Abschluss des Vorgangs wird die neue Startvorlage erfolgreich auf Ihre Auto-Scaling-Gruppe angewendet.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Inhalt von `config.json`.

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredConfiguration": {
      "LaunchTemplate": {
          "LaunchTemplateId": "lt-068f72b729example",
```

```
            "Version": "$Default"
        }
    },
    "Preferences": {
      "SkipMatching": true
    }
}
```

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Überspringen des Abgleichs (Gruppe mit gemischten Instances)

Wenn Sie über eine Auto Scaling Scaling-Gruppe mit einer Richtlinie für gemischte Instanzen verfügen, folgen Sie den Schritten in diesem Abschnitt, AWS CLI um eine Instanzaktualisierung mit Skip-Matching zu starten. Ihnen stehen folgende Optionen zur Verfügung:

- Stellen Sie eine neue Startvorlage bereit, die für alle in der Richtlinie angegebenen Instance-Typen gelten soll.
- Stellen Sie einen aktualisierten Satz von Instance-Typen mit oder ohne Änderung der Startvorlage in der Richtlinie bereit. Beispielsweise möchten Sie möglicherweise von unerwünschten Instance-Typen weg migrieren. Sie würden die Startvorlage unverändert verwenden, ohne das AMI, die Sicherheitsgruppen oder andere Besonderheiten der zu ersetzenden Instances zu ändern.

Befolgen Sie die Schritte in einem der folgenden Abschnitte, je nachdem, welche Option Ihren Anforderungen entspricht.

So verwenden Sie die Funktion zum Überspringen des Abgleichs mit einer neuen Startvorlage

1. Verwenden Sie den Befehl create-launch-template, um eine neue Startvorlage für Ihre Auto-Scaling-Gruppe zu erstellen. Schließen Sie die `--launch-template-data`-Option und die JSON-Eingabe ein, die die Details der Instances definiert, die für Ihre Auto-Scaling-Gruppe erstellt werden.

Verwenden Sie beispielsweise den folgenden Befehl, um eine Startvorlage mit der AMI-ID *ami-0123456789abcdef0* zu erstellen.

```
aws ec2 create-launch-template --launch-template-name my-new-template --version-
description version1 \
  --launch-template-data '{"ImageId":"ami-0123456789abcdef0"}'
```

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
   "LaunchTemplate": {
     "LaunchTemplateId": "lt-04d5cc9b88example",
     "LaunchTemplateName": "my-new-template",
     "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
     "CreateTime": "2023-01-31T15:56:02.000Z",
     "DefaultVersionNumber": 1,
     "LatestVersionNumber": 1
  }
}
```

Weitere Informationen finden Sie unter Beispiele für die Erstellung und Verwaltung von Startvorlagen mit dem AWS CLI.

2. Um die vorhandene Richtlinie für gemischte Instances Ihrer Auto-Scaling-Gruppe anzuzeigen, führen Sie den Befehl describe-auto-scaling-groups aus. Sie benötigen diese Informationen im nächsten Schritt, wenn Sie die Instance-Aktualisierung starten.

   Der folgende Beispielbefehl gibt die Richtlinie für gemischte Instances zurück, die für die benannte Auto-Scaling-Gruppe konfiguriert ist *my-asg*.

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
   ```

   Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

   ```
   {
     "AutoScalingGroups":[
       {
         "AutoScalingGroupName":"my-asg",
         "AutoScalingGroupARN":"arn",
         "MixedInstancesPolicy":{
   ```

```
        "LaunchTemplate":{
          "LaunchTemplateSpecification":{
            "LaunchTemplateId":"lt-073693ed27example",
            "LaunchTemplateName":"my-old-template",
            "Version":"$Default"
          },
          "Overrides":[
            {
              "InstanceType":"c5.large"
            },
            {
              "InstanceType":"c5a.large"
            },
            {
              "InstanceType":"m5.large"
            },
            {
              "InstanceType":"m5a.large"
            }
          ]
        },
        "InstancesDistribution":{
          "OnDemandAllocationStrategy":"prioritized",
          "OnDemandBaseCapacity":1,
          "OnDemandPercentageAboveBaseCapacity":50,
          "SpotAllocationStrategy":"price-capacity-optimized"
        }
      },
      "MinSize":1,
      "MaxSize":5,
      "DesiredCapacity":4,
      ...
    }
  ]
}
```

3. Verwenden Sie den Befehl start-instance-refresh, um den Workflow zum Ersetzen einer Instance zu starten und Ihre neue Startvorlage mit der ID *lt-04d5cc9b88example* anzuwenden. Da die Startvorlage neu ist, verfügt sie nur über eine Version. Dies bedeutet, dass die Version 1 der Startvorlage das Ziel dieser Instance-Aktualisierung ist. Falls während der Instance-Aktualisierung ein Aufskalierungsereignis auftritt und Amazon EC2 Auto Scaling neue Instances mit der Version 1 dieser Startvorlage bereitstellt, werden diese nicht ersetzt. Nach erfolgreichem

Abschluss des Vorgangs wird die aktualisierte Richtlinie für gemischte Instances erfolgreich auf Ihre Auto-Scaling-Gruppe angewendet.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Inhalt von `config.json`.

```
{
  "AutoScalingGroupName":"my-asg",
  "DesiredConfiguration":{
    "MixedInstancesPolicy":{
      "LaunchTemplate":{
        "LaunchTemplateSpecification":{
          "LaunchTemplateId":"lt-04d5cc9b88example",
          "Version":"$Default"
        },
        "Overrides":[
          {
            "InstanceType":"c5.large"
          },
          {
            "InstanceType":"c5a.large"
          },
          {
            "InstanceType":"m5.large"
          },
          {
            "InstanceType":"m5a.large"
          }
        ]
      },
      "InstancesDistribution":{
        "OnDemandAllocationStrategy":"prioritized",
        "OnDemandBaseCapacity":1,
        "OnDemandPercentageAboveBaseCapacity":50,
        "SpotAllocationStrategy":"price-capacity-optimized"
        }
      }
    }
  },
  "Preferences":{
    "SkipMatching":true
```

```
  }
}
```

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

In diesem nächsten Verfahren stellen Sie einen aktualisierten Satz von Instance-Typen bereit, ohne die Startvorlage zu ändern.

So verwenden Sie die Funktion zum Überspringen des Abgleichs mit einem aktualisierten Satz von Instance-Typen

1. Um die vorhandene Richtlinie für gemischte Instances Ihrer Auto-Scaling-Gruppe anzuzeigen, führen Sie den Befehl describe-auto-scaling-groups aus. Sie benötigen diese Informationen im nächsten Schritt, wenn Sie die Instance-Aktualisierung starten.

   Der folgende Beispielbefehl gibt die Richtlinie für gemischte Instances zurück, die für die benannte Auto-Scaling-Gruppe konfiguriert ist *my-asg*.

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
   ```

   Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

   ```
   {
     "AutoScalingGroups":[
       {
         "AutoScalingGroupName":"my-asg",
         "AutoScalingGroupARN":"arn",
         "MixedInstancesPolicy":{
           "LaunchTemplate":{
             "LaunchTemplateSpecification":{
               "LaunchTemplateId":"lt-073693ed27example",
               "LaunchTemplateName":"my-template-for-auto-scaling",
               "Version":"$Default"
             },
             "Overrides":[
               {
   ```

```
            "InstanceType":"c5.large"
          },
          {
            "InstanceType":"c5a.large"
          },
          {
            "InstanceType":"m5.large"
          },
          {
            "InstanceType":"m5a.large"
          }
        ]
      },
      "InstancesDistribution":{
        "OnDemandAllocationStrategy":"prioritized",
        "OnDemandBaseCapacity":1,
        "OnDemandPercentageAboveBaseCapacity":50,
        "SpotAllocationStrategy":"price-capacity-optimized"
      }
    },
    "MinSize":1,
    "MaxSize":5,
    "DesiredCapacity":4,
    ...
  }
 ]
}
```

2. Verwenden Sie den Befehl start-instance-refresh, um den Workflow zum Ersetzen der Instance zu starten und Ihre Aktualisierungen anzuwenden. Wenn Sie Instances ersetzen möchten, die bestimmte Instance-Typen verwenden, muss Ihre gewünschte Konfiguration die Richtlinie für gemischte Instances nur mit den gewünschten Instance-Typen angeben. Sie können wählen, ob Sie stattdessen neue Instance-Typen hinzufügen möchten.

   Der folgende Beispielbefehl startet eine Instance-Aktualisierung ohne den unerwünschten Instance-Typ *`m5a.large`*. Wenn ein Instance-Typ in Ihrer Gruppe nicht mit einem der übrigen drei Instance-Typen übereinstimmt, werden die Instances ersetzt. (Berücksichtigen Sie, dass eine Instance-Aktualisierung nicht die Instance-Typen auswählt, aus denen die neuen Instances bereitgestellt werden sollen. Dies wird stattdessen von den Zuweisungsstrategien erledigt.) Nach erfolgreichem Abschluss des Vorgangs wird die aktualisierte Richtlinie für gemischte Instances erfolgreich auf Ihre Auto-Scaling-Gruppe angewendet.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Inhalt von `config.json`

```
{
  "AutoScalingGroupName":"my-asg",
  "DesiredConfiguration":{
    "MixedInstancesPolicy":{
      "LaunchTemplate":{
        "LaunchTemplateSpecification":{
          "LaunchTemplateId":"lt-073693ed27example",
          "Version":"$Default"
        },
        "Overrides":[
          {
            "InstanceType":"c5.large"
          },
          {
            "InstanceType":"c5a.large"
          },
          {
            "InstanceType":"m5.large"
          }
        ]
      },
      "InstancesDistribution":{
        "OnDemandAllocationStrategy":"prioritized",
        "OnDemandBaseCapacity":1,
        "OnDemandPercentageAboveBaseCapacity":50,
        "SpotAllocationStrategy":"price-capacity-optimized"
        }
      }
    }
  },
  "Preferences":{
    "SkipMatching":true
  }
}
```

Prüfpunkte zu einer Instance-Aktualisierung hinzufügen

Wenn Sie eine Instance-Aktualisierung verwenden, können Sie Instances phasenweise ersetzen, damit Sie bei laufendem Betrieb Überprüfungen für Ihre Instances durchführen können. Um eine schrittweise Ersetzung durchzuführen, fügen Sie Checkpoints hinzu. Dies sind Zeitpunkte, an denen die Instance-Aktualisierung pausiert wird. Die Verwendung von Prüfpunkten gibt Ihnen eine bessere Kontrolle darüber, wie Sie Ihre Auto-Scaling-Gruppe aktualisieren. Damit können Sie bestätigen, dass Ihre Anwendung zuverlässig und vorhersehbar funktioniert.

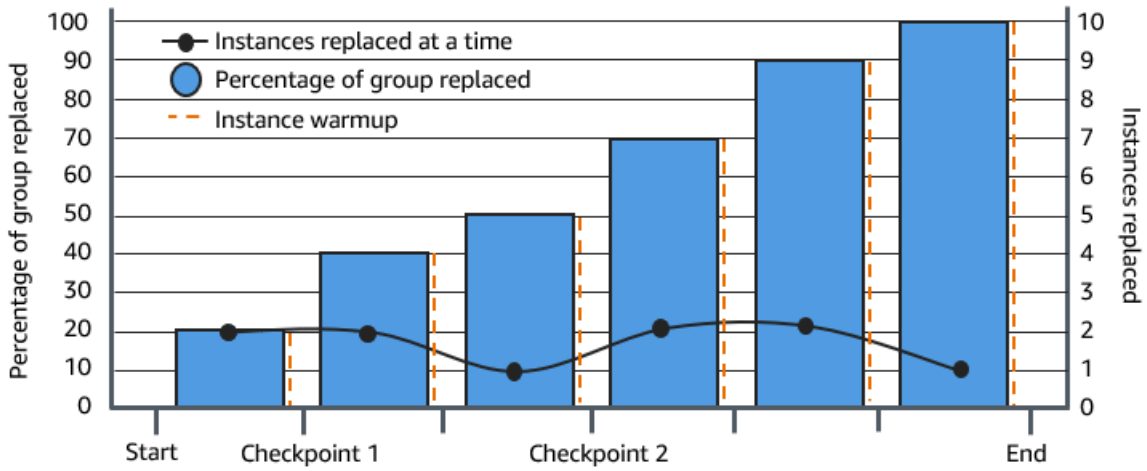
Inhalt

Funktionsweise

Wenn Sie eine Instanzaktualisierung starten, geben Sie Checkpoints als Prozentsätze der Gesamtzahl der Instances in der Auto Scaling Scaling-Gruppe an. Diese Checkpoints geben den Mindestprozentsatz der Instances in der Auto Scaling Scaling-Gruppe an, bei denen es sich um neue Instances handeln muss, bevor der Checkpoint als erreicht gilt. Wenn Ihre Checkpoints beispielsweise so sind`[20, 50, 100]`, ist der erste Checkpoint erreicht, wenn 20 Prozent der Instances neu sind, der zweite, wenn 50 Prozent neu sind, und der letzte Checkpoint, wenn alle Instances neu sind.

Amazon EC2 Auto Scaling passt Instance-Ersetzungen so an, dass sie die angegebenen Checkpoint-Prozentsätze einhalten und gleichzeitig den minimalen fehlerfreien Prozentsatz der Gruppe beibehalten. Um einen Checkpoint-Prozentsatz zu erreichen, ersetzt Amazon EC2 Auto Scaling manchmal weniger fest, aber nie mehr, als der minimale fehlerfreie Prozentsatz zulässt.

Stellen Sie sich die folgende Auto-Scaling-Gruppe mit 10 Instanzen vor. Die Prozentsätze der Kontrollpunkte sind `[20,50,100]`, der minimale fehlerfreie Prozentsatz ist 80 Prozent, und der maximale fehlerfreie Prozentsatz ist 100 Prozent. Um den minimalen fehlerfreien Prozentsatz aufrechtzuerhalten, können nur zwei Instances auf einmal ersetzt werden. Im folgenden Diagramm ist der Prozess zum Ersetzen aller Instances dargestellt, bevor ein Checkpoint erreicht wird.



Im obigen Beispiel gibt es für jede neue Instance, die gestartet wird, eine Instance-Aufwärmphase. Möglicherweise haben Sie auch einen Lebenszyklus-Hook, der eine Instance in einen Wartestatus versetzt und dann eine benutzerdefinierte Aktion ausführt, während sie gestartet oder beendet wird.

Amazon EC2 Auto Scaling gibt Ereignisse für jeden Checkpoint aus, mit Ausnahme des Checkpoints, der zu 100 Prozent abgeschlossen ist. Sie können eine EventBridge Regel hinzufügen, um die Ereignisse an ein Ziel wie Amazon SNS zu senden. So werden Sie benachrichtigt, wenn Sie die erforderlichen Überprüfungen durchführen können. Weitere Informationen finden Sie unter Erstellen Sie EventBridge Regeln für Instance-Aktualisierungsereignisse.

Überlegungen

Behalten Sie bei der Verwendung von Prüfpunkten die folgenden Überlegungen im Auge:

- Da Prüfpunkte auf Prozentsätzen basieren, ändert sich die Anzahl der zu ersetzenden Instances mit der Größe der Gruppe. Bei einer Aufskalierung und wenn die Größe der Gruppe zunimmt könnte eine laufende Operation wieder einen Prüfpunkt erreichen. In diesem Fall sendet Amazon EC2 Auto Scaling eine weitere Benachrichtigung und wiederholt die Wartezeit zwischen den Checkpoints, bevor Sie fortfahren.
- Unter bestimmten Umständen ist es möglich, einen Checkpoint zu überspringen. Angenommen, Ihre Auto-Scaling-Gruppe hat zwei Instances und Ihre Prüfpunkt-Prozentsätze sind `[10,40,100]`. Nachdem die erste Instance ersetzt wurde, berechnet Amazon EC2 Auto Scaling, dass 50 Prozent der Gruppe ersetzt wurden. Da 50 Prozent höher ist als die ersten beiden Prüfpunkte, überspringt es den ersten Prüfpunkt (`10`) und sendet eine Benachrichtigung für den zweiten Prüfpunkt (`40`).
- Wenn Sie den Vorgang abbrechen, werden alle weiteren Ersetzungen beendet. Wenn Sie den Vorgang abbrechen oder er vor dem Erreichen des letzten Checkpoints fehlschlägt, werden alle Instances, die bereits ersetzt wurden, nicht auf die vorherige Konfiguration zurückgesetzt.

- Bei einer teilweisen Aktualisierung startet Amazon EC2 Auto Scaling beim erneuten Ausführen des Vorgangs nicht ab dem Zeitpunkt des letzten Checkpoints neu und stoppt nicht, wenn nur die älteren Instances ersetzt werden. Es wird jedoch zuerst ältere Instances ersetzen, bevor es neue Instances ersetzt.
- Der tatsächliche Prozentsatz, der abgeschlossen ist, kann höher sein als der Prozentsatz für diesen Checkpoint, wenn der Prozentsatz des Checkpoints im Verhältnis zur Anzahl der Instances in der Gruppe zu niedrig ist. Nehmen wir zum Beispiel an, der Prozentsatz des Checkpoints liegt bei 20 Prozent und die Gruppe hat vier Instances. Wenn Amazon EC2 Auto Scaling eine der vier Instances ersetzt, ist der tatsächliche Prozentsatz, der ersetzt wurde (25 Prozent), höher als der Prozentsatz des Checkpoints (20 Prozent).
- Nachdem ein Checkpoint erreicht wurde, wird der angezeigte Gesamtprozentsatz für abgeschlossen erst aktualisiert, nachdem die Instances den Warmlauf abgeschlossen haben. Ihre Checkpoint-Prozentsätze weisen beispielsweise eine Verzögerung von 15 Minuten und einen fehlerfreien Mindestprozentsatz von 80 Prozent auf. `[20,50]` Ihre Auto Scaling Scaling-Gruppe hat 10 Instances und nimmt die folgenden Ersetzungen vor:
  - 0:00: Zwei ältere Instances werden durch neue ersetzt.
  - 0:10: Zwei neue Instances schließen das Aufwärmen ab.
  - 0:25: Zwei ältere Instances werden durch neue ersetzt. (Damit der minimale fehlerfreie Prozentsatz beibehalten wird, werden nur zwei Instances ersetzt.)
  - 0:35: Zwei neue Instances schließen das Aufwärmen ab.
  - 0:35: Eine ältere Instance wird durch eine neue ersetzt.
  - 0:45: Eine neue Instance schließt das Aufwärmen ab.

  Bei 0:35 hört der Vorgang auf, neue Instance zu starten. Der abgeschlossene Prozentsatz spiegelt die Anzahl der abgeschlossenen Ersetzungen noch nicht genau wider (50 Prozent), da die neue Instance nicht aufgewärmt ist. Nachdem die neue Instance ihre Aufwärmphase um 0:45 Uhr abgeschlossen hat, wird für den Prozentsatz „Abgeschlossen“ ein Wert von 50 Prozent angezeigt.

Aktivieren von Prüfpunkten (Konsole)

Sie können Prüfpunkte aktivieren, bevor Sie eine Instance-Aktualisierung starten, um Instances mit einem inkrementellen oder schrittweisen Ansatz zu ersetzen. Dies bietet zusätzliche Zeit für die Überprüfung.

So starten Sie eine Instance-Aktualisierung, die Checkpoints verwendet

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Unten auf der Seite Auto-Scaling-Gruppen wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Instance refresh (Instance-Aktualisierung) unter Active instance refresh (Aktive Instance-Aktualisierung) die Option Start instance refresh (Instance-Aktualisierung starten) aus.
4. Auf der Seite Start instance refresh (Instance-Aktualisierung starten) geben Sie die Werte für Minimum healthy percentage (Minimaler gesunder Prozentsatz) und Instance warmup (Instance-Aufwärmphase) ein.
5. Aktivieren Sie das Kontrollkästchen Enable checkpoints (Prüfpunkte aktivieren).

   Dadurch wird ein Feld angezeigt, in dem Sie den prozentualen Schwellenwert für den ersten Checkpoint definieren können.
6. Für Proceed until ____ % of the group is refreshed (Fortfahren, bis ____% der Gruppe aktualisiert wurde), geben Sie eine Zahl ein (1-100). Dies legt den Prozentsatz für den ersten Prüfpunkt fest.
7. Um einen weiteren Checkpoint hinzuzufügen, wählen Sie Hinzufügen eines Checkpoints aus und definieren Sie dann den Prozentsatz für den nächsten Checkpoint.
8. Um anzugeben, wie lange Amazon EC2 Auto Scaling wartet, nachdem ein Checkpoint erreicht wurde, aktualisieren Sie die Felder in Zwischen Checkpoints **`1 hour`** warten. Die Zeiteinheit kann Stunden, Minuten oder Sekunden sein.
9. Wenn Sie mit Ihrer Auswahl für die Instance-Aktualisierung fertig sind, klicken Sie auf Instance-Aktualisierung Starten.

Aktivieren von Prüfpunkten (AWS CLI)

Um eine Instanzaktualisierung mit aktivierten Checkpoints mithilfe von zu starten AWS CLI, benötigen Sie eine Konfigurationsdatei, die die folgenden Parameter definiert:

- `CheckpointPercentages`: Gibt Schwellenwerte für den Prozentsatz der zu ersetzenden Instances an. Diese Schwellenwerte stellen die Checkpoints zur Verfügung. Wenn der Prozentsatz der ersetzten und aufgewärmten Instances einen der angegebenen Schwellenwerte erreicht, wartet der Vorgang eine bestimmte Dauer. Geben Sie die Wartezeit im `CheckpointDelay` in Sekunden

an. Wenn der angegebene Zeitraum abgelaufen ist, wird die Instance-Aktualisierung fortgesetzt, bis sie den nächsten Checkpoint erreicht (falls zutreffend).

- `CheckpointDelay`: Gibt die Dauer in Sekunden an, die nach Erreichen eines Prüfpunkts gewartet wird, bevor fortgefahren wird. Wählen Sie einen Zeitraum, der genügend Zeit für die Durchführung Ihrer Überprüfungen bietet.

Der letzt im `CheckpointPercentages`-Array gezeigte Wert ist der Prozentsatz der Auto-Scaling-Gruppe, der erfolgreich ersetzt werden muss. Der Vorgang wechselt zu, `Successful` nachdem dieser Prozentsatz der Gruppe erfolgreich ersetzt wurde und jede Instance als abgeschlossen gilt.

So erstellen Sie mehrere Checkpoints

Um mehrere Checkpoints zu erstellen, verwenden Sie den folgenden start-instance-refresh-Beispielbefehl. In diesem Beispiel wird eine Instance-Aktualisierung konfiguriert, die zunächst ein Prozent der Auto-Scaling–Gruppe aktualisiert. Nach zehn Minuten Wartezeit aktualisiert sie dann die nächsten 19 Prozent und wartet weitere zehn Minuten. Schließlich aktualisiert es den Rest der Gruppe, bevor der Vorgang abgeschlossen wird.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Inhalt von `config.json`:

```
{
    "AutoScalingGroupName": "my-asg",
    "Preferences": {
      "InstanceWarmup": 60,
      "MinHealthyPercentage": 80,
      "CheckpointPercentages": [1,20,100],
      "CheckpointDelay": 600
    }
}
```

So erstellen Sie einen einzelnen Checkpoint

Um einen einzigen Checkpointszu erstellen, verwenden Sie den folgenden start-instance-refresh-Beispielbefehl. In diesem Beispiel wird eine Instance-Aktualisierung konfiguriert, die zunächst 20 Prozent der Auto-Scaling-Gruppe aktualisiert. Nach zehn Minuten Wartezeit aktualisiert sie den Rest der Gruppe, bevor der Vorgang abgeschlossen wird.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Inhalt von `config.json`:

```
{
    "AutoScalingGroupName": "my-asg",
    "Preferences": {
      "InstanceWarmup": 60,
      "MinHealthyPercentage": 80,
      "CheckpointPercentages": [20,100],
      "CheckpointDelay": 600
    }
}
```

So aktuslieren Sie die Auto-Scaling-Gruppe teilweise

Um nur einen Teil Ihrer Auto-Scaling-Gruppe zu ersetzen und dann vollständig zu stoppen, verwenden Sie den folgenden start-instance-refresh-Beispielbefehl. In diesem Beispiel wird eine Instance-Aktualisierung konfiguriert, die zunächst ein Prozent der Auto-Scaling–Gruppe aktualisiert. Nach zehn Minuten Wartezeit aktualisiert sie die nächsten 19 Prozent, bevor der Vorgang abgeschlossen wird.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Inhalt von `config.json`:

```
{
    "AutoScalingGroupName": "my-asg",
    "Preferences": {
      "InstanceWarmup": 60,
      "MinHealthyPercentage": 80,
      "CheckpointPercentages": [1,20],
      "CheckpointDelay": 600
    }
}
```

Auto-Scaling-Instances basierend auf der maximalen Instance-Lebensdauer ersetzen

Die maximale Lebensdauer der Instance gibt die maximale Zeit (in Sekunden) an, die eine Instance in Betrieb sein kann, bevor sie beendet und ersetzt wird. Ein häufiger Anwendungsfall könnte eine Anforderung sein, Instances aufgrund interner Sicherheitsrichtlinien oder externer Compliance-Kontrollen nach einem Zeitplan zu ersetzen.

Sie müssen einen Wert von mindestens 86.400 Sekunden (ein Tag) angeben. Um einen zuvor festgelegten Wert zu löschen, geben Sie den neuen Wert „0“ an. Diese Einstellung gilt für alle aktuellen und zukünftigen Instances in Ihrer Auto-Scaling-Gruppe.

Inhalt

Überlegungen

Bei der Verwendung dieser Funktion sollten Sie Folgendes beachten:

- Wenn eine ältere Instance ersetzt und eine neue Instance gestartet wird, verwendet die neue Instance die Startvorlage oder Startkonfiguration, die derzeit der Auto-Scaling-Gruppe zugeordnet ist. Wenn Ihre Startvorlage oder Startkonfiguration die Amazon Machine Image (AMI) -ID einer anderen Version Ihrer Anwendung angibt, wird diese Version Ihrer Anwendung automatisch bereitgestellt.
- Wenn Sie die maximale Instance-Lebensdauer zu niedrig einstellen, können Instances schneller als gewünscht ersetzt werden. Amazon EC2 Auto Scaling ersetzt die Instances normalerweise einzeln, mit einer Pause zwischen den Ersetzungen. Wenn die angegebene maximale Instance-Lebensdauer jedoch nicht genügend Zeit bietet, um jede Instance einzeln zu ersetzen, muss Amazon EC2 Auto Scaling mehr als eine Instance gleichzeitig ersetzen. Es können mehrere Instances gleichzeitig ersetzt werden, bis zu 10 Prozent der aktuellen Kapazität Ihrer Auto-Scaling-Gruppe. Um zu vermeiden, dass zu viele Instances gleichzeitig ersetzt werden, legen Sie entweder eine längere maximale Instance-Lebensdauer fest oder verwenden Sie den Instance-Scale-In-Schutz, um vorübergehend zu verhindern, dass einzelne Instances beendet werden. Weitere Informationen finden Sie unter Instance-Abskalierungsschutz verwenden.

- Standardmäßig erstellt Amazon EC2 Auto Scaling eine neue Skalierung zum Beenden der fehlerhaften Instance und beendet diese anschließend. Während die Instance beendet wird, startet eine andere Skalierungsaktivität eine neue Instance. Sie können dieses Verhalten so ändern, dass es vor dem Beenden gestartet wird, indem Sie eine Instance-Wartungsrichtlinie verwenden. Weitere Informationen finden Sie unter Wartungsrichtlinien für Instances.

Maximale Lebensdauer von Instances festlegen

Wenn Sie eine Auto-Scaling-Gruppe in der Konsole erstellen, können Sie die Einstellung für die maximale Lebensdauer der Instance nicht festlegen. Nachdem die Gruppe erstellt wurde, können Sie sie jedoch bearbeiten, um die maximale Instance-Lebensdauer festzulegen.

So legen Sie die maximale Instance-Lebensdauer für eine Gruppe fest (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Unten auf der Seite Auto-Scaling-Gruppen wird ein geteilter Bereich geöffnet, in dem Informationen zur Gruppe, die Sie ausgewählt haben, angezeigt werden.
3. Wählen Sie auf der Registerkarte Details die Option Erweiterte Konfigurationen, Bearbeiten.
4. Geben Sie bei Maximale Lebensdauer der Instance die maximale Anzahl von Sekunden ein, die eine Instance in Betrieb sein kann.
5. Wählen Sie Aktualisieren.

Auf der Registerkarte Activity (Aktivität) können Sie unter Activity history (Aktivitätsverlauf) während des gesamten Verlaufs die Ersetzung von Instances für die Gruppe anzeigen.

So legen Sie die maximale Instance-Lebensdauer für eine Gruppe fest (AWS CLI)

Sie können den auch verwenden AWS CLI , um die maximale Instanzlebensdauer für neue oder bestehende Auto Scaling Scaling-Gruppen festzulegen.

Für neue Auto-Scaling-Gruppen verwenden Sie den create-auto-scaling-group-Befehl.

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Im Folgenden finden Sie eine `config.json`-Beispieldatei, in der eine maximale Instance-Lebensdauer von `2592000` Sekunden (30 Tage) angegeben ist.

```
{
    "AutoScalingGroupName": "my-asg",
    "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Default"
    },
    "MinSize": 1,
    "MaxSize": 5,
    "MaxInstanceLifetime": 2592000,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
    "Tags": []
}
```

Verwenden Sie für vorhandene Auto-Scaling-Gruppen den update-auto-scaling-group-Befehl.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-existing-asg --
max-instance-lifetime 2592000
```

So überprüfen Sie die maximale Instance-Lebensdauer für eine Auto-Scaling-Gruppe

Verwenden Sie den describe-auto-scaling-groups-Befehl.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Einschränkungen

- Die maximale Lebensdauer ist nicht für jede Instance genau garantiert: Es ist nicht garantiert, dass die Instances erst am Ende ihrer maximalen Lebensdauer ersetzt werden. In einigen Situationen muss Amazon EC2 Auto Scaling möglicherweise sofort mit dem Ersetzen von Instances beginnen, nachdem Sie den Parameter für die maximale Lebensdauer der Instance aktualisiert haben. Der Grund für dieses Verhalten ist, dass nicht alle Instances gleichzeitig ersetzt werden sollen.
- Instance Scale-In Protection ausgezeichnet: Amazon EC2 Auto Scaling bietet Instance-Scale-In-Schutz, mit dem Sie kontrollieren können, welche Instances beendet werden können. Wenn dieser Schutz auf einer Instance aktiviert ist, beendet Amazon EC2 Auto Scaling die Instance nicht, auch wenn sie ihre maximale Instance-Lebensdauer erreicht hat.

- Instances, die vor dem Start beendet wurden: Wenn es nur eine Instance in der Auto-Scaling-Gruppe gibt, kann das Feature der maximalen Instance-Lebensdauer zu einem Ausfall führen, da Amazon EC2 Auto Scaling eine Instance beendet und dann standardmäßig eine neue Instance startet. Um dieses Verhalten so zu ändern, dass der Start vor dem Beenden erfolgt, siehe Wartungsrichtlinien für Instances

Skalieren der Größe Ihrer Auto-Scaling-Gruppe

Skalierung ist die Möglichkeit, die Rechenkapazität Ihrer Anwendung zu erhöhen oder zu verringern. Die Skalierung beginnt mit einem Ereignis bzw. einer Skalierungsaktion, die eine Auto-Scaling-Gruppe anweist, Amazon EC2-Instances entweder zu starten oder zu beenden.

Amazon EC2 Auto Scaling bietet zur bestmöglichen Erfüllung der Anforderungen Ihrer Anwendungen eine Reihe von Möglichkeiten, mit denen Sie die Skalierung anpassen können. Daher ist es wichtig, dass Sie die Anforderungen Ihrer Anwendung gut kennen. Beachten Sie folgende Überlegungen:

- Welche Rolle sollte Amazon EC2 Auto Scaling in der Architektur Ihrer Anwendung spielen? In der Regel wird das Auto Scaling primär als Möglichkeit zur Vergrößerung bzw. Reduzierung von Kapazität betrachtet, Sie können damit jedoch auch dafür sorgen, dass immer dieselbe Anzahl an Servern verwendet wird.
- Welche Kosteneinschränkungen sind für Sie wichtig? Da Amazon EC2 Auto Scaling EC2-Instances verwendet, zahlen Sie nur für die Ressourcen, die Sie nutzen. Die Kenntnis Ihrer Kosteneinschränkungen hilft Ihnen bei der Entscheidung, wann und in welchem Ausmaß Sie Ihre Anwendungen skalieren möchten.
- Welche Metriken sind für Ihre Anwendung wichtig? Amazon CloudWatch unterstützt eine Reihe verschiedener Metriken, die Sie mit Ihrer Auto Scaling Scaling-Gruppe verwenden können.

Inhalt

Wählen Sie Ihre Skalierungsmethode aus

Amazon EC2 Auto Scaling bietet Ihnen mehrere Möglichkeiten zur Skalierung Ihrer Auto-Scaling-Gruppe.

Eine feste Anzahl von Instances beibehalten

Standardmäßig hat eine Auto-Scaling-Gruppe keine angehängten Skalierungsrichtlinien oder geplanten Aktionen, sodass sie eine feste Größe beibehält. Sobald Sie Ihre Auto-Scaling-Gruppe erstellt haben, beginnt sie mit dem Start von genügend Instances für die gewünschte Kapazität. Sind der Gruppe keine Skalierungsbedingungen zugewiesen, erhält sie ihre gewünschte Kapazität auch dann aufrecht, wenn eine Instance fehlerhaft wird. Amazon EC2 Auto Scaling überwacht den Zustand jeder einzelnen Instance in Ihrer Auto-Scaling-Gruppe. Wenn festgestellt wird, dass eine Instance fehlerhaft geworden ist, wird sie durch eine neue Instance ersetzt. Eine ausführlichere Beschreibung dieses Prozesses finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Manuelles Skalieren

Die manuelle Skalierung ist die einfachste Möglichkeit zur Skalierung Ihrer Auto-Scaling-Gruppe. Sie können entweder die gewünschte Kapazität der Auto Scaling Scaling-Gruppe aktualisieren oder Instances in der Auto Scaling Scaling-Gruppe beenden. Weitere Informationen finden Sie unter Manuelle Skalierung für Amazon EC2 Auto Scaling.

Skalierung nach Zeitplan

Skalierung nach Zeitplan bedeutet, dass Skalierungsaktionen automatisch in Abhängigkeit von Datum und Uhrzeit ausgeführt werden. Dies ist nützlich, wenn Sie genau wissen, wann die Anzahl der Instances in Ihrer Gruppe vergrößert oder verringert werden müssen, weil der Bedarf nach vorhersehbaren Mustern entsteht. Weitere Informationen finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling.

Dynamische Skalierung je nach Bedarf

Die dynamische Skalierung ist eine fortgeschrittenere Skalierung von Ressourcen, die es Ihnen ermöglicht, eine Skalierungsrichtlinie zu definieren, die dynamisch die Größe Ihrer Auto-Scaling-Gruppe an die Bedarfsänderungen angepasst. Nehmen wir beispielsweise an, Sie haben eine Webanwendung, die derzeit auf zwei Instances ausgeführt wird, und Sie möchten, dass die CPU-Auslastung der Auto-Scaling-Gruppe bei etwa 50 Prozent bleibt, wenn die Last der Anwendung sich ändert. Diese Methode eignet sich für die Skalierung bei Verkehrsänderungen, wenn Sie nicht

wissen, wann sich der Verkehr ändern wird. Sie können die Skalierungsrichtlinien so konfigurieren, dass sie für Sie reagieren. Es gibt mehrere Richtlinientypen (oder eine Kombination davon), die Sie verwenden können, um auf Verkehrsänderungen zu reagieren. Weitere Informationen finden Sie unter Dynamische Skalierung für Amazon EC2 Auto Scaling.

Proaktiv skalieren

Sie können auch prädiktive und dynamische Skalierung (proaktiver bzw. reaktiver Ansatz) kombinieren, um Ihre EC2-Kapazität schneller zu skalieren. Verwenden Sie die prädiktive Skalierung, um die Anzahl der EC2-Instances in Ihrer Auto-Scaling-Gruppe vor täglichen und wöchentlichen Mustern in Datenverkehrsflüssen zu erhöhen. Weitere Informationen finden Sie unter Prädiktive Skalierung für Amazon EC2 Auto Scaling.

Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe

Skalierungslimits stellen die gewünschte minimale und maximale Gruppengröße für Ihre Auto-Scaling-Gruppe dar. Sie legen Grenzwerte für die Mindest- und die Höchstgröße separat fest.

Die gewünschte Kapazität der Gruppe kann auf einen Wert innerhalb des Bereichs zwischen der Mindest- und Maximalgröße festgelegt werden. Die gewünschte Kapazität darf die Mindestgröße der Gruppe nicht unterschreiten und die maximale Gruppengröße nicht übersteigen.

- Desired capacity (Gewünschte Kapazität): Stellt die Anfangskapazität der Auto-Scaling-Gruppe zum Zeitpunkt der Erstellung dar. Eine Auto-Scaling-Gruppe versucht, die gewünschte Kapazität aufrechtzuerhalten. Zu Beginn wird die Anzahl von Instances gestartet, die für die gewünschte Kapazität angegeben ist. Diese Anzahl wird beibehalten, solange der Auto-Scaling-Gruppe keine Skalierungsrichtlinien oder geplanten Aktionen zugeordnet sind.
- Minimum capacity (Minimale Kapazität): Stellt die minimale Gruppengröße dar. Wenn Skalierungsrichtlinien festgelegt sind, können sie die gewünschte Kapazität einer Gruppe nicht unter die Mindestkapazität bringen.
- Maximum capacity (Maximale Kapazität): Stellt die maximale Gruppengröße dar. Wenn Skalierungsrichtlinien festgelegt sind, können sie die gewünschte Kapazität einer Gruppe nicht über die Höchstkapazität bringen.

Mindest- und Maximalgröße gelten auch in den folgenden Szenarien:

- Manuelles Skalieren Ihrer Auto-Scaling-Gruppe durch Aktualisieren der gewünschten Kapazität

- Ausführen geplanter Aktionen, welche die gewünschte Kapazität aktualisieren. Wenn eine geplanten Aktion ohne Angabe einer neuen Mindest- und Maximalgröße für die Gruppe ausgeführt wird, gilt die aktuelle Mindest- bzw. Maximalgröße der Gruppe.

Eine Auto-Scaling-Gruppe versucht immer, die gewünschte Kapazität aufrechtzuerhalten. Wenn eine Instance unerwartet beendet wird (z. B. aufgrund einer Spot Instance-Unterbrechung, eines Fehlers bei der Zustandsprüfung oder eines menschlichen Eingriffs), startet die Gruppe automatisch eine neue Instance, um die gewünschte Kapazität aufrechtzuerhalten.

So verwalten Sie diese Einstellungen in der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Auto Scaling Auto Scaling Groups (Auto Scaling-Gruppe) aus.
3. Aktivieren Sie auf der Seite Auto Scaling Groups (Auto-Scaling-Gruppen) das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

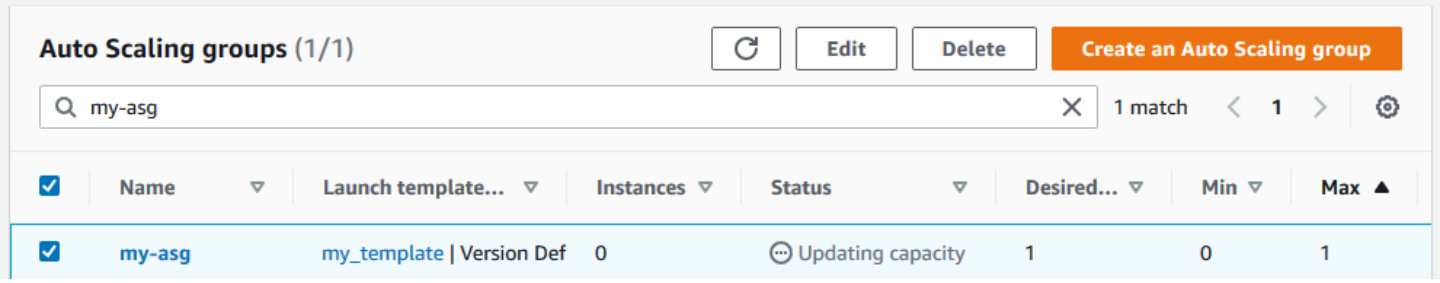
   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
4. Zeigen Sie im unteren Abschnitt auf der Registerkarte Details die aktuellen Einstellungen für die gewünschte Kapazität, die Mindest- und die Höchstkapazität an oder ändern Sie sie. Weitere Informationen finden Sie unter Ändern der gewünschten Kapazität einer vorhandenen Auto-Scaling-Gruppe.

Über dem Bereich Details finden Sie Informationen wie die aktuelle Anzahl von Instances in der Auto-Scaling-Gruppe, die gewünschte Kapazität, die Mindest- und die Höchstkapazität sowie eine Statusspalte. Wenn die Auto Scaling Scaling-Gruppe Instance-Gewichtungen verwendet, können Sie auch die Anzahl der Kapazitätseinheiten ermitteln, die zur gewünschten Kapazität beigetragen haben.

Um Spalten hinzuzufügen oder aus der Liste zu entfernen, wählen Sie das Einstellungssymbol oben auf der Seite aus. Aktivieren oder deaktivieren Sie anschließend für Auto Scaling groups attributes (Auto-Scaling-Gruppenattribute) die einzelnen Spalten und wählen Sie Confirm (Bestätigen) aus.

So überprüfen Sie die Größe der Auto-Scaling-Gruppe nach dem Vornehmen von Änderungen

In der Spalte Instances wird die Anzahl der aktuell ausgeführten Instances angezeigt. Während eine Instance gestartet oder beendet wird, zeigt die Spalte Status den Status Kapazität aktualisieren an, wie im folgenden Image dargestellt.



Warten Sie einige Minuten, und aktualisieren Sie die Ansicht, um den neuesten Status anzuzeigen. Nach Abschluss einer Skalierungsaktivität wird in der Spalte Instances ein aktualisierter Wert angezeigt.

Die Anzahl von Instances und der Status der aktuell ausgeführten Instances werden auch auf der Registerkarte Instance management (Instance-Verwaltung) unter Instances angezeigt.

Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest

CloudWatch sammelt und aggregiert Nutzungsdaten, wie CPU und Netzwerk-I/O, in Ihren Auto Scaling Scaling-Instances. Sie verwenden diese Metriken, um Skalierungsrichtlinien zu erstellen, welche die Anzahl der Instances in Ihrer Auto-Scaling-Gruppe anpassen, wenn der Wert der ausgewählten Metrik steigt und abnimmt.

Sie können angeben, wie lange eine Instance, nachdem sie den `InService` Status erreicht hat, wartet, bis sie Nutzungsdaten zu den aggregierten Metriken beiträgt. Diese angegebene Zeit wird als Standard-Instance-Warmup bezeichnet. Dadurch wird verhindert, dass die dynamische Skalierung durch Metriken für einzelne Instances beeinträchtigt wird, die noch keinen Anwendungsdatenverkehr verarbeiten und bei denen möglicherweise vorübergehend eine hohe Auslastung von Rechenressourcen auftritt.

Um die Leistung Ihrer Target-Tracking- und Step-Scaling-Richtlinien zu optimieren, empfehlen wir Ihnen dringend, das standardmäßige Instance-Warmup zu aktivieren und zu konfigurieren. Es ist standardmäßig nicht aktiviert oder konfiguriert.

Wenn Sie das standardmäßige Instance-Warmup aktivieren, sollten Sie bedenken, dass Sie verhindern können, dass Instances auf den Mindestwert für den fehlerfreien Zustand angerechnet werden, wenn Sie Instances durch eine Instanzaktualisierung ersetzen, wenn Ihre Auto Scaling Scaling-Gruppe so eingestellt ist, dass sie vor Abschluss der Initialisierung auf den Mindestwert für intakte Instanzen angerechnet werden.

Inhalt

Leistungsaspekte der Skalierung

Für die meisten Anwendungen ist es sinnvoll, eine Standardinstanz-Aufwärmzeit festzulegen, die für alle Funktionen gilt, und nicht unterschiedliche Aufwärmzeiten für verschiedene Funktionen. Wenn Sie beispielsweise keine Standardinstanzaufwärmzeit festlegen, verwendet die Instanzaktualisierungsfunktion die Kulanzzeit für die Integritätsprüfung als Standard-Aufwärmzeit. Wenn Sie über Richtlinien zur Zielverfolgung und schrittweisen Skalierung verfügen, verwenden diese den für die Standard-Abklingzeit festgelegten Wert als Standard-Aufwärmzeit. Wenn Sie über Richtlinien für vorausschauende Skalierung verfügen, haben diese keine standardmäßige Aufwärmzeit.

Während der Instances in der Warmlaufphase werden Ihre dynamischen Skalierungsrichtlinien nur dann skaliert, wenn der Metrikwert von Instances, die sich nicht in der Warmlaufphase befinden, den Schwellenwert für hohe Alarmwerte der Richtlinie (oder die Zielauslastung einer Skalierungsrichtlinie für die Zielverfolgung) überschreitet. Wenn die Nachfrage sinkt, wird die dynamische Skalierung konservativer, um die Verfügbarkeit Ihrer Anwendung zu schützen. Dadurch werden die Scale-In-Aktivitäten für die dynamische Skalierung blockiert, bis die neuen Instances vollständig warmlaufen.

Bei der Skalierung berücksichtigt Amazon EC2 Auto Scaling Instances, die sich gerade aufwärmen, als Teil der Kapazität der Gruppe, wenn entschieden wird, wie viele Instances der Gruppe hinzugefügt werden sollen. Daher führen mehrere Sicherheitslücken, für die eine ähnliche Menge an Kapazität hinzugefügt werden muss, zu einer einzigen Skalierungsaktivität. Es ist beabsichtigt, kontinuierlich zu skalieren, ohne dies übermäßig zu tun.

Wenn das standardmäßige Aufwärmen von Instances nicht aktiviert ist, variiert die Zeit, die eine Instance wartet, bevor sie Metriken an die aktuelle Kapazität sendet CloudWatch und diese auf die aktuelle Kapazität anrechnet, von Instance zu Instance. Es besteht also die Möglichkeit, dass

Ihre Skalierungsrichtlinien im Vergleich zur tatsächlich anfallenden Arbeitslast unvorhersehbar funktionieren.

Stellen Sie sich zum Beispiel eine Anwendung mit einem wiederkehrenden on-and-off Workload-Muster vor. Eine prädiktive Skalierungsrichtlinie wird verwendet, um wiederkehrende Entscheidungen darüber zu treffen, ob die Anzahl der Instances erhöht werden soll. Da es keine standardmäßige Aufwärmzeit für Richtlinien zur vorausschauenden Skalierung gibt, beginnen die Instances sofort, zu den aggregierten Metriken beizutragen. Wenn diese Instances beim Start eine höhere Ressourcennutzung haben, kann das Hinzufügen von Instances zu einem Anstieg der aggregierten Metriken führen. Abhängig davon, wie lange es dauert, bis sich die Nutzung stabilisiert hat, kann sich dies auf alle dynamischen Skalierungsrichtlinien auswirken, die diese Metriken verwenden. Wird der hohe Alarmschwellenwert einer dynamischen Skalierungsrichtlinie überschritten, nimmt die Größe der Gruppe wieder zu. Während sich die neuen Instances noch in der Vorbereitung befinden, werden Aktivitäten zum Abskalieren gesperrt.

Wählen Sie die Standard-Aufwärmzeit der Instanz

Der Schlüssel zur Einstellung der standardmäßigen Instance-Vorbereitung besteht darin, zu bestimmen, wie lange Ihre Instances benötigen, bis die Initialisierung abgeschlossen ist und wie lange der Ressourcenverbrauch benötigt, um sich zu stabilisieren, nachdem sie den `InService`-Status erreicht haben. Achten Sie bei der Wahl der Instance-Aufwärmzeit auf ein optimales Gleichgewicht zwischen der Erfassung von Nutzungsdaten für legitimen Datenverkehr und der Minimierung der Datenerfassung im Zusammenhang mit temporären Nutzungsspitzen beim Start.

Angenommen, Sie haben eine Auto-Scaling-Gruppe mit einem Elastic-Load-Balancing-Load-Balancer verbunden. Wenn der Start neuer Instances abgeschlossen ist, werden sie beim Load Balancer registriert, bevor sie in den Status `InService` wechseln. Nachdem die Instances den Status `InService` erreicht haben, kann der Ressourcenverbrauch immer noch vorübergehende Spitzen erleben und braucht Zeit, um sich zu stabilisieren. Beispielsweise dauert die Stabilisierung des Ressourcenverbrauchs für einen Anwendungsserver, der große Assets herunterladen und zwischenspeichern muss, länger als bei einem leichtgewichtigen Webserver, der keine großen Assets herunterzuladen hat. Die Instance-Vorbereitung bietet die Zeitverzögerung, die für die Stabilisierung des Ressourcenverbrauchs erforderlich ist.

> **Important**
>
> Wenn Sie sich nicht sicher sind, wie viel Zeit Sie für die Aufwärmzeit benötigen, können Sie mit 300 Sekunden beginnen. Verringern oder erhöhen Sie sie dann schrittweise, bis Sie die

beste Skalierungsleistung für Ihre Anwendung erhalten. Möglicherweise müssen Sie dies einige Male tun, um es richtig zu machen. Wenn Sie Skalierungsrichtlinien haben, die über eine eigene Aufwärmzeit (`EstimatedInstanceWarmup`) verfügen, können Sie alternativ diesen Wert für den Start verwenden. Weitere Informationen finden Sie unter Suchen Sie nach Skalierungsrichtlinien mit einer zuvor festgelegten Aufwärmzeit für Instanzen.

Erwägen Sie, Lebenszyklus-Hooks für Anwendungsfälle zu verwenden, in denen Konfigurationsaufgaben oder Skripte beim Start ausgeführt werden sollen. Lebenszyklus-Hooks können verzögern, dass neue Instances in Betrieb genommen werden, bis sie die Initialisierung abgeschlossen haben. Sie sind besonders nützlich, wenn Sie Bootstrapping-Skripte haben, die einige Zeit in Anspruch nehmen. Wenn Sie einen Lebenszyklus-Hook hinzufügen, können Sie den Wert der standardmäßigen Instance-Vorbereitung reduzieren. Weitere Informationen über das Verwenden von Lebenszyklus-Hooks finden Sie unter Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling.

Aktivieren Sie das Standard-Instance-Warmup für eine Gruppe

Sie können die standardmäßige Instance-Vorbereitung aktivieren, wenn Sie eine Auto-Scaling-Gruppe erstellen. Sie können sie auch für vorhandene Gruppen aktivieren.

Wenn Sie die Standardfunktion zum Aufwärmen von Instanzen aktivieren, müssen Sie für die folgenden Funktionen keine Werte mehr für Aufwärmparameter angeben:

- Instance-Aktualisierung
- Zielüberwachung der Skalierung
- Schrittweise Skalierung

Console

So aktivieren Sie die standardmäßige Instance-Vorbereitung für eine Gruppe (Konsole)

Wenn Sie die Auto-Scaling-Gruppe erstellen, wählen Sie auf der Seite Configure advanced options (Erweiterte Optionen konfigurieren) unter Additional Settings (Zusätzliche Einstellungen) die Option Enable default instance warmup (Standardmäßige Instance-Vorbereitung aktivieren). Wählen Sie die Aufwärmzeit, die Sie für Ihre Anwendung benötigen.

AWS CLI

So aktivieren Sie die standardmäßige Instance-Vorbereitung für eine neue Gruppe (AWS CLI)

Um die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe zu aktivieren, fügen Sie die Option `--default-instance-warmup` hinzu und geben Sie einen Wert in Sekunden von 0 bis 3 600 an. Nachdem dies aktiviert wurde, wird ein Wert von `-1` diese Einstellung ausschalten.

Der folgende create-auto-scaling-group-Befehl erstellt eine Auto-Scaling-Gruppe mit dem Namen *my-asg* und aktiviert die standardmäßige Instance-Vorbereitung mit einem Wert von *120* Sekunden.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --
default-instance-warmup 120 ...
```

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

Console

So aktivieren Sie die standardmäßige Instance-Vorbereitung für eine vorhandene Gruppe (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie in der Navigationsleiste oben die AWS-Region aus, in der Sie Ihre Auto-Scaling-Gruppe erstellt haben.
3. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
4. Wählen Sie auf der Registerkarte Details die Option Erweiterte Konfigurationen, Bearbeiten.
5. Wählen Sie unter Default instance warmup die Aufwärmzeit aus, die Sie für Ihre Anwendung benötigen.
6. Wählen Sie Aktualisieren.

AWS CLI

So aktivieren Sie die standardmäßige Instance-Vorbereitung für eine vorhandene Gruppe (AWS CLI)

Das Folgende Beispiel verwende den Befehl update-auto-scaling-group, um die standardmäßige Instance-Vorbereitung mit einem Wert von *120* Sekunden für eine vorhandene Auto-Scaling-Gruppe mit dem Namen *my-asg* zu aktivieren.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --
default-instance-warmup 120
```

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

Überprüfen Sie die standardmäßige Instance-Vorbereitung für eine Gruppe

So überprüfen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe (AWS CLI)

Verwenden Sie den folgenden describe-auto-scaling-groups-Befehl. Ersetzen Sie *my-asg* durch den Namen Ihrer Auto-Scaling-Gruppe.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn",
            ...
            "DefaultInstanceWarmup": 120
        }
    ]
```

```
}
```

Suchen Sie nach Skalierungsrichtlinien mit einer zuvor festgelegten Aufwärmzeit für Instanzen

Um festzustellen, ob Sie Richtlinien haben, für die eine eigene Aufwärmzeit gilt`EstimatedInstanceWarmup`, führen Sie den folgenden Befehl describe-policies mit dem aus. AWS CLI Ersetzen Sie *`my-asg`* durch den Namen Ihrer Auto-Scaling-Gruppe.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg
  --query 'ScalingPolicies[?EstimatedInstanceWarmup!=`null`]'
```

Es folgt eine Beispielausgabe.

```
[
  {
    "AutoScalingGroupName":"my-asg",
    "PolicyName":"cpu50-target-tracking-scaling-policy",
    "PolicyARN":"arn",
    "PolicyType":"TargetTrackingScaling",
    "StepAdjustments":[],
    "EstimatedInstanceWarmup":120,
    "Alarms":[{
        "AlarmARN": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:TargetTracking-my-
asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",
        "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
      },
      {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2",
            "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
    }],
    "TargetTrackingConfiguration":{
      "PredefinedMetricSpecification":{
        "PredefinedMetricType":"ASGAverageCPUUtilization"
      },
      "TargetValue":50.0,
      "DisableScaleIn":false
```

```
    },
    "Enabled":true
  },

    ... additional policies ...

]
```

Löschen Sie die zuvor festgelegte Instance-Vorbereitung für eine Skalierungsrichtlinie

Nachdem Sie das standardmäßige Aufwärmen der Instanz aktiviert haben, aktualisieren Sie alle Skalierungsrichtlinien, für die noch eine eigene Aufwärmzeit gilt, um den zuvor festgelegten Wert zu löschen. Andernfalls wird die standardmäßige Instance-Vorbereitung überschrieben.

Sie können Skalierungsrichtlinien mithilfe der Konsole oder AWS mithilfe von AWS CLI SDKs aktualisieren. In diesem Abschnitt werden die Schritte für die Konsole behandelt. Wenn Sie die AWS SDKs AWS CLI oder verwenden, stellen Sie sicher, dass Sie die bestehende Richtlinienkonfiguration beibehalten, aber die `EstimatedInstanceWarmup` Eigenschaft entfernen. Wenn Sie eine bestehende Skalierungsrichtlinie aktualisieren, wird die Richtlinie durch die Richtlinie ersetzt, die Sie angeben, wenn Sie Policy programmgesteuert aufrufen. PutScaling Die ursprünglichen Werte werden nicht beibehalten.

Um die zuvor festgelegte Instance-Vorbereitung für eine Skalierungsrichtlinie (Konsole) zu löschen

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Automatische Skalierung unter Dynamische Skalierungsrichtlinien die gewünschte Richtlinie aus und klicken Sie dann auf Aktionen, Bearbeiten.
4. Löschen Sie zum Beispiel Instance-Warmup den Instanz-Aufwärmwert, um stattdessen den Standardwert für das Aufwärmen der Instanz zu verwenden.
5. Wählen Sie Aktualisieren.

Manuelle Skalierung für Amazon EC2 Auto Scaling

Sie können die Anzahl der EC2-Instances in Ihrer Auto Scaling Scaling-Gruppe jederzeit manuell anpassen. Dieser Vorgang der manuellen Änderung der Anzahl der Instanzen wird als manuelle Skalierung bezeichnet. Die manuelle Skalierung ist eine Alternative zur auto Skalierung, insbesondere wenn Sie einmalige Kapazitätsänderungen vornehmen möchten.

Nachdem Sie Ihre Gruppe manuell skaliert haben, nimmt Amazon EC2 Auto Scaling die normalen Auto Scaling-Aktivitäten auf der Grundlage der von Ihnen definierten Skalierungsrichtlinien und geplanten Aktionen wieder auf. Bei Gruppen, bei denen das standardmäßige Aufwärmen von Instanzen aktiviert ist, durchlaufen alle neuen Instances eine Aufwärmphase, bevor sie zu den Metriken beitragen, die für die auto Skalierung verwendet werden. Diese Aufwärmphase hilft dabei, die Gruppe auf der neuen Kapazität zu stabilisieren. Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

Manchmal möchten Sie möglicherweise Skalierungsrichtlinien und geplante Aktionen vorübergehend deaktivieren, bevor Sie eine Gruppe manuell skalieren. Dadurch wird verhindert, dass Konflikte zwischen manuellen Skalierungsaktionen und automatisierten Skalierungsaktivitäten entstehen. Weitere Informationen finden Sie unter Skalierungsaktivitäten ausschalten.

Inhalt

Ändern der gewünschten Kapazität einer vorhandenen Auto-Scaling-Gruppe

Wenn Sie die gewünschte Kapazität Ihrer Auto Scaling-Gruppe ändern, verwaltet Amazon EC2 Auto Scaling den Prozess des Startens und Beendens von Instances, um die neue gewünschte Größe zu erreichen.

Console

Ändern der Größe einer Auto-Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.

2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Am unteren Rand der Seite wird ein geteilter Bereich angezeigt.

3. Wählen Sie auf der Registerkarte Details die Option Gruppendetails, Bearbeiten.
4. Erhöhen oder verringern Sie für Gewünschte Kapazität die gewünschte Kapazität. Um beispielsweise die Größe der Gruppe um eins zu erhöhen, geben Sie ein, wenn der aktuelle Wert lautet 12.

   Wenn Ihr neuer Wert für die gewünschte Kapazität größer als die gewünschte Mindestkapazität und die gewünschte Höchstkapazität ist, wird die gewünschte Höchstkapazität automatisch auf den neuen Wert für die gewünschte Kapazität erhöht.
5. Wählen Sie Aktualisieren aus, wenn Sie fertig sind.

Stellen Sie sicher, dass die von Ihnen angegebene Gruppengröße dazu geführt hat, dass dieselbe Anzahl von Instances gestartet wurde. Wenn Sie beispielsweise die Gruppengröße um eins erhöht haben, stellen Sie sicher, dass Ihre Auto Scaling Scaling-Gruppe eine zusätzliche Instance gestartet hat.

Überprüfen Sie wie folgt, ob sich die Größe der Auto-Scaling-Gruppe geändert hat:

1. Auf der Registerkarte Aktivität können Sie im Aktivitätsverlauf den Fortschritt der Aktivitäten anzeigen, die der Auto Scaling Scaling-Gruppe zugeordnet sind. In der Status-Spalte wird der aktuelle Status Ihrer Instance angezeigt. Während die Instance gestartet wird, zeigt die Statusspalte `Not yet in service` an. Nach dem Start der Instance ändert sich der Status in `Successful`. Sie können auch das Aktualisierungssymbol verwenden, um den aktuellen Status Ihrer Instance zu sehen. Weitere Informationen finden Sie unter Eine Skalierung für eine Auto-Scaling-Gruppe überprüfen.
2. Auf der Registerkarte Instanzverwaltung unter Instances können Sie den Status der Instance einsehen. Es dauert einige Zeit, bis die Instance startet.

   - In der Spalte Lifecycle (Lebenszyklus) wird Ihnen der Zustand Ihrer Instance angezeigt. Die Instance befindet sich zunächst im Status `Pending`. Wenn eine Instance für den Empfang von Datenverkehr bereit ist, lautet der Status `InService`.
   - In der Spalte Health Status wird das Ergebnis der Amazon EC2 Auto Scaling Scaling-Zustandsprüfungen für Ihre Instance angezeigt.

AWS CLI

Im folgenden Beispiel wird davon ausgegangen, dass Sie eine Auto-Scaling-Gruppe mit einer minimalen Größe von 1 und einer maximalen Größe von 5 erstellt haben. Also verfügt die Gruppe derzeit über eine laufenden Instance.

Ändern der Größe einer Auto-Scaling-Gruppe

Verwenden Sie den Befehl set-desired-capacity, um die Größe Ihrer Auto-Scaling-Gruppe wie im folgenden Beispiel zu ändern:

```
aws autoscaling set-desired-capacity --auto-scaling-group-name my-asg \
  --desired-capacity 2
```

Wenn Sie die standardmäßige Ruhephase für Ihre Auto-Scaling-Gruppe berücksichtigen möchten, müssen Sie die Option `--honor-cooldown` wie im folgenden Beispiel dargestellt angeben. Weitere Informationen finden Sie unter Skalierungsruhephasen für Amazon EC2 Auto Scaling.

```
aws autoscaling set-desired-capacity --auto-scaling-group-name my-asg \
  --desired-capacity 2 --honor-cooldown
```

So überprüfen Sie die Größe Ihrer Auto-Scaling-Gruppe

Verwenden Sie den Befehl describe-auto-scaling-groups, um wie im folgenden Beispiel zu bestätigen, dass sich die Größe der Auto-Scaling-Gruppe geändert hat:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Im Folgenden finden Sie eine Beispielausgabe, die Details zur Gruppe und den gestarteten Instances enthält.

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
```

```
            "MinSize": 1,
            "MaxSize": 5,
            "DesiredCapacity": 2,
            "DefaultCooldown": 300,
            "AvailabilityZones": [
                "us-west-2a"
            ],
            "LoadBalancerNames": [],
            "TargetGroupARNs": [],
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": 300,
            "Instances": [
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-05b4f7d5be44822a6",
                    "InstanceType": "t3.micro",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "Pending"
                },
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-0c20ac468fa3049e8",
                    "InstanceType": "t3.micro",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService"
                }
            ],
            "CreatedTime": "2019-03-18T23:30:42.611Z",
            "SuspendedProcesses": [],
            "VPCZoneIdentifier": "subnet-c87f2be0",
            "EnabledMetrics": [],
            "Tags": [],
```

```
            "TerminationPolicies": [
                "Default"
            ],
            "NewInstancesProtectedFromScaleIn": false,
            "ServiceLinkedRoleARN": "arn",
            "TrafficSources": []
        }
    ]
}
```

`DesiredCapacity` zeigt den neuen Wert. Ihre Auto-Scaling-Gruppe hat eine zusätzliche Instance gestartet.

Beenden einer Instance in Ihrer Auto-Scaling-Gruppe (AWS CLI)

Es kann vorkommen, dass Sie Ihre Auto-Scaling-Gruppe manuell abskalieren möchten, aber eine bestimmte Instance beenden möchten. Sie können Ihre Auto-Scaling-Gruppe manuell skalieren, indem Sie den terminate-instance-in-auto-scaling-group-Befehl verwenden und die ID der Instance, die Sie beenden möchten, sowie die Option `--should-decrement-desired-capacity` angeben, wie im folgenden Beispiel gezeigt.

```
aws autoscaling terminate-instance-in-auto-scaling-group \
  --instance-id i-026e4c9f62c3e448c --should-decrement-desired-capacity
```

Im Folgenden finden Sie eine Beispielausgabe, die Details zur Skalierungsaktivität enthält.

```
{
    "Activities": [
        {
            "ActivityId": "b8d62b03-10d8-9df4-7377-e464ab6bd0cb",
            "AutoScalingGroupName": "my-asg",
            "Description": "Terminating EC2 instance: i-026e4c9f62c3e448c",
            "Cause": "At 2023-09-23T06:39:59Z instance i-026e4c9f62c3e448c was taken
 out of service in response to a user request, shrinking the capacity from 1 to 0.",
            "StartTime": "2023-09-23T06:39:59.015000+00:00",
            "StatusCode": "InProgress",
            "Progress": 0,
            "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":\"us-
west-2c\"}"
        }
    ]
```

```
}
```

Diese Option ist in der Konsole nicht verfügbar. Sie können jedoch die Instance-Seite der Amazon EC2 EC2-Konsole verwenden, um eine Instance in Ihrer Auto Scaling Scaling-Gruppe zu beenden. Wenn Sie dies tun, erkennt Amazon EC2 Auto Scaling, dass die Instance nicht mehr läuft, und ersetzt sie automatisch im Rahmen der Zustandsprüfung. Nach dem Beenden der Instance dauert es ein oder zwei Minuten, bis eine neue Instance gestartet wird. Informationen zum Beenden einer Instance finden Sie unter Terminate an Instance im Amazon EC2 EC2-Benutzerhandbuch.

Wenn Sie Instances in Ihrer Gruppe beenden und dies zu einer ungleichmäßigen Verteilung auf die Availability Zones führt, gleicht Amazon EC2 Auto Scaling die Gruppe neu aus, um eine gleichmäßige Verteilung wiederherzustellen, sofern Sie den Vorgang nicht unterbrechen. `AZRebalance` Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling Scaling-Prozesse aussetzen und fortsetzen.

Geplante Skalierung für Amazon EC2 Auto Scaling

Mit der geplanten Skalierung können Sie eine automatische Skalierung für Ihre Anwendung einrichten, die auf vorhersehbaren Laständerungen basiert. Sie erstellen geplante Aktionen, die die gewünschte Kapazität Ihrer Gruppe zu bestimmten Zeiten erhöhen oder verringern.

Sie erleben beispielsweise ein regelmäßiges wöchentliches Verkehrsmuster, bei dem die Auslastung unter der Woche zunimmt und gegen Ende der Woche abnimmt. Sie können in Amazon EC2 Auto Scaling einen Skalierungsplan konfigurieren, der diesem Muster entspricht:

- Am Mittwochmorgen erhöht eine geplante Aktion die Kapazität, indem die zuvor festgelegte gewünschte Kapazität der Auto Scaling Scaling-Gruppe erhöht wird.
- Am Freitagabend verringert eine weitere geplante Aktion die Kapazität, indem die zuvor festgelegte gewünschte Kapazität der Auto Scaling Scaling-Gruppe 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, stellt aber zu anderen Zeiten nicht zu viel Kapazität bereit.

Sie können geplante Skalierung und Skalierungsrichtlinien zusammen verwenden, um die Vorteile beider Skalierungsansätze 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.

Inhalt

So funktioniert die geplante Skalierung

Um die geplante Skalierung zu verwenden, erstellen Sie geplante Aktionen, die Amazon EC2 Auto Scaling anweisen, Skalierungsaktivitäten zu bestimmten Zeiten durchzuführen. Wenn Sie eine geplante Aktion erstellen, geben Sie die Auto Scaling Scaling-Gruppe an, wann die Skalierungsaktivität stattfinden soll, die neue gewünschte Kapazität und optional eine neue Mindestkapazität und eine neue Höchstkapazität. Sie können geplante Aktionen erstellen, die nur einmal skalieren oder wiederholt geplant ausgeführt werden.

Zum angegebenen Zeitpunkt skaliert Amazon EC2 Auto Scaling auf der Grundlage der neuen Kapazitätswerte, indem die aktuelle Kapazität mit der angegebenen gewünschten Kapazität verglichen wird.

- Wenn die aktuelle Kapazität unter der angegebenen gewünschten Kapazität liegt, skaliert Amazon EC2 Auto Scaling die angegebene gewünschte Kapazität oder fügt Instances hinzu.
- Wenn die aktuelle Kapazität die angegebene gewünschte Kapazität übersteigt, skaliert Amazon EC2 Auto Scaling Instances auf die angegebene gewünschte Kapazität oder entfernt sie.

Eine geplante Aktion legt die gewünschte, minimale und maximale Kapazität der Gruppe zum angegebenen Datum und zur angegebenen Uhrzeit fest. Sie können eine geplante Aktion jeweils nur

für eine dieser Kapazitäten erstellen, z. B. für die gewünschte Kapazität. In einigen Fällen müssen Sie jedoch die Mindest- und Höchstkapazität angeben, um sicherzustellen, dass die gewünschte Kapazität, die Sie in der Aktion angegeben haben, diese Grenzwerte nicht überschreitet.

Wiederkehrende Zeitpläne

Um mit dem AWS CLI oder einem SDK einen wiederkehrenden Zeitplan zu erstellen, geben Sie einen Cron-Ausdruck und eine Zeitzone an, um zu beschreiben, wann die geplante Aktion wiederholt werden soll. Sie können optional ein Datum und eine Uhrzeit für die Startzeit, die Endzeit oder beides angeben.

Um mit dem einen wiederkehrenden Zeitplan zu erstellen AWS Management Console, geben Sie das Wiederholungsmuster, die Zeitzone, die Startzeit und optional die Endzeit Ihrer geplanten Aktion an. Alle Wiederholungsmusteroptionen basieren auf Cron-Ausdrücken. Alternativ können Sie Ihren eigenen benutzerdefinierten Cron-Ausdruck schreiben.

Der unterstützte Cron-Ausdruck besteht aus fünf Feldern, getrennt durch Leerzeichen: [Minute] [Stunde] [Tag_des_Monats] [Monat_des_Jahres] [Wochentag]. Beispielsweise konfiguriert der Cron-Ausdruck `30 6 * * 2` eine geplante Aktion, die jeden Dienstag um 6:30 Uhr wiederholt wird. Das Sternchen wird als Platzhalter verwendet, um alle Werte für ein Feld abzugleichen. Weitere Beispiele für Cron-Ausdrücke finden Sie unter https://crontab.guru/examples.html. Informationen zum Schreiben eigener Cron-Ausdrücke in diesem Format finden Sie unter Crontab.

Wählen Sie Ihre Start- und Endzeiten sorgfältig aus. Beachten Sie Folgendes:

- Wird ein Startzeitpunkt angegeben, führt Amazon EC2 Auto Scaling die Aktion zu diesem Zeitpunkt und dann auf Grundlage der angegebenen Wiederholung aus.
- Wenn Sie eine Endzeit angeben, wird die Aktion nach dieser Zeit nicht mehr wiederholt. Eine geplante Aktion bleibt nicht in Ihrem Konto, nachdem sie ihre Endzeit erreicht hat.
- Die Start- und Endzeit müssen in UTC festgelegt werden, wenn Sie das AWS CLI oder ein SDK verwenden.

Zeitzone

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, die Sommerzeit befolgt, wird die Aktion automatisch für Sommerzeit angepasst.

Die gültigen Werte sind die kanonischen Namen für Zeitzonen aus der Zeitzonendatenbank der Internet Assigned Numbers Authority (IANA). Die östliche Zeit der USA wird beispielsweise kanonisch als bezeichnet. `America/New_York` Weitere Informationen finden Sie unter https://www.iana.org/time-zones.

Ortsbezogene Zeitzonen passen sich z. B. `America/New_York` automatisch an die Sommerzeit an. Eine UTC-basierte Zeitzone wie `Etc/UTC` ist eine absolute Zeit und wird nicht der Sommerzeit angepasst.

Sie haben beispielsweise einen wiederkehrenden Zeitplan, dessen Zeitzone `America/New_York` ist. Die erste Skalierungsaktion findet in der `America/New_York`-Zeitzone vor dem Start der Sommerzeit statt. Die nächste Skalierungsaktion findet in der `America/New_York`-Zeitzone nach dem Start der Sommerzeit statt. Die erste Aktion beginnt um 8:00 Uhr UTC-5 Ortszeit, während das zweite Mal um 8:00 Uhr UTC-4 in Ortszeit beginnt.

Wenn Sie eine geplante Aktion mit der erstellen AWS Management Console und eine Zeitzone angeben, in der die Sommerzeit eingehalten wird, passen sich sowohl der wiederkehrende Zeitplan als auch die Start- und Endzeiten automatisch an die Sommerzeit an.

Überlegungen

Beachten Sie bei der Erstellung einer geplanten Aktion Folgendes:

- Die Reihenfolge der Ausführung geplanter Aktionen wird innerhalb derselben Gruppe, aber nicht gruppenübergreifend, garantiert.
- Eine geplante Aktion wird in der Regel innerhalb von Sekunden ausgeführt. Allerdings kann die Aktion um bis zu zwei Minuten nach der geplanten Startzeit verzögert sein. Da geplante Aktionen innerhalb einer Auto-Scaling-Gruppe in der festgelegten Reihenfolge ausgeführt werden, benötigen Aktionen mit nahe beieinanderliegenden geplanten Startzeiten in der Ausführung mehr Zeit.
- Sie können die geplante Skalierung für eine Auto-Scaling-Gruppe vorübergehend deaktivieren, indem Sie das `ScheduledActions`-Verfahren abbrechen. 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 Amazon EC2 Auto Scaling Scaling-Prozesse aussetzen und fortsetzen.
- Nachdem Sie eine geplante Aktion erstellt haben, können Sie alle Einstellungen mit Ausnahme des Namens aktualisieren.

Eine geplante Aktion erstellen

Verwenden Sie eine der folgenden Methoden, um eine geplante Aktion für Ihre Auto Scaling Scaling-Gruppe zu erstellen:

Console

Erstellen Sie eine geplante Aktion wie folgt:

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Automatic scaling (Automatische Skalierung) unter Scheduled actions (Geplante Aktionen) die Option Geplante Aktion erstellen (Create scheduled action) aus.
4. Geben Sie einen Namen für die geplante Aktion ein.
5. Für Gewünschte Kapazität, Min., Max. wählen Sie die neue gewünschte Größe der Gruppe und die neuen minimalen und maximalen Größenlimits aus. Die gewünschte Kapazität darf die Mindestgröße der Gruppe nicht unterschreiten und die maximale Gruppengröße nicht übersteigen.
6. Wählen Sie für Recurrence (Wiederholung) eine der verfügbaren Optionen aus.
   - Wenn Sie nach einem wiederkehrenden Zeitplan skalieren möchten, wählen Sie aus, wie oft Amazon EC2 Auto Scaling die geplante Aktion ausführen soll.
     - Wenn Sie eine Option auswählen, die mit Every (Alle) beginnt, wird der Cron-Ausdruck für Sie erstellt.
     - Wenn Sie Cron auswählen, geben Sie einen Cron-Ausdruck ein, der angibt, wann die Aktion ausgeführt werden soll.
   - Wenn Sie nur einmal skalieren möchten, wählen Sie Einmalig aus.
7. Wählen Sie für Zeitzone eine Zeitzone aus. Der Standardwert ist `Etc/UTC`.

   Alle aufgelisteten Zeitzonen stammen aus der IANA-Zeitzonendatenbank. Weitere Informationen finden Sie unter https://en.wikipedia.org/wiki/List_of_tz_database_time_zones.
8. Definieren Sie ein Datum und eine Uhrzeit für Bestimmte Startzeit.

- Wenn Sie einen wiederkehrenden Zeitplan gewählt haben, legt die Startzeit fest, wann die erste geplante Aktion in der wiederkehrenden Reihe ausgeführt wird.
- Wenn Sie Einmalig als Wiederholung ausgewählt haben, definiert die Startzeit das Datum und die Uhrzeit für die Ausführung der geplanten Aktion.

9. (Optional) Bei wiederkehrenden Zeitplänen können Sie eine Endzeit angeben, indem Sie Festlegen der Endzeit und dann ein Datum und eine Uhrzeit für Beenden bis auswählen.
10. Wählen Sie Erstellen. Die Konsole zeigt die geplanten Aktionen der Auto-Scaling-Gruppe an.

AWS CLI

Um eine geplante Aktion zu erstellen, können Sie einen der folgenden Beispielbefehle verwenden. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Beispiel: Einmalige Skalierung

Verwenden Sie den folgenden Befehl put-scheduled-update-group-action mit den Optionen und. `--start-time "YYYY-MM-DDThh:mm:ssZ"` `--desired-capacity`

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-one-time-action \
  --auto-scaling-group-name my-asg --start-time "2021-03-31T08:00:00Z" --desired-capacity 3
```

Beispiel: Um die Skalierung nach einem wiederkehrenden Zeitplan zu planen

Verwenden Sie den folgenden Befehl put-scheduled-update-group-action mit den Optionen und. `--recurrence "cron expression"` `--desired-capacity`

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-recurring-action \
  --auto-scaling-group-name my-asg --recurrence "0 9 * * *" --desired-capacity 3
```

Standardmäßig führt Amazon EC2 Auto Scaling den angegebenen Wiederholungsplan auf der Grundlage der UTC-Zeitzone aus. Um eine andere Zeitzone anzugeben, geben Sie die `--time-zone` Option und den Namen der IANA-Zeitzone an, wie im folgenden Beispiel.

```
--time-zone "America/New_York"
```

Weitere Informationen finden Sie unter https://en.wikipedia.org/wiki/List_of_tz_database_time_zones.

Details zu geplanten Aktionen anzeigen

Verwenden Sie eine der folgenden Methoden, um Details zu bevorstehenden geplanten Aktionen für Ihre Auto Scaling Scaling-Gruppe anzuzeigen:

Console

Um Details zu geplanten Aktionen anzuzeigen

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie Ihre Auto-Scaling-Gruppe aus.
3. Auf der Registerkarte Automatische Skalierung können Sie sich im Abschnitt Geplante Aktionen über bevorstehende geplante Aktionen informieren.

Beachten Sie, dass die Konsole die Werte für Startzeit und Endzeit in Ihrer Ortszeit anzeigt, wobei der UTC-Offset zum angegebenen Datum und zur angegebenen Uhrzeit gültig ist. Der UTC-Offset ist die Differenz (in Stunden und Minuten) von Ortszeit zu UTC. Der Wert für Zeitzone zeigt Ihre angeforderte Zeitzone an, z. B. `America/New_York`.

AWS CLI

Verwenden Sie den folgenden Befehl describe-scheduled-actions.

```
aws autoscaling describe-scheduled-actions --auto-scaling-group-name my-asg
```

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

```
{
  "ScheduledUpdateGroupActions": [
    {
      "AutoScalingGroupName": "my-asg",
      "ScheduledActionName": "my-recurring-action",
      "Recurrence": "30 0 1 1,6,12 *",
      "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
```

```
      "StartTime": "2020-12-01T00:30:00Z",
      "Time": "2020-12-01T00:30:00Z",
      "MinSize": 1,
      "MaxSize": 6,
      "DesiredCapacity": 4
    }
  ]
}
```

Überprüfen von Skalierungsaktivitäten

Informationen zur Überprüfung der Skalierungsaktivitäten im Zusammenhang mit der geplanten Skalierung finden Sie unter Eine Skalierung für eine Auto-Scaling-Gruppe überprüfen.

Löschen einer geplanten Aktion

Verwenden Sie eine der folgenden Methoden, um eine geplante Aktion zu löschen:

Console

Löschen einer geplanten Aktion

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie Ihre Auto-Scaling-Gruppe aus.
3. Wählen Sie auf der Registerkarte Automatic scaling (Automatische Skalierung) unter Scheduled actions (Geplante Aktionen) eine geplante Aktion aus.
4. Wählen Sie Aktionen, Löschen aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Ja, löschen.

AWS CLI

Verwenden Sie den folgenden Befehl delete-scheduled-action.

```
aws autoscaling delete-scheduled-action --auto-scaling-group-name my-asg \
  --scheduled-action-name my-recurring-action
```

Einschränkungen

- Die Namen der geplanten Aktionen müssen pro Auto-Scaling-Gruppe eindeutig sein.
- Eine geplante Aktion muss über einen eindeutigen Zeitwert verfügen. Wenn Sie versuchen, eine Aktivität zu einem Zeitpunkt zu planen, zu dem bereits eine andere Skalierung geplant ist, wird der Anruf abgelehnt und gibt einen Fehler zurück, der angibt, dass eine geplante Aktion mit dieser geplanten Startzeit bereits vorhanden ist.
- Sie können maximal 125 geplante Aktionen pro Auto-Scaling-Gruppe erstellen.

Dynamische Skalierung für Amazon EC2 Auto Scaling

Bei der dynamischen Skalierung wird die Kapazität Ihrer Auto-Scaling-Gruppe skaliert, wenn sich der Datenverkehr ändert.

Amazon EC2 Auto Scaling unterstützt die folgenden Typen von dynamischen Skalierungsrichtlinien:

- Skalierung der Zielverfolgung — Erhöhen und verringern Sie die aktuelle Kapazität der Gruppe auf der Grundlage einer CloudWatch Amazon-Metrik und eines Zielwerts. Das funktioniert ähnlich wie bei einem Thermostat, der die Temperatur in Ihrem Zuhause aufrechterhält: Sie wählen eine Temperatur und der Thermostat erledigt den Rest.
- Step scaling (Schrittweise Skalierung): Erhöht und verringert die aktuelle Kapazität der Gruppe auf der Grundlage einer Reihe von Skalierungsanpassungen, die als Schrittanpassungen bezeichnet werden und je nach Ausmaß der Alarmüberschreitung variieren.
- Simple scaling (Einfache Skalierung): Erhöht und verringert die aktuelle Kapazität der Gruppe auf der Grundlage einer einzelnen Skalierungsanpassung und mit einer Ruhephase zwischen den einzelnen Skalierungsaktivitäten.

Wir empfehlen dringend, dass Sie Skalierungsrichtlinien für die Zielverfolgung verwenden und eine Metrik wählen, die sich umgekehrt proportional zu einer Änderung der Kapazität Ihrer Auto Scaling Scaling-Gruppe ändert. Wenn Sie also die Größe Ihrer Auto Scaling Scaling-Gruppe verdoppeln, sinkt die Metrik um 50 Prozent. Auf diese Weise können die Metrikdaten genau proportionale Skalierungsereignisse auslösen. Enthalten sind Metriken wie die durchschnittliche CPU-Auslastung oder die durchschnittliche Anzahl von Anfragen pro Ziel.

Mit Target Tracking skaliert Ihre Auto Scaling-Gruppe direkt proportional zur tatsächlichen Auslastung Ihrer Anwendung. Das bedeutet, dass eine Zielverfolgungsrichtlinie nicht nur den

unmittelbaren Kapazitätsbedarf deckt, indem sie auf Laständerungen reagiert, sondern sich auch an Laständerungen anpassen kann, die im Laufe der Zeit auftreten (beispielsweise aufgrund saisonaler Schwankungen).

Richtlinien zur Zielverfolgung machen es außerdem überflüssig, CloudWatch Alarme und Skalierungsanpassungen manuell zu definieren. Amazon EC2 Auto Scaling verarbeitet dies automatisch auf der Grundlage des von Ihnen festgelegten Ziels.

Inhalt

Funktionsweise von dynamischen Skalierungsrichtlinien

Eine dynamische Skalierungsrichtlinie weist Amazon EC2 Auto Scaling an, eine bestimmte CloudWatch Metrik zu verfolgen, und sie definiert, welche Aktion zu ergreifen ist, wenn der zugehörige CloudWatch Alarm in ALARM ist. Die Metriken, die zum Auslösen des Alarmstatus verwendet werden, sind eine Aggregation von Metriken, die von allen Instances in der Auto-Scaling-Gruppe stammen. (Angenommen, Sie haben eine Auto-Scaling-Gruppe mit zwei Instances, bei denen eine Instance 60 Prozent CPU und die andere 40 Prozent CPU hat. Im Durchschnitt liegen sie bei 50 Prozent CPU.) Wenn die Richtlinie in Kraft ist, passt Amazon EC2 Auto Scaling die gewünschte Kapazität der Gruppe nach oben oder unten an, wenn die Schwelle eines Alarms überschritten wurde.

Wenn eine dynamische Skalierungsrichtlinie aufgerufen wird und die Kapazitätsberechnung eine Zahl außerhalb des minimalen und maximalen Größenbereichs der Gruppe erzeugt, stellt Amazon EC2 Auto Scaling sicher, dass die neue Kapazität niemals außerhalb der minimalen und maximalen Größengrenzen liegt. Die Kapazität wird auf zwei Arten gemessen: mit denselben Einheiten, die Sie

bei der Festlegung der gewünschten Kapazität in Form von Instances ausgewählt haben, oder mit Kapazitätseinheiten (wenn Instance-Gewichtungen angewendet werden).

- Beispiel 1: Eine Auto-Scaling-Gruppe hat eine maximale Kapazität von 3, eine aktuelle Kapazität von 2 und eine dynamische Skalierungsrichtlinie, die drei Instances hinzufügt. Beim Aufrufen dieser Skalierungsrichtlinie fügt Amazon EC2 Auto Scaling der Gruppe nur eine Instance hinzu, um zu verhindern, dass die Gruppe ihre maximale Größe überschreitet.
- Beispiel 2: Eine Auto-Scaling-Gruppe hat eine Mindestkapazität von 2, eine aktuelle Kapazität von 3 und eine dynamische Skalierungsrichtlinie, die zwei Instances entfernt. Wenn Sie diese Richtlinie aufrufen, entfernt Amazon EC2 Auto Scaling nur eine Instance aus der Gruppe, um zu verhindern, dass die Gruppe ihre Mindestgröße unterschreitet.

Wenn die gewünschte Kapazität die maximale Größengrenze erreicht, stoppt die Skalierung. Wenn der Bedarf sinkt und die aktuelle genutzte Kapazität abnimmt, kann Amazon EC2 Auto Scaling entsprechend aufwärts skalieren.

Die Ausnahme ist, wenn Sie Instance-Gewichte verwenden. In diesem Fall kann Amazon EC2 Auto Scaling über die maximale Größenbeschränkung hinaus skaliert werden, aber nur um das maximale Instance-Gewicht. Die Absicht ist, so nah wie möglich an die neue gewünschte Kapazität zu kommen, aber dennoch die für die Gruppe festgelegten Zuordnungsstrategien einzuhalten. Die Zuweisungsstrategien legen fest, welche Instance-Typen gestartet werden sollen. Die Gewichtungen legen fest, wie viele Kapazitätseinheiten jede Instance auf der Grundlage ihres Instance-Typs zur gewünschten Kapazität der Gruppe beiträgt.

- Beispiel 3: Eine Auto-Scaling-Gruppe hat eine maximale Kapazität von 12, eine aktuelle Kapazität von 10 und eine dynamische Skalierungsrichtlinie, die 5 Kapazitätseinheiten hinzufügt. Den Instance-Typen ist jeweils eine von drei Gewichtungen zugewiesen: 1, 4 oder 6. Wenn Sie die Skalierungsrichtlinie aufrufen, wählt Amazon EC2 Auto Scaling einen Instance-Typ mit einer Gewichtung von 6 basierend auf der Zuordnungsstrategie zum Start. Das Ergebnis dieses Scale-Out-Ereignisses ist eine Gruppe mit einer gewünschten Kapazität von 12 und einer aktuellen Kapazität von 16.

Mehrere dynamische Skalierungsrichtlinien

In den meisten Fällen reicht eine Skalierungsrichtlinie für die Zielnachverfolgung aus, um Ihre Auto-Scaling-Gruppe für eine automatische Auf- und Abwärtsskalierung zu konfigurieren. Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung ermöglicht es Ihnen, ein gewünschtes Ergebnis

auszuwählen und die Auto-Scaling-Gruppe nach Bedarf Instances hinzufügen und entfernen zu lassen, um dieses Ergebnis zu erreichen.

Für eine erweiterte Skalierungskonfiguration kann Ihre Auto-Scaling-Gruppe mehr als eine Skalierungsrichtlinie haben. So können Sie beispielsweise eine oder mehrere Skalierungsrichtlinien zur Ziel-Nachverfolgung, eine oder mehrere Richtlinien für schrittweise Skalierung oder beides definieren. Dies bietet eine größere Flexibilität, um mehrere Szenarien abzudecken.

Um zu sehen, wie mehrere dynamische Skalierungsrichtlinien zusammenarbeiten, betrachten Sie eine Anwendung, die eine Auto-Scaling-Gruppe und eine Amazon SQS-Warteschlange verwendet, um Anfragen an eine einzelne EC2-Instance zu senden. Zwei Richtlinien steuern, wann die Auto-Scaling-Gruppe eine horizontale Skalierung nach oben durchführt, um die optimale Leistung der Anwendung sicherzustellen. Eine davon ist eine Zielverfolgungsrichtlinie, die eine benutzerdefinierte Metrik verwendet, um Kapazität basierend auf der Anzahl der SQS-Nachrichten in der Warteschlange hinzuzufügen und zu entfernen. Die andere ist eine schrittweise Skalierungsrichtlinie, die die CloudWatch `CPUUtilization` Amazon-Metrik verwendet, um Kapazität hinzuzufügen, wenn die Instance für einen bestimmten Zeitraum eine Auslastung von 90 Prozent überschreitet.

Wenn mehrere Richtlinien gleichzeitig in Kraft sind, besteht die Möglichkeit, dass jede Richtlinie die Auto-Scaling-Gruppe anweisen könnte, sich gleichzeitig zu vergrößern (oder zu verkleinern). Es ist beispielsweise möglich, dass die `CPUUtilization` Metrik den Schwellenwert des CloudWatch Alarms erreicht und diesen überschreitet, während die benutzerdefinierte SQS-Metrik den Schwellenwert des benutzerdefinierten Metrik-Alarms ansteigt und überschreitet.

In einer solchen Situation wählt Amazon EC2 Auto Scaling die Richtlinie, welche die größte Kapazität für das Aufwärtskalieren und Abwärtskalieren bietet. Angenommen, die Richtlinie für `CPUUtilization` startet eine einzelne Instance, während die Richtlinie für die SQS-Warteschlange zwei Instances startet. Sind die Kriterien zur Skalierung nach oben für beide Richtlinien gleichzeitig erfüllt, gibt Amazon EC2 Auto Scaling der SQS-Warteschlange den Vorrang. Daher startet die Auto-Scaling-Gruppe zwei Instances.

Der Ansatz, der Richtlinie mit der größten Kapazität Vorrang einzuräumen, gilt auch dann, wenn die Richtlinien unterschiedliche Kriterien für das Herunterskalieren verwenden. Beispiel: Wenn eine Richtlinie beispielsweise drei Instances beendet, eine andere Richtlinie die Anzahl der Instances um 25 Prozent verringert und die Gruppe zum Zeitpunkt der Abwärtsskalierung über acht Instances verfügt, räumt Amazon EC2 Auto Scaling der Richtlinie Vorrang ein, welche die größte Anzahl von Instances für die Gruppe bereitstellt. Dies führt dazu, dass die Auto-Scaling-Gruppe zwei Instances beendet (25 Prozent von 8 = 2). Damit soll verhindert werden, dass Amazon EC2 Auto Scaling zu viele Instances entfernt.

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 Abwärtsskalierungsaktivität könnte die Zielverfolgungsrichtlinie die Gruppe anweisen, erneut aufwärts zu skalieren.

Skalierungsrichtlinien für die Ziel-Nachverfolgung für Amazon EC2 Auto Scaling

Eine Skalierungsrichtlinie für die Zielverfolgung skaliert automatisch die Kapazität Ihrer Auto Scaling Scaling-Gruppe auf der Grundlage eines Zielmetrikwerts. 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. Amazon EC2 Auto Scaling erstellt und verwaltet die CloudWatch Alarme, die Skalierungsereignisse auslösen, wenn die Metrik vom Ziel abweicht. Dies ähnelt beispielsweise der Art und Weise, wie ein Thermostat eine Zieltemperatur beibehält.

Ein Beispiel: Angenommen, Sie verfügen über eine Webanwendung, die derzeit in zwei Instances ausgeführt wird, und Sie möchten, dass die CPU-Auslastung der Auto-Scaling-Gruppe bei etwa 50 Prozent bleibt, wenn sich die Last der Anwendung 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 Ihre Auto Scaling Scaling-Gruppe die Kapazität oder erhöht die Kapazität, wenn die CPU 50 Prozent überschreitet, um die erhöhte Last zu bewältigen. Die Kapazität wird erhöht oder verringert, wenn die CPU unter 50 Prozent fällt, um die Kosten in Zeiten geringer Auslastung zu optimieren.

Themen

- Überlegungen
- Erstellen einer Zielverfolgungs-Skalierungsrichtlinie
- Erstellen einer Zielnachverfolgungs-Skalierungsrichtlinie für Amazon EC2 Auto Scaling mit Metrikberechnungen

Mehrere Skalierungsrichtlinien für die Zielverfolgung

Zur Optimierung der Skalierung können mehrere Skalierungsrichtlinien für die Zielverfolgung miteinander kombiniert werden. Diese müssen allerdings jeweils eine andere Metrik verwenden. Auslastung und Durchsatz können sich beispielsweise gegenseitig beeinflussen. Wenn sich eine dieser Metriken ändert, bedeutet das in der Regel, dass auch andere Metriken betroffen sind. Die Verwendung mehrerer Metriken liefert daher zusätzliche Informationen über die Last, unter der Ihre Auto Scaling Scaling-Gruppe steht. Dies kann Amazon EC2 Auto Scaling dabei helfen, fundiertere Entscheidungen zu treffen, wenn es darum geht, wie viel Kapazität zu Ihrer Gruppe hinzugefügt werden soll.

Die Absicht von Amazon EC2 Auto Scaling besteht darin, der Verfügbarkeit immer Priorität einzuräumen. Es wird die Auto Scaling-Gruppe skalieren, wenn eine der Zielverfolgungsrichtlinien für die Skalierung bereit ist. Die Skalierung erfolgt nur, wenn alle Richtlinien zur Zielverfolgung (bei aktivierter Skalierung) für die Skalierung bereit sind.

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 folgenden Liste vordefinierter Metriken aus.

- `ASGAverageCPUUtilization` – Durchschnittliche CPU-Nutzung der Auto-Scaling-Gruppe.
- `ASGAverageNetworkIn` – Die durchschnittliche Anzahl empfangener Bytes von einer einzigen Instance für alle Netzwerkschnittstellen.
- `ASGAverageNetworkOut` – Die durchschnittliche Anzahl gesendeter Bytes von einer einzigen Instance für alle Netzwerkschnittstellen.
- `ALBRequestCountPerTarget` – Die durchschnittliche Anzahl von Application Load Balancer-Anforderungen pro Ziel.

> **Important**
>
> Weitere wertvolle Informationen zu den Metriken für CPU-Auslastung, Netzwerk-I/O und Anzahl der Application Load Balancer-Anforderungen pro Ziel finden Sie im Thema Verfügbare CloudWatch Metriken für Ihre Instances auflisten im Amazon EC2 EC2-Benutzerhandbuch bzw. die CloudWatch Metriken für Ihren Application Load Balancer im Benutzerhandbuch für Application Load Balancers.

Sie können andere verfügbare CloudWatch Metriken oder Ihre eigenen Metriken auswählen, CloudWatch indem Sie eine benutzerdefinierte Metrik angeben. Sie müssen das AWS CLI oder ein SDK verwenden, um eine Zielverfolgungsrichtlinie mit einer benutzerdefinierten Metrikspezifikation zu erstellen. Ein Beispiel, das eine benutzerdefinierte Metrikspezifikation für eine Skalierungsrichtlinie für die Zielverfolgung mithilfe von spezifiziert AWS CLI, finden Sie unterBeispiel für Skalierungsrichtlinien für die AWS Command Line Interface (AWS CLI).

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

- Wir empfehlen, nur 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. Weitere Informationen zum Aktivieren der detaillierten Überwachung finden Sie unter Überwachung für Auto-Scaling-Instances konfigurieren.
- Nicht alle benutzerdefinierten Metriken funktionieren für die Zielverfolgung. Die Metrik muss eine gültige Auslastungsmetrik sein und beschreiben, wie ausgelastet eine Instance ist. Der Wert der Metrik muss sich proportional zur Anzahl der Instances in der Auto-Scaling-Gruppe erhöhen oder verringern. Das muss so sein, damit die Metrikdaten verwendet werden können, um die Anzahl der Instances proportional zu skalieren. Beispielsweise funktioniert die CPU-Auslastung einer Auto-Scaling-Gruppe (d. h. die Amazon EC2-Metrik `CPUUtilization` mit der Metrikdimension `AutoScalingGroupName`), wenn die Last auf der Auto-Scaling-Gruppe auf die Instances verteilt ist.
- Die folgenden Metriken funktionieren nicht für die Ziel-Nachverfolgung:

  - Die Anzahl der Anfragen, die vom Load Balancer empfangen werden, der der Auto-Scaling-Gruppe gegenüber liegt (d. h. die Elastic Load Balancing-Metrik `RequestCount`). Die Anzahl der Anfragen, die vom Load Balancer empfangen werden, ändert sich nicht basierend auf der Auslastung der Auto-Scaling-Gruppe.
  - Load Balancer-Anfragelatenz (d. h. die Elastic Load Balancing-Metrik `Latency`). Die Anfragelatenz kann aufgrund der zunehmenden Nutzung zunehmen, ändert sich aber nicht notwendigerweise proportional.
  - Die CloudWatch Amazon SQS SQS-Warteschlangenmetrik`ApproximateNumberOfMessagesVisible`. Die Anzahl der Nachrichten in einer Warteschlange ändert sich möglicherweise nicht proportional zur Größe der Auto-Scaling-Gruppe, die Nachrichten aus der Warteschlange verarbeitet. Eine benutzerdefinierte Metrik, welche die Anzahl der Nachrichten in der Warteschlange pro EC2-Instance in der Auto-Scaling-Gruppe misst, kann jedoch auch funktionieren. Weitere Informationen finden Sie unter Skalierung basierend auf Amazon SQS.
- Um die Metrik `ALBRequestCountPerTarget` zu verwenden, müssen Sie den Parameter `ResourceLabel` angeben, um die Load Balancer-Zielgruppe zu identifizieren, die der Metrik zugeordnet ist. Ein Beispiel, das den `ResourceLabel` Parameter für eine Skalierungsrichtlinie für die Zielverfolgung mithilfe von spezifiziert AWS CLI, finden Sie unterBeispiel für Skalierungsrichtlinien für die AWS Command Line Interface (AWS CLI).
- Wenn eine Metrik echte Werte von 0 ausgibt CloudWatch (z. B.`ALBRequestCountPerTarget`), kann eine Auto Scaling Scaling-Gruppe auf 0 skalieren, wenn über einen längeren Zeitraum kein Datenverkehr zu Ihrer Anwendung erfolgt. Damit Ihre Auto-Scaling-Gruppe auf 0 abskaliert werden kann, wenn keine Anfragen an sie weitergeleitet werden, muss die Mindestkapazität der Gruppe auf 0 festgelegt sein.
- 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 Zielnachverfolgungs-Skalierungsrichtlinie für Amazon EC2 Auto Scaling mit Metrikberechnungen.

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 die Auto-Scaling-Gruppe 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 Instance für einen Zeitraum von einer Minute dar.

Definieren Sie die Aufwärmzeit der Instanz

Sie können optional angeben, wie viele Sekunden die Vorbereitung einer neu gestarteten Instance dauert. Bis die angegebene Aufwärmzeit abgelaufen ist, wird eine Instance nicht auf die aggregierten EC2-Instance-Metriken der Auto Scaling Scaling-Gruppe angerechnet.

Während sich die Instances in der Aufwärmphase befinden, werden Ihre Skalierungsrichtlinien nur dann skaliert, wenn der Metrikwert von Instances, die sich nicht im Warmup befinden, größer ist als die Zielauslastung der Policy.

Wenn die Gruppe erneut skaliert wird, werden die Instances, die noch vorbereitet werden, als Teil der gewünschten Kapazität für die nächste Aufskalieraktivität gezählt. Der Zweck ist eine kontinuierliche (jedoch nicht exzessive) Erweiterung.

Während die Aufskalieraktivität läuft, werden alle durch Skalierungsrichtlinien initiierte Abskalieraktivitäten blockiert, bis die Instances vorbereitet wurden. Wenn die Instances mit dem Aufwärmen fertig sind und ein Abskalierungsereignis eintritt, werden alle Instances, die gerade beendet werden, bei der Berechnung der neuen gewünschten Kapazität auf die aktuelle Kapazität der Gruppe angerechnet. Deshalb entfernen wir nicht mehr Instances aus der Auto-Scaling-Gruppe als nötig.

Standardwert

Wenn kein Wert festgelegt ist, verwendet die Skalierungsrichtlinie den Standardwert. Dabei handelt es sich um den Wert für das für die Gruppe definierte Standardinstanz-Warmup. Wenn das Standard-Aufwärmen der Instanz Null ist, wird auf den Wert der Standard-Abklingzeit zurückgegriffen.
Wir empfehlen, den Standard-Instance-Warmup zu verwenden, um die Aktualisierung aller Skalierungsrichtlinien zu vereinfachen, wenn sich die Aufwärmzeit ändert.

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. Amazon EC2 Auto Scaling erstellt und verwaltet die CloudWatch Alarme, die Ihren Ziel-Tracking-Skalierungsrichtlinien zugeordnet sind, und löscht sie, wenn sie nicht mehr benötigt werden.
- Eine Skalierungsrichtlinie für die Zielverfolgung priorisiert die Verfügbarkeit bei Datenverkehrsschwankungen durch langsameres Abskalieren bei nachlassendem Datenverkehr. Wenn die Auto-Scaling-Gruppe sofort nach Abschluss einer Workload abskaliert werden soll, können Sie die Abskalierungskomponente der Richtlinie deaktivieren. Dadurch können Sie flexibel die Abskalierungsmethode nutzen, die Ihren Anforderungen bei geringer Auslastung am besten entspricht. Für eine möglichst schnelle Abskalierung empfiehlt es sich, keine einfache Skalierungsrichtlinie zu verwenden, um eine anschließende Ruhephase zu vermeiden.
- Wenn der Metrik Datenpunkte fehlen, führt dies dazu, dass der CloudWatch Alarmstatus auf geändert wird. `INSUFFICIENT_DATA` Amazon EC2 Auto Scaling kann Ihre Gruppe dann erst wieder skalieren, wenn neue Datenpunkte gefunden wurden.
- 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. Der Grund hierfür ist, dass wir konservativ agieren, indem beim Ermitteln der hinzuzufügenden oder zu entfernenden Instances Auf- oder Abrundungen vorgenommen werden. Dies hindert uns daran, eine unzureichende Anzahl von Instances hinzufügen oder zu viele Instances zu entfernen. Bei kleineren Auto-Scaling-Gruppen mit weniger Instances scheint die Auslastung der Gruppe jedoch weit vom Zielwert entfernt zu sein. Zum Beispiel: Sie setzen einen Zielwert von 50 Prozent für die CPU-Auslastung fest, und Ihre Auto-Scaling-Gruppe überschreitet dann diesen Zielwert. Wir könnten bestimmen, dass durch das Hinzufügen von 1,5 Instances die CPU-Auslastung auf beinahe 50 Prozent sinkt. Da es nicht möglich ist, 1,5 Instances hinzuzufügen, runden wir diesen Wert auf und fügen zwei Instances hinzu. Dadurch wird die CPU-Auslastung möglicherweise auf einen Wert unter 50 Prozent verringert, es wird jedoch sichergestellt, dass Ihre Anwendung über genügend Ressourcen verfügt, um dies zu unterstützen. Entsprechend entfernen wir nur eine Instance, wenn wir feststellen, dass das Entfernen von 1,5 Instances die CPU-Auslastung auf über 50 Prozent erhöht.

  Bei größeren Auto-Scaling-Gruppen mit mehr Instances wird die Auslastung auf eine größere Anzahl von Instances verteilt, wobei durch das Hinzufügen oder Entfernen von Instances eine kleinere Lücke zwischen dem Zielwert und den tatsächlichen metrischen Datenpunkten entsteht.

- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung geht davon aus, dass Ihre Auto-Scaling-Gruppe aufskaliert werden soll, wenn die angegebene Metrik über dem Zielwert liegt. Sie können keine Skalierungsrichtlinie für die Ziel-Nachverfolgung verwenden, um Ihre Auto-Scaling-Gruppe zu aufzuskalieren, wenn die angegebene Metrik unter dem Zielwert liegt.

Erstellen einer Zielverfolgungs-Skalierungsrichtlinie

Verwenden Sie eine der folgenden Methoden, um eine Skalierungsrichtlinie für die Zielverfolgung für Ihre Auto Scaling Scaling-Gruppe zu erstellen.

Bevor Sie beginnen, stellen Sie sicher, dass Ihre bevorzugte Metrik in Intervallen von 1 Minute verfügbar ist (im Vergleich zum Standardintervall von 5 Minuten für Amazon-EC2-Metriken).

Console

So erstellen Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung für eine neue Auto-Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
3. Wählen Sie in den Schritten 1, 2 und 3 die gewünschten Optionen aus, und fahren Sie mit Schritt 4: Konfigurieren von Gruppengrößen- und Skalierungsrichtlinien fort.
4. Geben Sie unter Skalierung den Bereich an, zwischen dem Sie skalieren möchten, indem Sie die minimale gewünschte Kapazität und maximale gewünschte Kapazität aktualisieren. Mit diesen beiden Einstellungen kann die Auto-Scaling-Gruppe dynamisch skaliert werden. Weitere Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.
5. Wählen Sie unter Automatische Skalierung Skalierungsrichtlinie für die Zielnachverfolgung aus.
6. Gehen Sie wie folgt vor, um eine Richtlinie zu definieren:

   a. Geben Sie einen Namen für die Richtlinie an.

   b. Wählen Sie unter Metriktyp einen Metriktyp aus.

      Wenn Sie Anzahl der Application Load Balancer pro Ziel auswählen, wählen Sie anschließend in Zielgruppe eine Zielgruppe aus.

c. Geben Sie einen Target value für die Metrik an.

d. (Optional) Aktualisieren Sie für das Aufwärmen der Instanz den Wert für das Aufwärmen der Instanz nach Bedarf.

e. (Optional) Wählen Sie Disable scale in to create only a scale-out policy (Abwärtsskalierung deaktivieren, um nur eine Richtlinie für die Aufwärtsskalierung zu erstellen). Auf diese Weise können Sie bei Bedarf eine separate Richtlinie für die horizontale Skalierung nach unten erstellen, die einen anderen Typ aufweist.

7. Fahren Sie mit dem Erstellen der Auto-Scaling-Gruppe fort. Ihre Skalierungsrichtlinie wird erstellt, nachdem die Auto-Scaling-Gruppe erstellt wurde.

So erstellen Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung für eine vorhandene Auto-Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Stellen Sie sicher, dass die Skalierungslimits entsprechend festgelegt sind. Wenn die gewünschte Kapazität der Gruppe z. B. bereits erreicht ist, müssen Sie ein neues Maximum angeben, um eine Aufskalierung durchführen zu können. Weitere Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.
4. Wählen Sie auf der Registerkarte Automatic scaling (Automatische Skalierung) unter Dynamic scaling policies (Dynamische Skalierungsrichtlinien) die Option Create dynamic scaling policy (Richtlinie für die dynamische Skalierung erstellen) aus.
5. Gehen Sie wie folgt vor, um eine Richtlinie zu definieren:

   a. Für den Richtlinientyp behalten Sie die Standardeinstellung für Zielverfolgungsskalierung bei.

   b. Geben Sie einen Namen für die Richtlinie an.

   c. Wählen Sie unter Metriktyp einen Metriktyp aus. Sie können nur einen Metriktyp auswählen. Um mehr als eine Metrik zu verwenden, erstellen Sie mehrere Richtlinien.

      Wenn Sie Anzahl der Application Load Balancer pro Ziel auswählen, wählen Sie anschließend in Zielgruppe eine Zielgruppe aus.

d. Geben Sie einen Target value für die Metrik an.

e. (Optional) Aktualisieren Sie für das Aufwärmen der Instanz den Wert für das Aufwärmen der Instanz nach Bedarf.

f. (Optional) Wählen Sie Disable scale in to create only a scale-out policy (Abwärtsskalierung deaktivieren, um nur eine Richtlinie für die Aufwärtsskalierung zu erstellen). Auf diese Weise können Sie bei Bedarf eine separate Richtlinie für die horizontale Skalierung nach unten erstellen, die einen anderen Typ aufweist.

6. Wählen Sie Erstellen.

AWS CLI

Um eine Skalierungsrichtlinie für die Zielverfolgung zu erstellen, können Sie das folgende Beispiel verwenden, um Ihnen den Einstieg zu erleichtern. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

> **Note**
>
> Weitere Beispiele finden Sie unter Beispiel für Skalierungsrichtlinien für die AWS Command Line Interface (AWS CLI).

So erstellen Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung (AWS CLI)

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": "ASGAverageCPUUtilization"
    }
}
```

Weitere Informationen finden Sie unter PredefinedMetricSpezifikation in der Amazon EC2 Auto Scaling API-Referenz.

2. Verwenden Sie den Befehl put-scaling-policy zusammen mit der Datei `config.json`, die Sie im vorherigen Schritt erstellt haben, um Ihre Skalierungsrichtlinie zu erstellen.

```
aws autoscaling put-scaling-policy --policy-name cpu50-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-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:us-
west-2:123456789012:scalingPolicy:228f02c2-c665-4bfd-
aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/cpu50-target-tracking-
scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e",
            "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2",
            "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
        }
    ]
}
```

Erstellen einer Zielnachverfolgungs-Skalierungsrichtlinie für Amazon EC2 Auto Scaling mit Metrikberechnungen

Mithilfe der metrischen 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 GetMetricDaten-API verwenden.

> **Note**
>
> Sie können mithilfe von metrischer Mathematik nur dann eine Skalierungsrichtlinie für die Zielverfolgung erstellen, wenn Sie das AWS CLI AWS CloudFormation, oder ein SDK verwenden. Dieses Feature ist auf der Konsole noch nicht verfügbar.

Beispiel: Amazon-SQS-Warteschlangenrückstand pro Instance

Um den Amazon-SQS-Warteschlangenrückstand pro Instance zu erhalten, nehmen Sie die ungefähre Anzahl der Nachrichten, die für den Abruf aus der Warteschlange zur Verfügung stehen, und dividieren diese Zahl durch die laufende Kapazität der Auto-Scaling-Gruppe im `InService`-Zustand. Weitere Informationen finden Sie unter Skalierung basierend auf Amazon SQS.

Die Logik für den Ausdruck lautet wie folgt:

```
sum of (number of messages in the queue)/(number of InService instances)
```

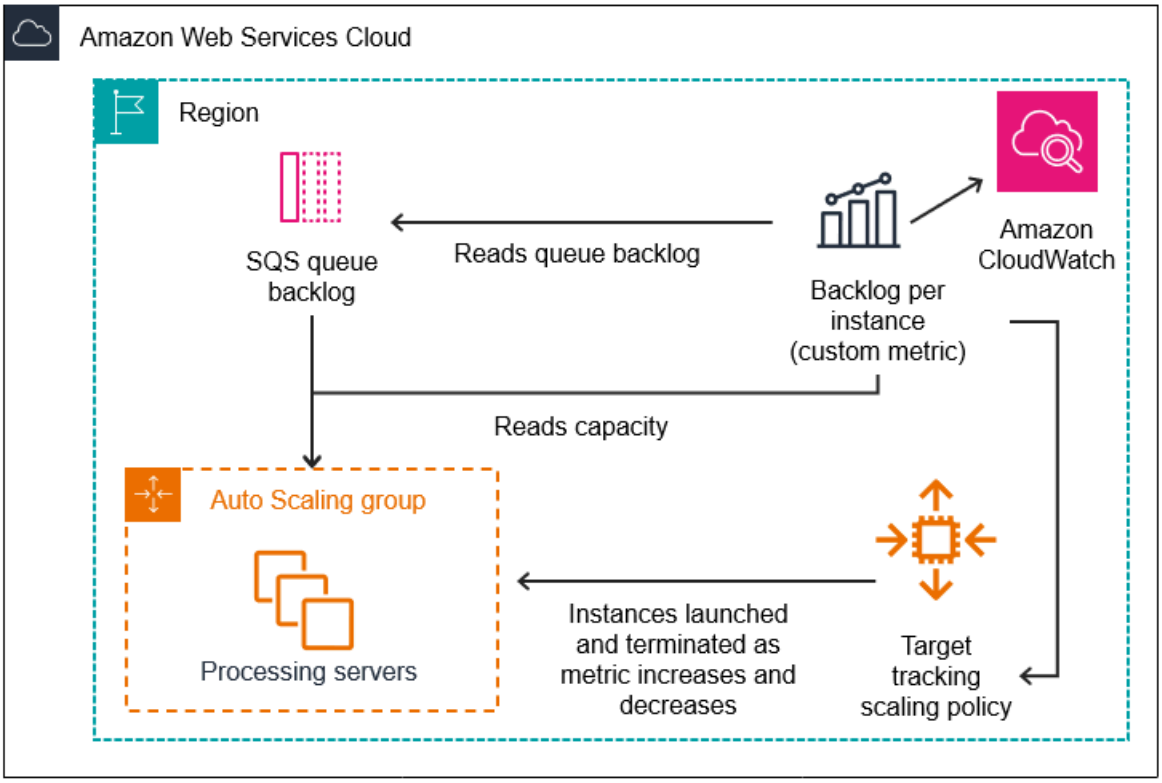
Dann lauten Ihre CloudWatch Metrikinformationen wie folgt.

ID	CloudWatch metrisch	Statistik	Intervall
m1	ApproximateNumberOfMessagesSichtbar	Summe	1 Minute
m2	GroupInServiceInstances	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 group size (the number of InService instances)",
                "Id": "m2",
                "MetricStat": {
                    "Metric": {
                        "MetricName": "GroupInServiceInstances",
                        "Namespace": "AWS/AutoScaling",
                        "Dimensions": [
                            {
                                "Name": "AutoScalingGroupName",
                                "Value": "my-asg"
                            }
```

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

Weitere Informationen finden Sie unter TargetTrackingKonfiguration in der Amazon EC2 Auto Scaling API-Referenz.

> **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 Befehl put-scaling-policy mit der JSON-Datei als Eingabe wie im folgenden Beispiel beschrieben aus.

```
aws autoscaling put-scaling-policy --policy-name sqs-backlog-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-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-2:123456789012:scalingPolicy:228f02c2-c665-4bfd-
aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/sqs-backlog-target-
tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e",
            "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2",
            "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
        }
    ]
}
```

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

Schrittweise und einfache Skalierungsrichtlinien für Amazon EC2 Auto Scaling

Schrittweise Skalierung und einfache Skalierungsrichtlinien skalieren die Kapazität Ihrer Auto Scaling Scaling-Gruppe 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 schrittweiser Skalierung und einfacher Skalierung erstellen und verwalten Sie die CloudWatch Alarme, die den Skalierungsprozess auslösen. Wenn ein Alarm verletzt wird, initiiert Amazon EC2 Auto Scaling die mit diesem Alarm verknüpfte Skalierungsrichtlinie.

Wir empfehlen dringend, Skalierungsrichtlinien für die Zielverfolgung zu verwenden, um anhand von Kennzahlen wie der durchschnittlichen CPU-Auslastung oder der durchschnittlichen Anzahl von Anfragen pro Ziel zu skalieren. Metriken, die sich verringern, wenn die Kapazität zunimmt, und zunehmen, wenn die Kapazität abnimmt, können zur proportionalen Aufwärts- oder Abwärtsskalierung der Anzahl der Instances verwendet werden, welche die Zielverfolgung verwenden. Dadurch wird sichergestellt, dass Amazon EC2 Auto Scaling die Bedarfskurve für Ihre Anwendungen genau einhält. Weitere Informationen finden Sie unter Skalierungsrichtlinien für die Ziel-Nachverfolgung.

Inhalt

Funktionsweise von Skalierungsrichtlinien

Um Step Scaling zu verwenden, erstellen Sie zunächst einen CloudWatch Alarm, der eine Metrik für Ihre Auto Scaling Scaling-Gruppe überwacht. Sie definieren die Metrik, den Schwellenwert und die Anzahl der Bewertungszeiträume, die einen Alarmverstoß bestimmen. Erstellen Sie dann eine Richtlinie zur schrittweisen Skalierung, die definiert, wie Ihre Gruppe skaliert werden soll, wenn der Alarmschwellenwert überschritten wird.

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

- Skalieren Sie um 10 Instanzen, wenn die Alarmmetrik 60 Prozent erreicht
- Skalieren Sie die Anzeige um 30 Instanzen, wenn die Alarmmetrik 75 Prozent erreicht
- Skalieren Sie die Skala um 40 Instanzen, wenn die Alarmmetrik 85 Prozent erreicht

Wenn der Alarmschwellenwert für die angegebene Anzahl von Testzeiträumen überschritten wird, wendet Amazon EC2 Auto Scaling die in der Richtlinie definierten schrittweisen Anpassungen an. Die Anpassungen können bei weiteren Überschreitungen des Alarms fortgesetzt werden, bis der Alarmstatus 0K wieder erreicht ist.

Jede Instance hat eine Aufwärmphase, um zu verhindern, dass Skalierungsaktivitäten zu reaktiv auf Änderungen reagieren, die sich über kurze Zeiträume ergeben. Sie können optional die Aufwärmphase für Ihre Skalierungsrichtlinie konfigurieren. Wir empfehlen jedoch, das standardmäßige Aufwärmen der Instanz zu verwenden, um die Aktualisierung aller Skalierungsrichtlinien zu vereinfachen, wenn sich die Aufwärmzeit ändert. Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

Einfache Skalierungsrichtlinien ähneln den Richtlinien zur schrittweisen Skalierung, außer dass sie auf einer einzigen Skalierungsanpassung basieren und zwischen den einzelnen Skalierungsaktivitäten eine Abklingzeit besteht. Weitere Informationen finden Sie unter Einfache Skalierungsrichtlinien.

Schrittanpassungen für die Schrittskalierung

Wenn Sie eine Richtlinie zur schrittweise Skalierung erstellen, geben Sie eine oder mehrere Stufenanpassungen an, die automatisch die Anzahl der Instances 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 Metrik, die Ihrem Alarm zugeordnet ist. CloudWatch Wenn der Alarm ausgelöst wird, wird die entsprechende Skalierungsrichtlinie ausgelöst. Amazon EC2 Auto Scaling wendet den 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 für jede Richtlinie eine Reihe von schrittweisen Anpassungen mit dem Anpassungstyp `PercentChangeInCapacity` (oder Prozent der Gruppe in der Konsole) 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%
-------------------------------------------------------------------------
```

Nehmen wir nun an, Sie verwenden diese Skalierungskonfiguration für eine Auto Scaling Scaling-Gruppe, die sowohl eine aktuelle als auch eine gewünschte Kapazität von 10 hat. Die folgenden Punkte fassen das Verhalten der Skalierungskonfiguration in Bezug auf die gewünschte und aktuelle Kapazität der Gruppe zusammen:

- Die gewünschte und die aktuelle Kapazität werden aufrechterhalten, solange der aggregierte Metrikwert größer als 40 und kleiner als 60 ist.
- Steigt der Metrikwert auf 60, wird die gewünschte Kapazität der Gruppe auf Grundlage der zweiten Schrittanpassung der Richtlinie für die horizontale Skalierung nach oben (Erhöhen um 10 % von 10 Instances) um 1 Instance auf 11 Instances erhöht. Nachdem die neue Instanz ausgeführt wird und die angegebene Aufwärmzeit abgelaufen ist, erhöht sich die aktuelle Kapazität der Gruppe auf 11 Instanzen. Steigt der Metrikwert auch nach dieser Kapazitätserhöhung auf 70, erhöht sich die gewünschte Kapazität der Gruppe um weitere 3 Instances auf 14 Instances. Dies basiert auf der dritten Schrittanpassung der Richtlinie für die horizontale Skalierung nach oben (Erhöhung um 30 Prozent von 11 Instances, 3,3 Instances, abgerundet auf 3 Instances).
- Fällt der Metrikwert auf 40 ab, wird die gewünschte Kapazität der Gruppe auf Grundlage der zweiten Schrittanpassung der Richtlinie für die horizontale Skalierung nach unten (Verringern um 10 % von 14 Instances, 1,4 Instances abgerundet auf 1 Instance) um 1 Instance auf 13 Instances reduziert. Wenn der Metrikwert selbst nach dieser Abnahme der Kapazität auf 30 fällt, sinkt die gewünschte Kapazität der Gruppe um weitere 3 Instances auf 10 Instances. Dies basiert auf der dritten Schrittanpassung der Richtlinie für die horizontale Skalierung nach unten (Abzug um 30 Prozent von 13 Instances, 3,9 Instances, abgerundet auf 3 Instances).

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

- Wenn Sie die verwenden AWS Management Console, geben Sie die Ober- und Untergrenzen als absolute Werte an. Wenn Sie das AWS CLI oder ein SDK verwenden, geben Sie die Ober- und Untergrenzen relativ zum Schwellenwert für Sicherheitsverletzungen an.
- 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 Ihrer Auto-Scaling-Gruppe oder in Kapazitätseinheiten angeben. Normalerweise bedeutet eine Kapazitätseinheit eine Instanz, es sei denn, Sie verwenden die Funktion zur Gewichtung von Instanzen.

Amazon EC2 Auto Scaling unterstützt die folgenden Anpassungstypen für die schrittweise und die einfache Skalierung:

- `ChangeInCapacity` – Erhöhen oder Verringern der aktuellen Kapazität der Gruppe um den angegebenen Wert. Ein positiver Wert erhöht die Kapazität, ein negativer Anpassungswert verringert die Kapazität. Beispiel: Wenn die aktuelle Kapazität der Gruppe 3 und die Anpassung 5 beträgt, dann fügen wir bei der Durchführung dieser Richtlinie 5 Kapazitätseinheiten zur Kapazität hinzu, insgesamt also 8 Kapazitätseinheiten.
- `ExactCapacity` – Ändern Sie die aktuelle Kapazität der Gruppe auf den angegebenen Wert. Geben Sie bei diesem Anpassungstyp einen nicht-negativen Wert an. Beispiel: Wenn die aktuelle Kapazität der Gruppe 3 und die Anpassung 5 beträgt, dann ändern wir bei der Durchführung dieser Richtlinie die Kapazität auf 5 Kapazitätseinheiten.
- `PercentChangeInCapacity` – Erhöhen oder Verringern der aktuellen Kapazität der Gruppe um den angegebenen Prozentsatz. Ein positiver Wert erhöht die Kapazität, ein negativer Anpassungswert verringert die Kapazität. Beispiel: Wenn die aktuelle Kapazität 10 und die Anpassung 10 Prozent beträgt, dann fügen wir bei der Durchführung dieser Richtlinie 1 Kapazitätseinheit zur Kapazität hinzu, insgesamt also 11 Kapazitätseinheiten.

> **Note**
>
> Handelt es sich bei dem resultierenden Wert nicht um eine ganze Zahl, wird wie folgt gerundet:

- 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 `PercentChangeInCapacity` können Sie auch die minimale Anzahl von Instances angeben, die mit dem `MinAdjustmentMagnitude`-Parameter skaliert werden sollen. Angenommen, Sie erstellen eine Richtlinie zum Hinzufügen von 25 % und geben an, dass mindestens 2 Instances hinzugefügt werden sollen. Wenn Sie über eine Auto-Scaling-Gruppe mit 4 Instances verfügen und die Skalierungsrichtlinie umgesetzt wird, ergeben 25 % von 4 Instances 1 Instance. Da Sie aber angegeben haben, dass mindestens 2 Instances hinzugefügt werden sollen, werden 2 Instances hinzugefügt.

Wenn Sie Instance-Gewichtungen verwenden, ändert sich der Effekt, wenn Sie den `MinAdjustmentMagnitude` Parameter auf einen Wert ungleich Null setzen. Der Wert wird in Kapazitätseinheiten angegeben. Um die Mindestanzahl der zu skalierenden Instances festzulegen, legen Sie diesen Parameter auf einen Wert fest, der mindestens so groß ist wie die größte Instance-Gewichtung.

Wenn Sie Instance-Gewichtungen verwenden, denken Sie daran, dass die aktuelle Kapazität Ihrer Auto Scaling Scaling-Gruppe bei Bedarf die gewünschte Kapazität überschreiten kann. Wenn Ihre absolute Zahl, die zu verringern ist, oder der Betrag, den der Prozentsatz zum Verringern angibt, geringer ist als die Differenz zwischen der aktuellen und der gewünschten Kapazität, wird keine Skalierungsaktion durchgeführt. Sie müssen diese Verhalten berücksichtigen, wenn Sie sich das Ergebnis einer Skalierungsrichtlinie ansehen, wenn die Schwelle eines Alarms überschritten wird. Angenommen, die gewünschte Kapazität beträgt 30 und die aktuelle Kapazität 32. Wenn beim Auslösen des Alarms die gewünschte Kapazität mittels Skalierungsrichtlinie um 1 verringert wird, wird keine Skalierungsaktion durchgeführt.

Instance-Aufwärmphase

Sie können für Stufenskalierung optional angeben, wie viele Sekunden die Vorbereitung einer neu gestarteten Instance dauert. Bis die angegebene Aufwärmzeit abgelaufen ist, wird eine Instance nicht auf die aggregierten EC2-Instance-Metriken der Auto Scaling Scaling-Gruppe angerechnet.

Solange sich die Instances in der Aufwärmphase befinden, werden Ihre Skalierungsrichtlinien nur dann skaliert, wenn der Metrikwert von Instances, die sich nicht in der Warmlaufphase befinden, den höchsten Alarmschwellenwert der Richtlinie überschreitet.

Wenn die Gruppe erneut skaliert wird, werden die Instances, die noch vorbereitet werden, als Teil der gewünschten Kapazität für die nächste Aufskalieraktivität gezählt. Daher führen mehrere Alarmüberschreitungen, die in den Bereich derselben Schrittanpassung fallen, zu einer einzigen Skalierung. Der Zweck ist eine kontinuierliche (jedoch nicht exzessive) Erweiterung.

Nehmen wir an, Sie erstellen eine Richtlinie mit zwei Schritten. Im ersten Schritt werden 10 Prozent hinzugefügt, wenn die Metrik 60 erreicht, und im zweiten Schritt werden 30 Prozent hinzugefügt, wenn die Metrik 70 Prozent erreicht. Ihre Auto-Scaling-Gruppe hat eine gewünschte und eine aktuelle Kapazität von 10. Die gewünschte und die aktuelle Kapazität ändern sich nicht, solange der aggregierte Metrikwert kleiner als 60 ist. Nehmen wir an, die Metrik erreicht den Wert 60, sodass 1 Instance hinzugefügt wird (10 Prozent von 10 Instances). Dann erreicht die Metrik den Wert 62, während die neue Instance sich noch in der Aufwärmphase befindet. Die Skalierungsrichtlinie berechnet die neue gewünschte Kapazität auf Grundlage der aktuellen Kapazität, die immer noch 10 beträgt. Allerdings ist die gewünschte Kapazität der Gruppe bereits auf 11 Instances gestiegen, weswegen die Skalierungsrichtlinie die gewünschte Kapazität nicht weiter erhöht. Erreicht die Metrik jedoch den Wert 70, während sich die neue Instance noch in der Aufwärmphase befindet, müssen wir 3 Instances (30 % von 10 Instances) hinzufügen. Die gewünschte Kapazität der Gruppe beträgt jedoch bereits 11, sodass wir für die jetzt gewünschte Kapazität von 13 Instances nur weitere 2 Instances hinzufügen.

Während die Aufskalieraktivität läuft, werden alle durch Skalierungsrichtlinien initiierte Abskalieraktivitäten blockiert, bis die Instances vorbereitet wurden. Wenn die Instances mit dem Aufwärmen fertig sind und ein Abskalierungsereignis eintritt, werden alle Instances, die gerade beendet werden, bei der Berechnung der neuen gewünschten Kapazität auf die aktuelle Kapazität der Gruppe angerechnet. Deshalb entfernen wir nicht mehr Instances aus der Auto-Scaling-Gruppe als nötig. Wenn beispielsweise eine Instance bereits beendet ist und ein Alarm im Bereich der gleichen Schrittanpassung auftritt, die die gewünschte Kapazität um 1 verringert hat, wird keine Skalierungsmaßnahme ergriffen.

Standardwert

Wenn kein Wert festgelegt ist, verwendet die Skalierungsrichtlinie den Standardwert. Dabei handelt es sich um den Wert für das für die Gruppe definierte Standardinstanz-Warmup. Wenn das Standard-Aufwärmen der Instanz Null ist, wird auf den Wert der Standard-Abklingzeit zurückgegriffen.

Überlegungen

Bei der Arbeit mit Richtlinien zur schrittweisen und einfachen 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.

Erstellen Sie eine Richtlinie zur schrittweisen Skalierung für die horizontale Skalierung

Verwenden Sie eine der folgenden Methoden, um eine schrittweise Skalierungsrichtlinie für die horizontale Skalierung für Ihre Auto Scaling-Gruppe zu erstellen:

Console

Schritt 1: Erstellen Sie einen CloudWatch Alarm für den hohen Schwellenwert der Metrik

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Ändern Sie, falls erforderlich, die Region. Wählen Sie auf der Navigationsleiste die Region aus, in der sich Ihre Auto-Scaling-Gruppe befindet.
3. Wählen Sie im Navigationsbereich Alarms > All alarms (Alarme > Alle Alarme) und anschließend Create alarm (Alarm erstellen) aus.
4. Wählen Sie Select metric (Metrik auswählen) aus.
5. Wählen Sie auf der Registerkarte Alle Metriken die Option EC2 und Nach Auto-Scaling-Gruppe aus und geben Sie den Namen der Auto-Scaling-Gruppe in das Suchfeld ein. Wählen Sie dann `CPUUtilization` und anschließend Metrik auswählen aus. Die Seite Specify metric and conditions (Metrik und Bedingungen festlegen) mit einem Diagramm und weiteren Informationen über die Metrik werden anzeigt.

6. Wählen Sie unter Period (Zeitraum) den Auswertungszeitraum für den Alarm aus, z. B. 1 Minute. Beim Auswerten des Alarms wird jeder Zeitraum in einem Datenpunkt zusammengefasst.

   > **Note**
   >
   > Ein kürzerer Zeitraum erzeugt eine höhere Alarmempfindlichkeit.

7. Führen Sie unter Bedingungen die folgenden Schritte aus:

   - Wählen Sie für Threshold type (Schwellenwerttyp) die Option Static (Statisch) aus.
   - Geben Sie für Whenever **`CPUUtilization`** is an, ob der Wert der Metrik größer oder größer als oder gleich dem Schwellenwert sein soll, ab dem der Alarm überschritten werden kann. Geben Sie dann unter than (als) den Schwellenwert ein, der den Alarm auslösen soll.

     > **Important**
     >
     > Für einen Alarm, der mit einer Scale-Out-Richtlinie (Metrik hoch) verwendet werden soll, stellen Sie sicher, dass Sie nicht weniger als oder weniger als oder gleich dem Schwellenwert wählen.

8. Führen Sie unter Zusätzliche Konfiguration die folgenden Schritte aus:

   - Geben Sie unter Datenpunkte zum Alarm die Anzahl der Datenpunkte (Auswertungszeiträume) ein, während denen der Metrikwert die Schwellenbedingungen des Alarms erfüllen muss. So würde es bei zwei aufeinanderfolgenden Zeiträume von je 5 Minuten z. B. 10 Minuten dauern, den Alarmstatus auszulösen.
   - Wählen Sie für Fehlende Datenbehandlung die Option Fehlende Daten als ungültig behandeln (Überschreitungsschwelle) aus. Weitere Informationen finden Sie unter Konfiguration der Behandlung fehlender Daten durch CloudWatch Alarme im CloudWatch Amazon-Benutzerhandbuch.

9. Wählen Sie Weiter aus.

   Die Seite Configure actions (Konfigurieren von Aktionen) wird angezeigt.

10. Wählen Sie unter Notification (Benachrichtigung) ein Amazon-SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Zustand `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.

Damit der Alarm keine Benachrichtigungen sendet, wählen Sie Remove (Entfernen).

11. Sie können die anderen Abschnitte der Seite Configure actions (Konfigurieren von Aktionen) leer lassen. Wenn Sie die anderen Abschnitte leer lassen, wird ein Alarm erstellt, ohne diesen einer Skalierungsrichtlinie zuzuordnen. Sie können den Alarm dann über die Amazon EC2 Auto Scaling-Konsole mit einer Skalierungsrichtlinie verknüpfen.
12. Wählen Sie Weiter aus.
13. Geben Sie einen Namen (beispielsweise `Step-Scaling-AlarmHigh-AddCapacity`) und optional eine Beschreibung des Alarms ein. Wählen Sie anschließend Next (Weiter) aus.
14. Wählen Sie Alarm erstellen aus.

Gehen Sie wie folgt vor, um dort weiterzumachen, wo Sie nach der Erstellung Ihres CloudWatch Alarms aufgehört haben.

Schritt 2: Erstellen Sie eine Richtlinie zur schrittweisen Skalierung für die horizontale Skalierung

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Stellen Sie sicher, dass die Skalierungslimits entsprechend festgelegt sind. Wenn die gewünschte Kapazität der Gruppe z. B. bereits erreicht ist, müssen Sie ein neues Maximum angeben, um eine Aufskalierung durchführen zu können. Weitere Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.
4. Wählen Sie auf der Registerkarte Automatic scaling (Automatische Skalierung) unter Dynamic scaling policies (Dynamische Skalierungsrichtlinien) die Option Create dynamic scaling policy (Richtlinie für die dynamische Skalierung erstellen) aus.
5. Wählen Sie als Richtlinientyp die Option Step Scaling aus, und geben Sie dann einen Namen für die Richtlinie an.
6. Wählen Sie für CloudWatch Alarm Ihren Alarm aus. Wenn Sie noch keinen Alarm erstellt haben, wählen Sie Alarm erstellen und führen Sie die Schritte 4 bis 14 des vorherigen Verfahrens aus, um einen Alarm zu erstellen. CloudWatch

7. Geben Sie die Änderung der aktuellen Gruppengröße an, die diese Richtlinie vornehmen soll, wenn sie mit Take the action (Aktion ausführen) ausgeführt wird. Sie können eine bestimmte Anzahl von Instances oder einen Prozentsatz der vorhandenen Gruppengröße hinzufügen, oder die Gruppe auf eine genaue Größe festlegen.

   Um beispielsweise eine Scale-Out-Richtlinie zu erstellen, die die Kapazität der Gruppe um 30 Prozent erhöht, wählen SieAdd, geben Sie `30` in das nächste Feld ein, und wählen Sie dann. `percent of group` Standardmäßig ist die Untergrenze dieser Schrittanpassung der Alarmschwellenwert, und die Obergrenze ist positive (+) Unendlichkeit.
8. Um einen weiteren Schritt hinzuzufügen, wählen Sie Add step (Schritt hinzufügen) und definieren dann den Betrag, um den skaliert werden soll, sowie die untere und obere Grenze des Schritts relativ zum Alarmschwellenwert.
9. Um eine Mindestanzahl von zu skalierenden Instances festzulegen, aktualisieren Sie das Zahlenfeld unter Add capacity units in increments of at least (Kapazitätseinheiten hinzufügen in Schritten von mindestens) `1` Kapazitätseinheiten.
10. (Optional) Aktualisieren Sie für Instance-Warmup den Instanz-Warmup-Wert nach Bedarf.
11. Wählen Sie Erstellen.

AWS CLI

Um eine Richtlinie zur schrittweisen Skalierung für die horizontale Skalierung (Erhöhung der Kapazität) zu erstellen, können Sie die folgenden Beispielbefehle verwenden. Ersetzen Sie jedes *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Wenn Sie die verwenden AWS CLI, erstellen Sie zunächst eine Richtlinie zur schrittweisen Skalierung, die Amazon EC2 Auto Scaling Anweisungen zur Skalierung bei steigendem Wert einer Metrik bereitstellt. Anschließend erstellen Sie den Alarm, indem Sie die zu überwachende Metrik identifizieren, den Schwellenwert für die Metrik und andere Details für die Alarme definieren und den Alarm der Skalierungsrichtlinie zuordnen.

Schritt 1: Erstellen Sie eine Richtlinie für Scale-Out

Verwenden Sie den folgenden Befehl put-scaling-policy, um eine schrittweise Skalierungsrichtlinie mit dem Namen`my-step-scale-out-policy`, mit einem Anpassungstyp zu erstellen, der `PercentChangeInCapacity` die Kapazität der Gruppe auf der Grundlage der folgenden schrittweisen Anpassungen erhöht (unter der Annahme eines CloudWatch Alarmschwellenwerts von 60 Prozent):

- Erhöhen Sie die Anzahl der Instances um 10 Prozent, wenn der Wert der Metrik größer oder gleich 60 Prozent, aber kleiner als 75 Prozent ist.
- Erhöhen Sie die Anzahl der Instances um 20 Prozent, wenn der Wert der Metrik größer oder gleich 75 Prozent, aber kleiner als 85 Prozent ist.
- Erhöhen Sie die Anzahl der Instances um 30 Prozent, wenn der Wert der Metrik größer oder gleich 85 Prozent ist.

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg  \
  --policy-name my-step-scale-out-policy \
  --policy-type StepScaling \
  --adjustment-type PercentChangeInCapacity \
  --metric-aggregation-type Average \
  --step-adjustments
 MetricIntervalLowerBound=0.0,MetricIntervalUpperBound=15.0,ScalingAdjustment=10 \

 MetricIntervalLowerBound=15.0,MetricIntervalUpperBound=25.0,ScalingAdjustment=20 \
                     MetricIntervalLowerBound=25.0,ScalingAdjustment=30 \
  --min-adjustment-magnitude 1
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen ihn, um einen CloudWatch Alarm für die Richtlinie zu erstellen.

```
{
    "PolicyARN":
 "arn:aws:autoscaling:region:123456789012:scalingPolicy:4ee9e543-86b5-4121-b53b-
aa4c23b5bbcc:autoScalingGroupName/my-asg:policyName/my-step-scale-in-policy
}
```

Schritt 2: Erstellen Sie einen CloudWatch Alarm für den hohen Schwellenwert der Metrik

Verwenden Sie den folgenden Befehl CloudWatch put-metric-alarm, um einen Alarm zu erstellen, der die Auto Scaling Scaling-Gruppe auf der Grundlage eines durchschnittlichen CPU-Schwellenwerts von 60 Prozent für mindestens zwei aufeinanderfolgende Evaluierungsperioden von zwei Minuten vergrößert. Geben Sie zum Verwenden einer selbst erstellten Metrik den Namen der Metrik im Feld `--metric-name` und ihren Namespace im Feld `--namespace` an.

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmHigh-AddCapacity \
```

```
--metric-name CPUUtilization --namespace AWS/EC2 --statistic Average \
--period 120 --evaluation-periods 2 --threshold 60 \
--comparison-operator GreaterThanOrEqualToThreshold \
--dimensions "Name=AutoScalingGroupName,Value=my-asg" \
--alarm-actions PolicyARN
```

Erstellen Sie eine Richtlinie zur schrittweisen Skalierung für die Skalierung

Verwenden Sie eine der folgenden Methoden, um eine schrittweise Skalierungsrichtlinie für die Skalierung für Ihre Auto Scaling-Gruppe zu erstellen:

Console

Schritt 1: Erstellen Sie einen CloudWatch Alarm für den niedrigen Schwellenwert der Metrik

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Ändern Sie, falls erforderlich, die Region. Wählen Sie auf der Navigationsleiste die Region aus, in der sich Ihre Auto-Scaling-Gruppe befindet.
3. Wählen Sie im Navigationsbereich Alarms > All alarms (Alarme > Alle Alarme) und anschließend Create alarm (Alarm erstellen) aus.
4. Wählen Sie Select metric (Metrik auswählen) aus.
5. Wählen Sie auf der Registerkarte Alle Metriken die Option EC2 und Nach Auto-Scaling-Gruppe aus und geben Sie den Namen der Auto-Scaling-Gruppe in das Suchfeld ein. Wählen Sie dann `CPUUtilization` und anschließend Metrik auswählen aus. Die Seite Specify metric and conditions (Metrik und Bedingungen festlegen) mit einem Diagramm und weiteren Informationen über die Metrik werden anzeigt.
6. Wählen Sie unter Period (Zeitraum) den Auswertungszeitraum für den Alarm aus, z. B. 1 Minute. Beim Auswerten des Alarms wird jeder Zeitraum in einem Datenpunkt zusammengefasst.

   > **Note**
   >
   > Ein kürzerer Zeitraum erzeugt eine höhere Alarmempfindlichkeit.

7. Führen Sie unter Bedingungen die folgenden Schritte aus:

   - Wählen Sie für Threshold type (Schwellenwerttyp) die Option Static (Statisch) aus.

- Geben Sie für Whenever **`CPUUtilization`** is an, ob der Wert der Metrik kleiner oder kleiner als oder gleich dem Schwellenwert sein soll, ab dem der Alarm überschritten werden kann. Geben Sie dann unter than (als) den Schwellenwert ein, der den Alarm auslösen soll.

> **Important**
>
> Stellen Sie sicher, dass Sie für einen Alarm, der mit einer Scale-in-Richtlinie (Metrik niedrig) verwendet werden soll, nicht größer als oder größer als oder gleich dem Schwellenwert wählen.

8. Führen Sie unter Zusätzliche Konfiguration die folgenden Schritte aus:

   - Geben Sie unter Datenpunkte zum Alarm die Anzahl der Datenpunkte (Auswertungszeiträume) ein, während denen der Metrikwert die Schwellenbedingungen des Alarms erfüllen muss. So würde es bei zwei aufeinanderfolgenden Zeiträume von je 5 Minuten z. B. 10 Minuten dauern, den Alarmstatus auszulösen.
   - Wählen Sie für Fehlende Datenbehandlung die Option Fehlende Daten als ungültig behandeln (Überschreitungsschwelle) aus. Weitere Informationen finden Sie unter Konfiguration der Behandlung fehlender Daten durch CloudWatch Alarme im CloudWatch Amazon-Benutzerhandbuch.

9. Wählen Sie Weiter aus.

   Die Seite Configure actions (Konfigurieren von Aktionen) wird angezeigt.

10. Wählen Sie unter Notification (Benachrichtigung) ein Amazon-SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Zustand `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

    Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.

    Damit der Alarm keine Benachrichtigungen sendet, wählen Sie Remove (Entfernen).

11. Sie können die anderen Abschnitte der Seite Configure actions (Konfigurieren von Aktionen) leer lassen. Wenn Sie die anderen Abschnitte leer lassen, wird ein Alarm erstellt, ohne diesen einer Skalierungsrichtlinie zuzuordnen. Sie können den Alarm dann über die Amazon EC2 Auto Scaling-Konsole mit einer Skalierungsrichtlinie verknüpfen.

12. Wählen Sie Weiter aus.

13. Geben Sie einen Namen (beispielsweise `Step-Scaling-AlarmLow-RemoveCapacity`) und optional eine Beschreibung des Alarms ein. Wählen Sie anschließend Next (Weiter) aus.
14. Wählen Sie Alarm erstellen aus.

Gehen Sie wie folgt vor, um dort weiterzumachen, wo Sie nach der Erstellung Ihres CloudWatch Alarms aufgehört haben.

Schritt 2: Erstellen Sie eine Richtlinie zur schrittweisen Skalierung für die Skalierung

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Stellen Sie sicher, dass die Skalierungslimits entsprechend festgelegt sind. Wenn zum Beispiel die gewünschte Kapazität Ihrer Gruppe bereits erreicht ist, müssen Sie für die Skalierung eine neue Mindestkapazität angeben. Weitere Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.
4. Wählen Sie auf der Registerkarte Automatic scaling (Automatische Skalierung) unter Dynamic scaling policies (Dynamische Skalierungsrichtlinien) die Option Create dynamic scaling policy (Richtlinie für die dynamische Skalierung erstellen) aus.
5. Wählen Sie als Richtlinientyp die Option Schrittweise Skalierung aus, und geben Sie dann einen Namen für die Richtlinie an.
6. Wählen Sie für CloudWatch Alarm Ihren Alarm aus. Wenn Sie noch keinen Alarm erstellt haben, wählen Sie Alarm erstellen und führen Sie die Schritte 4 bis 14 des vorherigen Verfahrens aus, um einen Alarm zu erstellen. CloudWatch
7. Geben Sie die Änderung der aktuellen Gruppengröße an, die diese Richtlinie vornehmen soll, wenn sie mit Take the action (Aktion ausführen) ausgeführt wird. Sie können eine bestimmte Anzahl von Instances oder einen Prozentsatz der vorhandenen Gruppengröße entfernen, oder die Gruppe auf eine genaue Größe festlegen.

   Um beispielsweise eine Scale-In-Richtlinie zu erstellen, die die Kapazität der Gruppe um zwei Instanzen verringert, wählen Sie`Remove`, geben Sie 2 in das nächste Feld ein, und wählen Sie dann. `capacity units` Standardmäßig ist die Obergrenze dieser Schrittanpassung der Alarmschwellenwert, und die Untergrenze ist negative (-) Unendlichkeit.

8. Um einen weiteren Schritt hinzuzufügen, wählen Sie Add step (Schritt hinzufügen) und definieren dann den Betrag, um den skaliert werden soll, sowie die untere und obere Grenze des Schritts relativ zum Alarmschwellenwert.
9. Wählen Sie Erstellen.

AWS CLI

Um eine Richtlinie zur schrittweisen Skalierung für die Skalierung (Kapazität verringern) zu erstellen, können Sie die folgenden Beispielbefehle verwenden. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Wenn Sie die verwenden AWS CLI, erstellen Sie zunächst eine Richtlinie zur schrittweisen Skalierung, die Amazon EC2 Auto Scaling Anweisungen zur Skalierung bereitstellt, wenn der Wert einer Metrik sinkt. Anschließend erstellen Sie den Alarm, indem Sie die zu überwachende Metrik identifizieren, den unteren Schwellenwert für die Metrik und andere Details für die Alarme definieren und den Alarm der Skalierungsrichtlinie zuordnen.

Schritt 1: Erstellen Sie eine Richtlinie für die Skalierung

Verwenden Sie den folgenden Befehl put-scaling-policy, um eine schrittweise Skalierungsrichtlinie mit dem Namen`my-step-scale-in-policy`, mit dem Anpassungstyp zu erstellen, der `ChangeInCapacity` die Kapazität der Gruppe um 2 Instanzen verringert, wenn der zugehörige CloudWatch Alarm den unteren Schwellenwert der Metrik überschreitet.

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg  \
  --policy-name my-step-scale-in-policy \
  --policy-type StepScaling \
  --adjustment-type ChangeInCapacity \
  --step-adjustments MetricIntervalUpperBound=0.0,ScalingAdjustment=-2
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen ihn, um den Alarm für die CloudWatch Richtlinie zu erstellen.

```
{
    "PolicyARN": "arn:aws:autoscaling:region:123456789012:scalingPolicy:ac542982-
cbeb-4294-891c-a5a941dfa787:autoScalingGroupName/my-asg:policyName/my-step-scale-
out-policy
}
```

Schritt 2: Erstellen Sie einen CloudWatch Alarm für den unteren Schwellenwert der Metrik

Verwenden Sie den folgenden Befehl CloudWatch put-metric-alarm, um einen Alarm zu erstellen, der die Größe der Auto Scaling Scaling-Gruppe auf der Grundlage eines durchschnittlichen CPU-Schwellenwerts von 40 Prozent für mindestens zwei aufeinanderfolgende Evaluierungsperioden von zwei Minuten verringert. Geben Sie zum Verwenden einer selbst erstellten Metrik den Namen der Metrik im Feld `--metric-name` und ihren Namespace im Feld `--namespace` an.

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmLow-RemoveCapacity \
  --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average \
  --period 120 --evaluation-periods 2 --threshold 40 \
  --comparison-operator LessThanOrEqualToThreshold \
  --dimensions "Name=AutoScalingGroupName,Value=my-asg" \
  --alarm-actions PolicyARN
```

Einfache Skalierungsrichtlinien

Die folgenden Beispiele zeigen, wie Sie CLI-Befehle verwenden können, um einfache Skalierungsrichtlinien zu erstellen. Sie bleiben in diesem Dokument als Referenz für alle Kunden, die sie verwenden möchten, enthalten. Wir empfehlen jedoch, stattdessen Target-Tracking- oder Step-Scaling-Richtlinien zu verwenden.

Ähnlich wie bei Richtlinien zur schrittweisen Skalierung müssen Sie bei einfachen Skalierungsrichtlinien CloudWatch Alarme für Ihre Skalierungsrichtlinien erstellen. In den Richtlinien, die Sie erstellen, müssen Sie auch definieren, ob und wie viele Instanzen hinzugefügt oder entfernt werden sollen, oder die Gruppe auf eine exakte Größe festlegen.

Einer der Hauptunterschiede zwischen Step Scaling-Richtlinien und einfachen Skalierungsrichtlinien sind die schrittweisen Anpassungen, die Sie mit Step Scaling-Richtlinien erhalten. Bei der schrittweisen Skalierung können Sie auf der Grundlage der von Ihnen angegebenen schrittweisen Anpassungen größere oder kleinere Änderungen an der Gruppengröße vornehmen.

Eine einfache Skalierungsrichtlinie muss außerdem warten, bis eine laufende Skalierungsaktivität oder ein Ersatz für eine Integritätsprüfung abgeschlossen ist und eine Abklingzeit abgelaufen ist, bevor sie auf weitere Alarme reagiert. Im Gegensatz dazu reagiert die Richtlinie bei der schrittweisen Skalierung weiterhin auf zusätzliche Alarme, selbst wenn eine Skalierungsaktivität oder ein Ersatz für einen Gesundheitscheck im Gange ist. Das bedeutet, dass Amazon EC2 Auto Scaling alle Alarmverletzungen bewertet, sobald es die Alarmmeldungen empfängt. Aus diesem Grund empfehlen

wir, stattdessen Richtlinien zur schrittweisen Skalierung zu verwenden, auch wenn Sie nur eine einzige Skalierungsanpassung haben.

Amazon EC2 Auto Scaling hat ursprünglich nur einfache Skalierungsrichtlinien unterstützt. Wenn Sie Ihre Skalierungsrichtlinie vor der Einführung von Richtlinien zur Zielverfolgung und schrittweisen Skalierung erstellt haben, wird Ihre Richtlinie als einfache Skalierungsrichtlinie behandelt.

Erstellen Sie eine einfache Skalierungsrichtlinie für Scale-Out

Verwenden Sie den folgenden Befehl put-scaling-policy, um eine einfache Skalierungsrichtlinie mit dem Namen`my-simple-scale-out-policy`, mit dem Anpassungstyp zu erstellen, der `PercentChangeInCapacity` die Kapazität der Gruppe um 30 Prozent erhöht, wenn der zugehörige CloudWatch Alarm den oberen Schwellenwert der Metrik überschreitet.

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-out-policy \
  --auto-scaling-group-name my-asg --scaling-adjustment 30 \
  --adjustment-type PercentChangeInCapacity
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen ihn, um den Alarm für die CloudWatch Richtlinie zu erstellen.

Erstellen Sie eine einfache Skalierungsrichtlinie für die Skalierung

Verwenden Sie den folgenden Befehl put-scaling-policy, um eine einfache Skalierungsrichtlinie mit dem Namen`my-simple-scale-in-policy`, mit dem Anpassungstyp zu erstellen, der `ChangeInCapacity` die Kapazität der Gruppe um eine Instanz verringert, wenn der zugehörige CloudWatch Alarm den unteren Schwellenwert der Metrik überschreitet.

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-in-policy \
  --auto-scaling-group-name my-asg --scaling-adjustment -1 \
  --adjustment-type ChangeInCapacity --cooldown 180
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen ihn, um den Alarm für die CloudWatch Richtlinie zu erstellen.

Skalierungsruhephasen für Amazon EC2 Auto Scaling

> **Important**
>
> Als bewährte Methode empfehlen wir Ihnen, keine einfachen Skalierungsrichtlinien und Skalierungs-Cooldowns zu verwenden. Eine Skalierungsrichtlinie für die Zielverfolgung oder

eine Richtlinie für die schrittweise Skalierung ist besser für die Skalierung der Leistung. Für eine Skalierungsrichtlinie, welche die Größe Ihrer Auto-Scaling-Gruppe proportional verändert, wenn der Wert der Skalierungsmetrik ab- oder zunimmt, sollte eine Zielverfolgung statt einer einfachen oder Schrittskalierung verwendet werden.

Wenn Sie einfache Skalierungsrichtlinien für Ihre Auto-Scaling-Gruppe erstellen, empfehlen wir, dass Sie gleichzeitig die Skalierungs-Ruhephase konfigurieren.

Nachdem Ihre Auto-Scaling-Gruppe Instances gestartet oder beendet hat, wartet sie auf das Ende der Ruhephase, bevor weitere Skalierungsaktivitäten gestartet werden können, die durch einfache Skalierungsrichtlinien initiiert werden. Der Zweck der Ruhephase besteht darin, dass sich Ihre Auto-Scaling-Gruppe stabilisiert und verhindert, dass weitere Instances gestartet oder beendet werden, bevor die Auswirkungen der vorherigen Skalierungsaktivität sichtbar sind.

Angenommen, eine einfache Skalierungsrichtlinie für die CPU-Auslastung empfiehlt, zwei Instances zu starten. Amazon EC2 Auto Scaling startet zwei Instances und unterbricht dann die Skalierungen bis zum Ende der Ruhephase. Nach Ende der Ruhephase können alle Skalierungen, die durch einfache Skalierungsrichtlinien ausgelöst werden, fortgesetzt werden. Wenn die CPU-Auslastung erneut die Alarmhöhe verletzt, skaliert die Auto-Scaling-Gruppe erneut, und die Ruhephase tritt erneut in Kraft. Wenn jedoch zwei Instances ausgereicht haben, um den Metrikwert zu verringern, bleibt die aktuelle Größe der Gruppe erhalten.

Inhalt

Überlegungen

Die folgenden Überlegungen gelten bei der Arbeit mit einfachen Skalierungsrichtlinien und Skalierung von Cooldowns:

- Richtlinien zur Zielverfolgung und zur Schrittskalierung können sofort eine Aufskalierungs-Aktivität initiieren, ohne auf das Ende der Ruhephase zu warten. Stattdessen haben die einzelnen Instances immer dann, wenn Ihre Auto Scaling Scaling-Gruppe Instances startet, eine Aufwärmphase.

Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

- Wenn eine geplante Aktion während einer Ruhephase zum geplanten Zeitpunkt beginnt, kann sie umgehend eine Skalierung auslösen, ohne das Ende der Ruhephase abzuwarten.
- Wenn eine Instance fehlerhaft wird, wartet Amazon EC2 Auto Scaling nicht auf das Ende der Ruhephase, bevor die Instance ersetzt wird.
- Wenn mehrere Instances launchen oder beenden, beginnt die Ruhephase (sei es die standardmäßige oder die Skalierungsrichtlinienspezifische Ruhephase) nach erfolgtem Start oder erfolgter Beendigung der letzten Instance.
- Wenn Sie Ihre Auto-Scaling-Gruppe manuell skalieren, wird standardmäßig nicht auf das Ende eines Cooldown-Vorgangs gewartet. Sie können dieses Verhalten jedoch überschreiben und die Standard-Abklingzeit beibehalten, wenn Sie das AWS CLI oder ein SDK für die manuelle Skalierung verwenden.
- Standardmäßig wartet Elastic Load Balancing 300 Sekunden, um den Abmeldeprozess (Connection Draining) abzuschließen. Wenn sich die Gruppe hinter einer Elastic Load Balancing-Load-Balancer befindet, wartet sie, bis sich die abschließenden Instances abmelden, bevor die Ruhephase beginnt.

Lebenszyklus-Hooks können zusätzliche Verzögerungen verursachen

Wenn ein Lebenszyklus-Hook aufgerufen wird, beginnt die Ruhephase, nachdem Sie die Lebenszyklus-Aktion abgeschlossen haben oder nach Ende des Timeout-Zeitraums. Angenommen, eine Auto-Scaling-Gruppe besitzt einen Lebenszyklus-Hook für den Instance-Start. Steigt die Auslastung der Anwendung, startet die Gruppe eine Instance, um die Kapazität zu erhöhen. Da es einen Lebenszyklus-Hook gibt, wird die Instance in einen Wartezustand versetzt und Skalierungen aufgrund einfacher Skalierungsrichtlinien werden angehalten. Die Ruhephase beginnt, wenn die Instance in den Status `InService` versetzt wird. Nach Ablauf der Ruhephase werden einfache Skalierungsrichtlinien wieder aufgenommen.

Wenn Elastic Load Balancing für die Skalierung aktiviert ist, beginnt die Abklingzeit, wenn die für die Kündigung ausgewählte Instance mit dem Verbindungsabbau beginnt (Abmeldeverzögerung). Die Abklingzeit wartet nicht darauf, dass der Verbindungsabbau abgeschlossen ist oder der Lifecycle-Hook seine Aktion abgeschlossen hat. Dies bedeutet, dass alle Skalierungsaktivitäten aufgrund einfacher Skalierungsrichtlinien fortgesetzt werden können, sobald sich das Ergebnis des Abskalierungs-Ereignisses in der Kapazität der Gruppe widerspiegelt. Andernfalls erhöht das Warten

auf den Abschluss aller drei Aktivitäten (Connection-Draining, Lebenszyklus-Hook und Ruhephase) die Zeit, welche die Auto-Scaling-Gruppe benötigt, um die Skalierung zu unterbrechen.

Ändern der standardmäßigen Ruhephase

Sie können den standardmäßige Cooldown-Vorgang nicht festlegen, wenn Sie zunächst eine Auto-Scaling-Gruppe in der Scaling-Konsole von Amazon EC2 Auto Scaling erstellen. Standardmäßig ist diese Ruhephase auf 300 Sekunden (5 Minuten) festgelegt. Bei Bedarf können Sie dies aktualisieren, nachdem die Gruppe erstellt wurde.

So ändern Sie die standardmäßige Ruhephase (Konsole)

Nach dem Erstellen der Auto-Scaling-Gruppe wählen Sie auf der Registerkarte Details, Advances configurations (Erweiterte Konfigurationen) und dann Edit (Bearbeiten) aus. Für Default cooldown (Standardmäßige Ruhephase/Cooldown-Vorgang) wählen Sie die gewünschte Zeit auf der Grundlage der Startzeit für Ihre Instance oder anderer Anwendungsanforderungen aus.

Ändern der standardmäßigen Ruhephase (AWS CLI)

Verwenden Sie die folgenden Befehle, um die standardmäßige Ruhephase für neue oder vorhandene Auto-Scaling-Gruppen zu ändern. Wenn die standardmäßige Ruhephase nicht definiert ist, wird der Standardwert von 300 Sekunden verwendet.

- create-auto-scaling-group
- update-auto-scaling-group

Um den Wert der Standard-Ruhephase zu bestätigen, verwenden Sie den Befehl describe-auto-scaling-groups.

Festlegen einer Ruhephase für bestimmte einfache Skalierungsrichtlinien

Standardmäßig verwenden alle einfachen Skalierungsrichtlinien die für die Auto-Scaling-Gruppe definierte Standard-Ruhephase. Wenn Sie eine Ruhephase für bestimmte einfache Skalierungsrichtlinien angeben möchten, verwenden Sie den optionalen Parameter für die Ruhephase beim Erstellen oder Aktualisieren der Richtlinie. Wenn für eine Richtlinie eine Ruhephase angegeben wird, überschreibt sie die standardmäßige Ruhephase.

Eine skalierungsspezifische Ruhephase wird häufig mit einer Scale-In-Richtlinie verwendet. Da diese Richtlinie Instances beendet, benötigt Amazon EC2 Auto Scaling weniger Zeit, um zu ermitteln, ob weitere Instances zu beenden sind. Die Beendigung von Instances sollte viel schneller als der

Start von Instances erfolgen. Die standardmäßige Ruhephase von 300 Sekunden ist daher zu lang. In diesem Fall kann Ihnen eine skalierungsspezifische Ruhephase mit einem niedrigeren Wert für Ihre Abskalierungs-Richtlinie helfen, Kosten zu senken, da die Gruppe schneller nach unten skaliert werden kann.

Um einfache Skalierungsrichtlinien in der Konsole zu erstellen oder zu aktualisieren, wählen Sie die Auto Scaling (Automatische Skalierung) nachdem Sie die Gruppe erstellt haben. Verwenden Sie den Befehl put-scaling-policy AWS CLI, um einfache Skalierungsrichtlinien mit dem zu erstellen oder zu aktualisieren. Weitere Informationen finden Sie unter Schrittweise und einfache Skalierungsrichtlinien.

Skalierung basierend auf Amazon SQS

> **Important**
>
> Die folgenden Informationen und Schritte zeigen Ihnen, wie Sie den Amazon SQS-Warteschlangen-Backlog pro Instance anhand des `ApproximateNumberOfMessages` Queue-Attributs berechnen, bevor Sie ihn als benutzerdefinierte Metrik für veröffentlichen. CloudWatch Sie können jetzt jedoch die Kosten und den Aufwand für die Veröffentlichung Ihrer eigenen Metrik sparen, indem Sie metrische Berechnungen verwenden. Weitere Informationen finden Sie unter Erstellen einer Zielnachverfolgungs-Skalierungsrichtlinie für Amazon EC2 Auto Scaling mit Metrikberechnungen.

In diesem Abschnitt erfahren Sie, wie Sie Ihre Auto-Scaling-Gruppe als Reaktion auf Änderungen der Systemauslastung in einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange skalieren. Weitere Informationen zur Verwendung von Amazon SQS finden Sie im Amazon Simple Queue Service-Entwicklerhandbuch.

Es gibt einige Szenarien, in denen Sie als Reaktion auf Aktivitäten in einer Amazon SQS-Warteschlange eine Skalierung erwägen könnten. Angenommen, Sie haben eine Webanwendung, mit der Benutzer Bilder hochladen und online verwenden können. In diesem Szenario erfordert jedes Image eine Größenanpassung und Codierung, bevor es veröffentlicht werden kann. Die App wird auf EC2-Instances einer Auto-Scaling-Gruppe ausgeführt und ist für die Verarbeitung Ihrer typischen Upload-Raten konfiguriert. Fehlerhafte Instances werden beendet und ersetzt, um stets dieselbe Anzahl an Instances zu nutzen. Die App platziert die Raw-Bitmap-Daten der Bilder in eine SQS-Warteschlange für die Verarbeitung. Sie verarbeitet die Bilder und veröffentlicht die verarbeiteten Bilder, wo sie von Benutzern angezeigt werden können. Die Architektur für dieses Szenario funktioniert gut, wenn die Anzahl der Image-Uploads nicht im Laufe der Zeit variiert. Wenn

sich die Anzahl der Uploads jedoch mit der Zeit ändert, könnten Sie eine dynamische Skalierung in Betracht ziehen, um die Kapazität Ihrer Auto-Scaling-Gruppe zu skalieren.

Inhalt

Zielverfolgung mit der richtigen Metrik verwenden

Wenn Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung verwenden, die auf einer benutzerdefinierten Amazon SQS-Warteschlangenmetrik basiert, kann die dynamische Skalierung eine effektivere Anpassung an die Bedarfskurve Ihrer Anwendung vornehmen. Weitere Informationen zum Auswählen von Metriken für die Zielverfolgung finden Sie unter Auswahl von Metriken.

Das Problem bei der Verwendung einer CloudWatch Amazon SQS-Metrik wie `ApproximateNumberOfMessagesVisible` für die Zielverfolgung besteht darin, dass sich die Anzahl der Nachrichten in der Warteschlange möglicherweise nicht proportional zur Größe der Auto Scaling Scaling-Gruppe ändert, die Nachrichten aus der Warteschlange verarbeitet. Das liegt daran, dass die Anzahl der Nachrichten in der SQS-Warteschlange nicht alleine die Anzahl der erforderlichen Instances definiert. Tatsächlich kann die Anzahl der Instances in Ihrer Auto-Scaling-Gruppe von mehreren Faktoren abhängen, einschließlich der für die Verarbeitung einer Nachricht benötigten Zeit und der akzeptablen zeitlichen Latenz (Warteschlangenverzögerung).

Die Lösung besteht darin, eine Rückstand pro Instance-Metrik zu verwenden, deren Zielwert der zulässige Rückstand pro Instance ist, der aufrechterhalten werden soll. Sie können diese Zahlen wie folgt berechnen:

- Rückstand pro Instance: Um den Rückstand pro Instance zu berechnen, verwenden Sie zunächst das Warteschlangenattribut `ApproximateNumberOfMessages`, um die Länge der SQS-Warteschlange (Anzahl der Nachrichten, die zum Abrufen aus der Warteschlange verfügbar sind) zu bestimmen. Dividieren Sie diese Zahl durch die laufende Kapazität der Flotte, welche bei einer Auto-Scaling-Gruppe die Anzahl der Instances mit dem Status `InService` ist, um so den Rückstand pro Instance zu erhalten.
- Zulässiger Rückstand pro Instance: Um Ihren Zielwert zu berechnen, bestimmen Sie zunächst, welche zeitliche Latenz Ihre Anwendung akzeptieren kann. Teilen Sie diesen akzeptablen

Latenzwert dann durch die durchschnittliche Zeit, die eine EC2-Instance für die Verarbeitung einer Nachricht benötigt.

Ein Beispiel: Angenommen, Sie verfügen über eine Auto-Scaling-Gruppe mit 10 Instances und die Anzahl sichtbarer Nachrichten in der Warteschlange (`ApproximateNumberOfMessages`) ist 1 500. Wenn die durchschnittliche Verarbeitungszeit pro Nachricht 0,1 Sekunden beträgt und die längste akzeptable Latenz 10 Sekunden ist, dann ist der zulässige Rückstand pro Instance 10/0,1, was 100 Nachrichten entspricht. Dies bedeutet, dass 100 der Zielwert für Ihre Ziel-Nachverfolgungsrichtlinie ist. Wenn der Rückstand pro Instance den Zielwert erreicht, wird ein Aufskalierungsereignis ausgelöst. Da der Rückstand pro Instance bereits bei 150 Nachrichten liegt (1 500 Nachrichten / 10 Instances), wird Ihre Gruppe um fünf Instances aufskaliert, um die Proportion zum Zielwert beizubehalten.

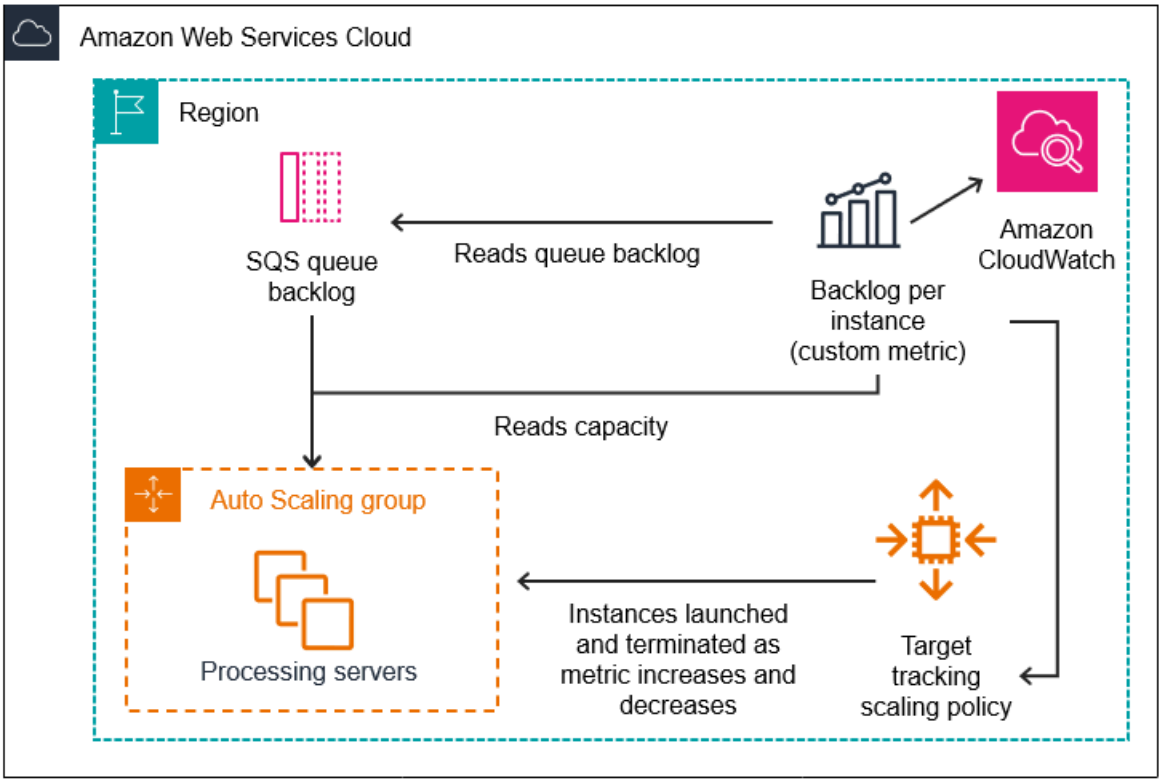
Die folgenden Verfahren veranschaulichen, wie Sie die benutzerdefinierte Metrik veröffentlichen und die Skalierungsrichtlinie für die Ziel-Nachverfolgung erstellen, die Ihre Auto-Scaling-Gruppe zum Skalieren basierend auf diesen Berechnungen konfiguriert.

> **Important**
>
> Denken Sie daran, zur Kostensenkung stattdessen metrische Berechnungen zu verwenden. Weitere Informationen finden Sie unter Erstellen einer Zielnachverfolgungs-Skalierungsrichtlinie für Amazon EC2 Auto Scaling mit Metrikberechnungen.

Es gibt drei Hauptkomponenten dieser Konfiguration:

- Eine Auto-Scaling-Gruppe zur Verwaltung von EC2-Instances für die Verarbeitung von Nachrichten aus einer SQS-Warteschlange.
- Eine benutzerdefinierte Metrik zum Senden an Amazon CloudWatch , die die Anzahl der Nachrichten in der Warteschlange pro EC2-Instance in der Auto Scaling Scaling-Gruppe misst.
- Eine Zielverfolgungsrichtlinie, die Ihre Auto Scaling Scaling-Gruppe so konfiguriert, dass sie auf der Grundlage der benutzerdefinierten Metrik und eines festgelegten Zielwerts skaliert. CloudWatch Alarme rufen die Skalierungsrichtlinie auf.

Das folgende Diagramm illustriert die Architektur dieser Konfiguration.



Einschränkungen und Voraussetzungen

Um diese Konfiguration verwenden zu können, müssen Sie die folgenden Einschränkungen beachten:

- Sie müssen das AWS CLI oder ein SDK verwenden, um Ihre benutzerdefinierte Metrik zu CloudWatch veröffentlichen. Anschließend können Sie Ihre Metrik mit dem überwachen AWS Management Console.
- Die Amazon EC2 Auto Scaling-Konsole unterstützt Skalierungsrichtlinien zur Ziel-Nachverfolgung, die benutzerdefinierte Metriken verwenden. Sie müssen das AWS CLI oder ein SDK verwenden, um eine benutzerdefinierte Metrik für Ihre Skalierungsrichtlinie anzugeben.

In den folgenden Abschnitten erfahren Sie, wie Sie das AWS CLI für die Aufgaben verwenden, die Sie ausführen müssen. Um beispielsweise Metrikdaten abzurufen, welche die gegenwärtige Verwendung der Warteschlange widerspiegeln, verwenden Sie den get-queue-attribute-SQS-Befehl. Stellen Sie sicher, dass die CLI installiert und konfiguriert ist.

Bevor Sie beginnen, müssen Sie über eine Amazon SQS-Warteschlange verfügen. In den folgenden Abschnitten wird davon ausgegangen, dass Sie bereits über eine Warteschlange

(Standard oder FIFO), eine Auto-Scaling-Gruppe und EC2-Instances mit der Anwendung verfügen, welche die Warteschlange verwendet. Weitere Informationen zu Amazon SQS, finden Sie unter Entwicklerhandbuch für Amazon Simple Queue Service.

Konfigurieren der Skalierung basierend auf Amazon SQS

Aufgaben

- Schritt 1: Erstellen Sie eine CloudWatch benutzerdefinierte Metrik
- Schritt 2: Erstellen einer Skalierungsrichtlinie für die Ziel-Nachverfolgung
- Schritt 3: Testen Ihrer Skalierungsrichtlinie

Schritt 1: Erstellen Sie eine CloudWatch benutzerdefinierte Metrik

Eine benutzerdefinierte Metrik wird mit einem Metriknamen und Namespace Ihrer Wahl definiert. Namespaces für benutzerdefinierte Metriken können nicht mit `AWS/` beginnen. Weitere Informationen zum Veröffentlichen von benutzerdefinierten Metriken finden Sie unter dem Thema Veröffentlichen von benutzerdefinierten Kennzahlen im CloudWatch Amazon-Benutzerhandbuch.

Gehen Sie wie folgt vor, um die benutzerdefinierte Metrik zu erstellen, indem Sie zunächst die Informationen aus Ihrem AWS Konto lesen. Berechnen Sie dann den Rückstand pro Instance-Metrik, wie in einem früheren Abschnitt empfohlen. Veröffentlichen Sie diese Zahl abschließend mit CloudWatch einer Genauigkeit von 1 Minute. Wenn möglich, empfehlen wir dringend, anhand von Metriken mit einer Granularität von 1 Minute zu skalieren, um eine schnellere Reaktion auf Änderungen der Systemauslastung zu gewährleisten.

Um eine CloudWatch benutzerdefinierte Metrik zu erstellen ()AWS CLI

1. Fordern Sie mit dem SQS-Befehl get-queue-attributes die Anzahl der in der Warteschlange wartenden Nahrichten an (`ApproximateNumberOfMessages`).

   ```
   aws sqs get-queue-attributes --queue-url https://
   sqs.region.amazonaws.com/123456789/MyQueue \
     --attribute-names ApproximateNumberOfMessages
   ```

2. Verwenden Sie den Befehl describe-auto-scaling-groups, um die laufende Kapazität der Gruppe anzufordern, wobei es sich um die Anzahl von Instances mit dem Status `InService` handelt. Dieser Befehl gibt die Instances einer Auto-Scaling-Gruppe zusammen mit ihren Lebenszyklusstatus zurück.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg
```

3. Berechnen Sie den Rückstand pro Instance, indem Sie die ungefähre Anzahl der Nachrichten, die für den Abruf aus der Warteschlange verfügbar sind, durch die laufende Kapazität der Gruppe dividieren.
4. Erstellen Sie ein Skript, das jede Minute ausgeführt wird, um den Backlog-Wert pro Instanz abzurufen und ihn in einer CloudWatch benutzerdefinierten Metrik zu veröffentlichen. Beim Veröffentlichen einer benutzerdefinierten Metrik geben Sie den Namen, den Namespace, die Einheit, den Wert und null oder mehr Dimensionen für die Metrik an. Dimensionen bestehen aus einem Dimensionsnamen und einem Dimensionswert.

   Um Ihre benutzerdefinierte Metrik zu veröffentlichen, ersetzen Sie *kursiv* gedruckte Platzhalterwerte durch Ihren bevorzugten Metriknamen, den Wert der Metrik, einen Namespace (solange er nicht mit „AWS“ beginnt) und Dimensionen (optional) und führen Sie dann den folgenden put-metric-data-Befehl aus.

   ```
   aws cloudwatch put-metric-data --metric-name MyBacklogPerInstance --
   namespace MyNamespace \
     --unit None --value 20 --
   dimensions MyOptionalMetricDimensionName=MyOptionalMetricDimensionValue
   ```

Nachdem Ihre Anwendung die gewünschte Metrik ausgegeben hat, werden die Daten an CloudWatch gesendet. Die Metrik ist in der CloudWatch Konsole sichtbar. Sie können darauf zugreifen, indem Sie sich bei der anmelden AWS Management Console und zu der CloudWatch Seite navigieren. Anschließend können Sie die Metrik anzeigen, indem Sie zur Seite der Metriken navigieren oder im Suchfeld danach suchen. Informationen zum Anzeigen von Metriken finden Sie unter Verfügbare Metriken anzeigen im CloudWatch Amazon-Benutzerhandbuch.

Schritt 2: Erstellen einer Skalierungsrichtlinie für die Ziel-Nachverfolgung

Die von Ihnen erstellte Metrik kann jetzt zu einer Skalierungsrichtlinie für die Zielnachverfolgung hinzugefügt werden.

So erstellen Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung (AWS CLI)

1. Verwenden Sie den folgenden `cat`-Befehl, um einen Zielwert für Ihre Skalierungsrichtlinie und eine benutzerdefinierte Metrikspezifikation in einer JSON-Datei namens `config.json` in Ihrem Stammverzeichnis zu speichern. Ersetzen Sie jedes *Platzhalter für Benutzereingaben*

durch Ihre eigenen Informationen. Berechnen Sie für den `TargetValue` die Metrik für den akzeptablen Rückstand pro Instance und geben Sie sie hier ein. Basieren Sie die Berechnung dieser Zahl auf einem normalen Latenzwert und teilen Sie ihn durch die durchschnittliche Zeit, die für die Verarbeitung einer Nachricht benötigt wird, wie in einem vorherigen Abschnitt beschrieben.

Wenn Sie keine Dimensionen für die Metrik angegeben haben, die Sie in Schritt 1 erstellt haben, nehmen Sie keine Dimensionen in die benutzerdefinierte Metrikspezifikation auf.

```
$ cat ~/config.json
{
   "TargetValue":100,
   "CustomizedMetricSpecification":{
      "MetricName":"MyBacklogPerInstance",
      "Namespace":"MyNamespace",
      "Dimensions":[
         {
            "Name":"MyOptionalMetricDimensionName",
            "Value":"MyOptionalMetricDimensionValue"
         }
      ],
      "Statistic":"Average",
      "Unit":"None"
   }
}
```

2. Verwenden Sie den Befehl put-scaling-policy zusammen mit der Datei `config.json`, die Sie im vorherigen Schritt erstellt haben, um Ihre Skalierungsrichtlinie zu erstellen.

```
aws autoscaling put-scaling-policy --policy-name sqs100-target-tracking-scaling-
policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://~/config.json
```

Dabei werden zwei Alarme erstellt: einer für die Aufwärtsskalierung und einer für die Abwärtsskalierung. Es gibt auch den Amazon-Ressourcennamen (ARN) der Richtlinie zurück CloudWatch, mit der die CloudWatch Skalierung aufgerufen wird, wenn der metrische Schwellenwert überschritten wird.

Schritt 3: Testen Ihrer Skalierungsrichtlinie

Wenn die Einrichtung abgeschlossen ist, überprüfen Sie, ob Ihre Skalierungsrichtlinie funktioniert. Sie testen sie, indem Sie die Anzahl der Nachrichten in der SQS-Warteschlange vergrößern und dann prüfen, ob die Auto-Scaling-Gruppe eine weitere EC2-Instance gestartet hat. Gleichermaßen testen Sie sie, indem Sie die Anzahl der Nachrichten in der SQS-Warteschlange verkleinern und dann prüfen, ob die Auto-Scaling-Gruppe eine EC2-Instance beendet hat.

So testen Sie die Funktion für die horizontale Skalierung nach oben

1. Folgen Sie den Schritten unter Erstellen einer Amazon SQS SQS-Standardwarteschlange und Senden einer Nachricht oder Erstellen einer Amazon SQS SQS-FIFO-Warteschlange und Senden einer Nachricht, um Nachrichten zu Ihrer Warteschlange hinzuzufügen. Stellen Sie sicher, dass Sie die Anzahl der Nachrichten in der Warteschlange so erhöht haben, dass die Metrik für den Rückstand pro Instance den Zielwert überschreitet.

   Es kann einige Minuten dauern, bis Ihre Änderungen den Alarm auslösen.
2. Überprüfen Sie mit dem Befehl describe-auto-scaling-groups, ob die Gruppe eine Instance gestartet hat:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

So testen Sie die Funktion für die horizontale Skalierung nach oben

1. Folgen Sie den Schritten unter Nachricht empfangen und löschen (Konsole), um Nachrichten aus der Warteschlange zu löschen. Stellen Sie sicher, dass Sie die Anzahl der Nachrichten in der Warteschlange so verringert haben, dass die Metrik für den Rückstand pro Instance den Zielwert unterschreitet.

   Es kann einige Minuten dauern, bis Ihre Änderungen den Alarm auslösen.
2. Überprüfen Sie mit dem Befehl describe-auto-scaling-groups, ob die Gruppe eine Instance terminiert hat:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Amazon SQS und Instance-Skaliterungsschutz

Sofern beim Beenden einer Instance noch nicht verarbeitete Nachrichten vorhanden sind, werden diese an die SQS-Warteschlange zurückgegeben und können von einer anderen ausgeführten Instance verarbeitet werden. Für Anwendungen, in denen Aufgaben mit langer Ausführung ausgeführt werden, können Sie optional den Instance-Scale-in-Schutz verwenden, um die Kontrolle darüber zu haben, welche Warteschlangen-Worker beendet werden, wenn Ihre Auto-Scaling-Gruppe skaliert wird.

Der folgende Pseudocode zeigt eine Möglichkeit zum Schutz langjähriger, warteschlangengesteuerter Worker-Prozesse vor Scale-In-Beendigung.

```
while (true)
{
  SetInstanceProtection(False);
  Work = GetNextWorkUnit();
  SetInstanceProtection(True);
  ProcessWorkUnit(Work);
  SetInstanceProtection(False);
}
```

Weitere Informationen finden Sie unter Entwerfen Sie Ihre Anwendungen auf Amazon EC2 Auto Scaling, um die Instance-Beendigung ordnungsgemäß zu handhaben.

Eine Skalierung für eine Auto-Scaling-Gruppe überprüfen

Im Abschnitt Amazon EC2 Auto Scaling der Amazon-EC2-Konsole können Sie über Activity history (Aktivitätsverlauf) für eine Auto-Scaling-Gruppe den aktuellen Status einer laufenden Skalierungsaktivität einsehen. Wenn die Skalierungsaktivität abgeschlossen ist, können Sie sehen, ob sie erfolgreich war oder nicht. Dies ist besonders nützlich, wenn Sie Auto-Scaling-Gruppen erstellen oder Skalierungsbedingungen zu bestehenden Gruppen hinzufügen.

Wenn Sie eine Zielverfolgungs-, Stufen- oder einfache Skalierungsrichtlinie zu Ihrer Auto-Scaling-Gruppe hinzufügen, beginnt Amazon EC2 Auto Scaling sofort mit der Auswertung der Richtlinie anhand der Metrik. Der Metrik-Alarm geht in den ALARM-Zustand, wenn die Metrik für eine bestimmte Anzahl von Auswertungszeiträumen den Schwellenwert überschreitet. Das bedeutet, dass eine Skalierungsrichtlinie kurz nach ihrer Erstellung zu einer Skalierungsaktivität führen kann. Nachdem Amazon EC2 Auto Scaling die gewünschte Kapazität als Reaktion auf eine Skalierungsrichtlinie angepasst hat, können Sie die Skalierungsaktivität in Ihrem Konto überprüfen.

Wenn Sie eine E-Mail-Benachrichtigung von Amazon EC2 Auto Scaling erhalten möchten, die Sie über eine Skalierungsaktivität informiert, folgen Sie den Anweisungen in Amazon SNS-Benachrichtigungsoptionen für Amazon EC2 Auto Scaling.

> **Tip**
>
> Im folgenden Verfahren sehen Sie sich die Abschnitte Activity history (Verlauf der Aktivität) und Instances (Instances) für die Auto- Scaling-Gruppe an. In beiden sollten die benannten Spalten bereits angezeigt werden. Um ausgeblendete Spalten anzuzeigen oder die Anzahl der angezeigten Zeilen zu ändern, wählen Sie das Zahnradsymbol in der oberen rechten Ecke jedes Abschnitts, um die Einstellungen zu öffnen, die Einstellungen nach Bedarf zu aktualisieren und Confirm (Bestätigen) auszuwählen.

So zeigen Sie die Skalierungsaktivitäten für eine Auto-Scaling-Gruppe an (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie in der Navigationsleiste oben die Region aus, in der sich Ihre Auto-Scaling-Gruppe befindet.
3. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
4. Auf der Registerkarte Activity (Aktivität) unter Activity history (Aktivitätsverlauf) zeigt die Spalte Status an, ob Ihre Auto-Scaling-Gruppe-Instances erfolgreich gestartet oder beendet hat oder ob die Skalierungsaktivität noch im Gange ist.
5. Wenn Sie viele Skalierungen haben, können Sie auf das Symbol **>** am oberen Rand des Aktivitätsverlaufs klicken, um die nächste Seite der Skalierungen anzuzeigen.
6. Auf der Registerkarte Instance management (Instance-Verwaltung) wird unter Instances in der Spalte Lifecycle (Lebenszyklus) der Zustand Ihrer Instances angezeigt. Nach dem Start der Instance und dem Ende der Lebenszyklus-Hooks ändert sich ihr Lebenszyklusstatus in `InService`. In der Spalte Health status (Zustandsstatus) wird Ihnen das Ergebnis der Zustandsprüfung Ihrer EC2-Instance angezeigt.

So zeigen Sie die Skalierungsaktivitäten für eine Auto-Scaling-Gruppe an (AWS CLI)

Verwenden Sie den folgenden describe-scaling-activities-Befehl.

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

Es folgt eine Beispielausgabe.

Skalierungsaktivitäten werden nach Startzeit sortiert. Die noch laufenden Aktivitäten werden zuerst beschrieben.

```
{
  "Activities": [
    {
      "ActivityId": "5e3a1f47-2309-415c-bfd8-35aa06300799",
      "AutoScalingGroupName": "my-asg",
      "Description": "Terminating EC2 instance: i-06c4794c2499af1df",
      "Cause": "At 2020-02-11T18:34:10Z a monitor alarm TargetTracking-my-asg-AlarmLow-
b9376cab-18a7-4385-920c-dfa3f7783f82 in state ALARM triggered policy my-target-
tracking-policy changing the desired capacity from 3 to 2.  At 2020-02-11T18:34:31Z
 an instance was taken out of service in response to a difference between desired and
 actual capacity, shrinking the capacity from 3 to 2.  At 2020-02-11T18:34:31Z instance
 i-06c4794c2499af1df was selected for termination.",
      "StartTime": "2020-02-11T18:34:31.268Z",
      "EndTime": "2020-02-11T18:34:53Z",
      "StatusCode": "Successful",
      "Progress": 100,
      "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-west-2a
\"...}",
      "AutoScalingGroupARN": "arn"
    },
...
  ]
}
```

Eine Beschreibung der Felder in der Ausgabe finden Sie unterAktivität in der Amazon EC2 Auto Scaling API-Referenz.

Hilfe beim Abrufen der Skalierungsaktivitäten für eine gelöschte Gruppe und Informationen über die Arten von Fehlern, die auftreten können, und deren Behandlung finden Sie unter Fehlersuche bei Amazon EC2 Auto Scaling.

Eine Skalierungsrichtlinie für eine Auto-Scaling-Gruppe deaktivieren

In diesem Thema wird beschrieben, wie eine Skalierungsrichtlinie vorübergehend deaktiviert wird, damit keine Änderungen an der Anzahl der Instances in der Auto-Scaling-Gruppe ausgelöst werden.

Wenn Sie eine Skalierungsrichtlinie deaktivieren, bleiben die Konfigurationsdetails erhalten, so dass Sie die Richtlinie schnell wieder aktivieren können. Dies ist einfacher, als eine Richtlinie vorübergehend zu löschen, wenn Sie sie nicht benötigen, und später neu zu erstellen.

Wenn eine Skalierungsrichtlinie deaktiviert ist, skaliert die Auto-Scaling-Gruppe nicht für die Metrikalarme, die verletzt werden, während die Skalierungsrichtlinie deaktiviert ist. Alle noch laufenden Skalierungsaktivitäten werden jedoch nicht angehalten.

Beachten Sie, dass deaktivierte Skalierungsrichtlinien weiterhin für Ihre Kontingente für die Anzahl der Skalierungsrichtlinien zählen, die Sie einer Auto-Scaling-Gruppe hinzufügen können.

So deaktivieren Sie eine Skalierungsrichtlinie (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Aktivieren Sie auf der Registerkarte Automatische Skalierung unter Dynamische Skalierungsrichtlinien das Kontrollkästchen in der oberen rechten Ecke der gewünschten Skalierungsrichtlinie.
4. Scrollen Sie zum oberen Rand des Abschnitts Dynamische Skalierungsrichtlinien und wählen Sie Aktionen, Deaktivieren.

Wenn Sie bereit sind, die Skalierungsrichtlinie erneut zu aktivieren, wiederholen Sie diese Schritte, und wählen Sie dann Actions (Aktionen), Enable (Aktivieren). Nachdem Sie eine Skalierungsrichtlinie erneut aktiviert haben, kann Ihre Auto-Scaling-Gruppe sofort eine Skalierungsaktion initiieren, wenn derzeit Alarme im ALARM-Status vorhanden sind.

So deaktivieren Sie eine Skalierungsrichtlinie (AWS CLI):

Verwenden Sie den put-scaling-policy-Befehl mit der `--no-enabled`-Option wie folgt. Geben Sie alle Optionen in dem Befehl so an, wie Sie sie beim Erstellen der Richtlinie angeben würden.

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \
   --policy-name my-scaling-policy --policy-type TargetTrackingScaling \
   --estimated-instance-warmup 360 \
```

```
  --target-tracking-configuration '{ "TargetValue": 70,
 "PredefinedMetricSpecification": { "PredefinedMetricType":
 "ASGAverageCPUUtilization" } }' \
  --no-enabled
```

So aktivieren Sie eine Skalierungsrichtlinie erneut (AWS CLI):

Verwenden Sie den put-scaling-policy-Befehl mit der `--enabled`-Option wie folgt. Geben Sie alle Optionen in dem Befehl so an, wie Sie sie beim Erstellen der Richtlinie angeben würden.

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \
  --policy-name my-scaling-policy --policy-type TargetTrackingScaling \
  --estimated-instance-warmup 360 \
  --target-tracking-configuration '{ "TargetValue": 70,
 "PredefinedMetricSpecification": { "PredefinedMetricType":
 "ASGAverageCPUUtilization" } }' \
  --enabled
```

So beschreiben Sie eine Skalierungsrichtlinie (AWS CLI):

Verwenden Sie den describe-policies-Befehl, um den aktivierten Status einer Skalierungsrichtlinie zu überprüfen.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg \
  --policy-names my-scaling-policy
```

Es folgt eine Beispielausgabe.

```
{
    "ScalingPolicies": [
        {
            "AutoScalingGroupName": "my-asg",
            "PolicyName": "my-scaling-policy",
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:1d52783a-b03b-4710-
bb0e-549fd64378cc:autoScalingGroupName/my-asg:policyName/my-scaling-policy",
            "PolicyType": "TargetTrackingScaling",
            "StepAdjustments": [],
            "Alarms": [
                {
                    "AlarmName": "TargetTracking-my-asg-
AlarmHigh-9ca53fdd-7cf5-4223-938a-ae1199204502",
```

```
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-9ca53fdd-7cf5-4223-938a-
ae1199204502"
                },
                {
                    "AlarmName": "TargetTracking-my-asg-AlarmLow-7010c83d-d55a-4a7a-
abe0-1cf8b9de6d6c",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-7010c83d-d55a-4a7a-
abe0-1cf8b9de6d6c"
                }
            ],
            "TargetTrackingConfiguration": {
                "PredefinedMetricSpecification": {
                    "PredefinedMetricType": "ASGAverageCPUUtilization"
                },
                "TargetValue": 70.0,
                "DisableScaleIn": false
            },
            "Enabled": true
        }
    ]
}
```

Löschen einer Skalierungsrichtlinie

Wenn Sie für die Skalierung keine Richtlinie mehr benötigen, können Sie diese löschen. Je nach Art der Skalierungsrichtlinie müssen Sie möglicherweise auch die CloudWatch Alarme löschen. Durch das Löschen einer Skalierungsrichtlinie für die Zielverfolgung werden auch alle zugehörigen CloudWatch Alarme gelöscht. Durch das Löschen einer schrittweisen Skalierungsrichtlinie oder einer einfachen Skalierungsrichtlinie wird die zugrunde liegende Alarmaktion gelöscht, der CloudWatch Alarm wird jedoch nicht gelöscht, auch wenn ihm keine Aktion mehr zugeordnet ist.

Löschen einer Skalierungsrichtlinie (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.

3. Aktivieren Sie auf der Registerkarte Automatische Skalierung unter Dynamische Skalierungsrichtlinien das Kontrollkästchen in der oberen rechten Ecke der gewünschten Skalierungsrichtlinie.
4. Scrollen Sie zum oberen Rand des Abschnitts Dynamische Skalierungsrichtlinien und wählen Sie Aktionen, Löschen.
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Ja, löschen.
6. (Optional) Wenn Sie eine Richtlinie zur schrittweisen Skalierung oder eine einfache Skalierungsrichtlinie gelöscht haben, gehen Sie wie folgt vor, um den CloudWatch Alarm zu löschen, der mit der Richtlinie verknüpft war. Sie können diese Teilschritte überspringen, um den Alarm für die zukünftige Verwendung beizubehalten.

   a. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
   b. Wählen Sie im Navigationsbereich Alarms (Alarme) aus.
   c. Wählen Sie zunächst den Alarm (z. B. `Step-Scaling-AlarmHigh-AddCapacity`) und anschließend Action (Aktion), Delete (Löschen) aus.
   d. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).

So rufen Sie die Skalierungsrichtlinien für eine Auto-Scaling-Gruppe ab (AWS CLI)

Bevor Sie eine Skalierungsrichtlinie löschen, verwenden Sie den Befehl describe-policies, um die Skalierungsrichtlinien zu ermitteln, die für die Auto-Scaling-Gruppe erstellt wurden. Sie können die Ausgabe verwenden, wenn Sie die Richtlinie und die CloudWatch Alarme löschen.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg
```

Sie können die Ergebnisse mithilfe des Parameters `--query` nach dem Typ der Skalierungsrichtlinie filtern. Diese Syntax für `query` funktioniert unter Linux oder macOS. Unter Windows ändern Sie die einfachen Anführungszeichen in doppelte Anführungszeichen.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg
  --query 'ScalingPolicies[?PolicyType==`TargetTrackingScaling`]'
```

Es folgt eine Beispielausgabe.

```
[
    {
```

```
        "AutoScalingGroupName": "my-asg",
        "PolicyName": "cpu50-target-tracking-scaling-policy",
        "PolicyARN": "PolicyARN",
        "PolicyType": "TargetTrackingScaling",
        "StepAdjustments": [],
        "Alarms": [
            {
                "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e",
                "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
            },
            {
                "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2",
                "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
            }
        ],
        "TargetTrackingConfiguration": {
            "PredefinedMetricSpecification": {
                "PredefinedMetricType": "ASGAverageCPUUtilization"
            },
            "TargetValue": 50.0,
            "DisableScaleIn": false
        },
        "Enabled": true
    }
]
```

So löschen Sie Ihre Skalierungsrichtlinie (AWS CLI)

Verwenden Sie den folgenden Befehl delete-policy.

```
aws autoscaling delete-policy --auto-scaling-group-name my-asg \
  --policy-name cpu50-target-tracking-scaling-policy
```

Um deinen CloudWatch Alarm zu löschen (AWS CLI)

Verwenden Sie für schrittweise und einfache Skalierungsrichtlinien den Befehl delete-alarms, um den CloudWatch Alarm zu löschen, der der Richtlinie zugeordnet war. Sie können diesen Schritt

überspringen, um den Alarm für die spätere Verwendung beizubehalten. Sie können einen oder mehrere Alarme gleichzeitig löschen. Sie können beispielsweise den folgenden Befehl verwenden, um die Alarme `Step-Scaling-AlarmHigh-AddCapacity` und `Step-Scaling-AlarmLow-RemoveCapacity` zu löschen.

```
aws cloudwatch delete-alarms --alarm-name Step-Scaling-AlarmHigh-AddCapacity Step-Scaling-AlarmLow-RemoveCapacity
```

Beispiel für Skalierungsrichtlinien für die AWS Command Line Interface (AWS CLI)

Sie können Skalierungsrichtlinien für Amazon EC2 Auto Scaling über die AWS Management Console AWS CLI, oder SDKs erstellen.

Die folgenden Beispiele zeigen, wie Sie Skalierungsrichtlinien für Amazon EC2 Auto Scaling mit dem Befehl AWS CLI put-scaling-policy erstellen können. Ersetzen Sie jedes *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Informationen zu den ersten Schritten beim Schreiben von Skalierungsrichtlinien mithilfe von finden Sie in den AWS CLI einführenden Übungen unter und. Skalierungsrichtlinien für die Ziel-Nachverfolgung Schrittweise und einfache Skalierungsrichtlinien

Beispiel 1: So wenden Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung mit einer vordefinierten Metrikspezifikation an

```
aws autoscaling put-scaling-policy --policy-name cpu50-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
{
  "TargetValue": 50.0,
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ASGAverageCPUUtilization"
  }
}
```

Weitere Informationen finden Sie unter PredefinedMetricSpezifikation in der Amazon EC2 Auto Scaling API-Referenz.

> **Note**
>
> Wenn sich die Datei nicht im aktuellen Verzeichnis befindet, geben Sie den vollständigen Dateipfad ein. Weitere Informationen zum Lesen von AWS CLI Parameterwerten aus einer Datei finden Sie im AWS Command Line Interface Benutzerhandbuch unter Laden von AWS CLI Parametern aus einer Datei.

Beispiel 2: So wenden Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung mit einer benutzerdefinierten Metrikspezifikation an

```
aws autoscaling put-scaling-policy --policy-name sqs100-target-tracking-scaling-policy
 \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
{
  "TargetValue": 100.0,
  "CustomizedMetricSpecification": {
    "MetricName": "MyBacklogPerInstance",
    "Namespace": "MyNamespace",
    "Dimensions": [{
      "Name": "MyOptionalMetricDimensionName",
      "Value": "MyOptionalMetricDimensionValue"
    }],
    "Statistic": "Average",
    "Unit": "None"
  }
}
```

Weitere Informationen finden Sie unter CustomizedMetricSpezifikation in der Amazon EC2 Auto Scaling API-Referenz.

Beispiel 3: So wenden Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung nur für die horizontale Skalierung nach oben an

```
aws autoscaling put-scaling-policy --policy-name alb1000-target-tracking-scaling-policy
 \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
{
  "TargetValue": 1000.0,
```

```
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ALBRequestCountPerTarget",
    "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-
group/943f017f100becff"
  },
  "DisableScaleIn": true
}
```

Beispiel 4: So wenden Sie eine Schrittskalierungsrichtlinie für die horizontale Skalierung nach oben an

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg  \
  --policy-name my-step-scale-out-policy \
  --policy-type StepScaling \
  --adjustment-type PercentChangeInCapacity \
  --metric-aggregation-type Average \
  --step-adjustments
 MetricIntervalLowerBound=10.0,MetricIntervalUpperBound=20.0,ScalingAdjustment=10 \

 MetricIntervalLowerBound=20.0,MetricIntervalUpperBound=30.0,ScalingAdjustment=20 \
                     MetricIntervalLowerBound=30.0,ScalingAdjustment=30 \
  --min-adjustment-magnitude 1
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen den ARN, wenn Sie den CloudWatch Alarm erstellen.

Beispiel 5: So wenden Sie eine Schrittskalierungsrichtlinie für die horizontale Skalierung nach unten an

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg  \
  --policy-name my-step-scale-in-policy \
  --policy-type StepScaling \
  --adjustment-type ChangeInCapacity \
  --step-adjustments MetricIntervalUpperBound=0.0,ScalingAdjustment=-2
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen den ARN, wenn Sie den CloudWatch Alarm erstellen.

Beispiel 6: So wenden Sie eine einfache Skalierungsrichtlinie für die horizontale Skalierung nach oben an

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-out-policy \
  --auto-scaling-group-name my-asg --scaling-adjustment 30 \
  --adjustment-type PercentChangeInCapacity --min-adjustment-magnitude 2
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen den ARN, wenn Sie den CloudWatch Alarm erstellen.

Beispiel 7: So wenden Sie eine einfache Skalierungsrichtlinie für die horizontale Skalierung nach unten an

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-in-policy \
  --auto-scaling-group-name my-asg --scaling-adjustment -1 \
  --adjustment-type ChangeInCapacity --cooldown 180
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen den ARN, wenn Sie den CloudWatch Alarm erstellen.

Prädiktive Skalierung für Amazon EC2 Auto Scaling

Bei der prädiktiven Skalierung werden historische Lastdaten analysiert, um tägliche oder wöchentliche Muster im Verkehrsfluss zu erkennen. Es verwendet diese Informationen, um future Kapazitätsanforderungen zu prognostizieren, sodass Amazon EC2 Auto Scaling die Kapazität Ihrer Auto Scaling Scaling-Gruppe proaktiv erhöhen kann, um sie an die erwartete Auslastung anzupassen.

Die prädiktive Skalierung eignet sich gut für Situationen, in denen Sie Folgendes haben:

- Zyklischen Datenverkehr, z. B. hohe Auslastung von Ressourcen während der normalen Geschäftszeiten und niedrige Auslastung von Ressourcen am Abend und am Wochenende
- Wiederkehrende on-and-off Workload-Muster, wie z. B. Stapelverarbeitung, Tests oder regelmäßige Datenanalysen
- Anwendungen, die eine lange Zeit in Anspruch nehmen, was zu einer spürbaren Latenzauswirkung auf die Anwendungsleistung bei Aufskalierungsereignissen führt

Im Allgemeinen sollten Sie prädiktive Skalierung in Betracht ziehen, wenn Sie regelmäßige Datenverkehrszuwächse haben und Anwendungen nutzen, deren Initialisierung lange dauert. Mit der prädiktiven Skalierung können Sie im Vergleich mit nur dynamischer Skalierung (die reaktiv

ist) schneller skalieren, indem Sie die Kapazität vor der prognostizierten Last starten. Durch vorausschauende Skalierung können Sie möglicherweise auch Geld bei Ihrer EC2-Rechnung sparen, da Sie verhindern können, dass Sie zu viel Kapazität bereitstellen müssen.

Nehmen wir als Beispiel eine Anwendung, die eine hohe Auslastung während der Geschäftszeiten und eine geringe Nutzung über Nacht aufweist. Zu Beginn eines jeden Werktages kann die prädiktive Skalierung die Kapazität vor dem ersten Zustrom des Datenverkehrs erhöhen. Auf diese Weise kann Ihre Anwendung eine hohe Verfügbarkeit und Leistung aufrechterhalten, wenn sie von einer geringeren Auslastung zu einem Zeitraum mit höherer Auslastung übergeht. Sie müssen nicht warten, bis die dynamische Skalierung auf sich ändernden Datenverkehr reagiert. Sie müssen auch keine Zeit damit verbringen, die Lastmuster Ihrer Anwendung zu überprüfen und mithilfe der geplanten Skalierung die richtige Kapazität zu planen.

Themen

So funktioniert Auto Scaling

In diesem Thema wird erklärt, wie Predictive Scaling funktioniert, und es wird beschrieben, was bei der Erstellung einer Predictive Scaling-Richtlinie zu beachten ist.

Themen

Funktionsweise

Um Predictive Scaling zu verwenden, erstellen Sie eine Predictive Scaling-Richtlinie, die die zu überwachende und zu analysierende CloudWatch Metrik festlegt. Damit die prädiktive Skalierung

mit der Prognose future Werte beginnen kann, muss diese Metrik Daten für mindestens 24 Stunden enthalten.

Nachdem Sie die Richtlinie erstellt haben, beginnt die prädiktive Skalierung mit der Analyse von Metrikdaten der letzten 14 Tage, um Muster zu identifizieren. Anhand dieser Analyse wird eine stündliche Prognose des Kapazitätsbedarfs für die nächsten 48 Stunden erstellt. Die Prognose wird alle 6 Stunden anhand der neuesten CloudWatch Daten aktualisiert. Wenn neue Daten eintreffen, kann die prädiktive Skalierung die Genauigkeit zukünftiger Prognosen kontinuierlich verbessern.

Wenn Sie die prädiktive Skalierung zum ersten Mal aktivieren, wird sie nur im Prognosemodus ausgeführt. In diesem Modus werden Kapazitätsprognosen generiert, Ihre Auto Scaling-Gruppe jedoch nicht auf der Grundlage dieser Prognosen skaliert. Auf diese Weise können Sie die Genauigkeit und Eignung der Prognose bewerten. Sie können Prognosedaten mithilfe der `GetPredictiveScalingForecast` API-Operation oder der AWS Management Console anzeigen.

Nachdem Sie die Prognosedaten überprüft und beschlossen haben, mit der Skalierung auf der Grundlage dieser Daten zu beginnen, schalten Sie die Skalierungsrichtlinie in den Prognose- und Skalierungsmodus um. In diesem Modus:

- Wenn in der Prognose ein Anstieg der Auslastung erwartet wird, wird Amazon EC2 Auto Scaling die Kapazität durch Skalierung erhöhen.
- Wenn in der Prognose ein Rückgang der Auslastung erwartet wird, wird sie nicht skaliert, um Kapazität zu verringern. Wenn Sie nicht mehr benötigte Kapazität entfernen möchten, müssen Sie dynamische Skalierungsrichtlinien erstellen.

Standardmäßig skaliert Amazon EC2 Auto Scaling Ihre Auto Scaling Scaling-Gruppe zu Beginn jeder Stunde auf der Grundlage der Prognose für diese Stunde. Sie können optional eine frühere Startzeit angeben, indem Sie die `SchedulingBufferTime` Eigenschaft im `PutScalingPolicy` API-Vorgang oder die Einstellung Pre-Launch Instances in der verwenden. AWS Management Console Dies veranlasst Amazon EC2 Auto Scaling, neue Instances vor dem prognostizierten Bedarf zu starten, sodass sie Zeit haben, zu starten und für die Verarbeitung des Datenverkehrs bereit zu sein.

Um das Starten neuer Instances vor dem prognostizierten Bedarf zu unterstützen, empfehlen wir Ihnen dringend, das Standard-Instance-Warmup für Ihre Auto Scaling Scaling-Gruppe zu aktivieren. Dies gibt einen Zeitraum nach einer Scale-Out-Aktivität an, in dem Amazon EC2 Auto Scaling nicht skaliert, auch wenn dynamische Skalierungsrichtlinien darauf hinweisen, dass die Kapazität verringert werden sollte. Auf diese Weise können Sie sicherstellen, dass neu gestartete Instances ausreichend Zeit haben, um mit der Bearbeitung des erhöhten Datenverkehrs zu beginnen, bevor sie für Scale-

In-Operationen in Betracht gezogen werden. Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

Maximales Kapazitätslimit

Auto Scaling Scaling-Gruppen haben eine maximale Kapazitätseinstellung, die die maximale Anzahl von EC2-Instances begrenzt, die für die Gruppe gestartet werden können. Wenn Skalierungsrichtlinien festgelegt sind, können sie die Kapazität standardmäßig nicht über die maximale Kapazität hinaus erhöhen.

Alternativ können Sie zulassen, dass die maximale Kapazität der Gruppe automatisch erhöht wird, wenn sich die prognostizierte Kapazität der Auto Scaling-Gruppe der maximalen Kapazität der Auto Scaling Scaling-Gruppe nähert oder diese überschreitet. Um dieses Verhalten zu aktivieren, verwenden Sie die `MaxCapacityBuffer` Eigenschaften `MaxCapacityBreachBehavior` und im `PutScalingPolicy` API-Vorgang oder die Einstellung für das Verhalten „Max. Kapazität“ in AWS Management Console.

> **Warning**
>
> Seien Sie vorsichtig, wenn Sie zulassen, dass die maximale Kapazität automatisch erhöht wird. Dies kann dazu führen, dass mehr Instances gestartet werden als beabsichtigt, wenn die erhöhte maximale Kapazität nicht überwacht und verwaltet wird. Die erhöhte maximale Kapazität wird dann zur neuen normalen maximalen Kapazität für die Auto Scaling Scaling-Gruppe, bis Sie sie manuell aktualisieren. Die maximale Kapazität wird nicht automatisch wieder auf das ursprüngliche Maximum reduziert.

Überlegungen

- Bestätigen Sie, ob die prädiktive Skalierung für Ihren Workload geeignet ist. Ein Workload eignet sich gut für die prädiktive Skalierung, wenn er wiederkehrende Lastmuster aufweist, die spezifisch für den Wochentag oder die Tageszeit sind. Um dies zu überprüfen, konfigurieren Sie die Richtlinien für prädiktive Skalierung im Modus Nur Prognose und lesen dann die Empfehlungen in der Konsole. Amazon EC2 Auto Scaling bietet Empfehlungen basierend auf Beobachtungen zur potenziellen Richtlinienleistung. Bewerten Sie die Prognose und die Empfehlungen, bevor Sie Ihre Anwendung durch prädiktive Skalierung aktiv skalieren lassen.
- Für die prädiktive Skalierung werden mindestens 24 Stunden an historischen Daten benötigt, damit mit der Prognose begonnen werden kann. Prognosen sind jedoch effektiver, wenn Verlaufsdaten

für zwei volle Wochen zur Verfügung stehen. Wenn Sie Ihre Anwendung aktualisieren, indem Sie eine neue Auto-Scaling-Gruppe erstellen und die alte löschen, benötigt die neue Auto-Scaling-Gruppe 24 Stunden an historischen Lastdaten, bevor die prädiktive Skalierung wieder Prognosen generieren kann. Sie können benutzerdefinierte Metriken verwenden, um Metriken über alte und neue Auto-Scaling-Gruppen hinweg zu aggregieren. Andernfalls müssen Sie möglicherweise einige Tage auf eine genauere Prognose warten.

- Wählen Sie eine Auslastungsmetrik, die die volle Auslastung Ihrer Anwendung genau wiedergibt und den Aspekt Ihrer Anwendung darstellt, der für die Skalierung am wichtigsten ist.
- Die dynamische Skalierung mit vorausschauender Skalierung hilft Ihnen dabei, die Nachfragekurve für Ihre Anwendung genau zu verfolgen. Sie skalieren in Zeiten mit geringem Datenverkehr ein und skalieren wieder heraus, wenn der Verkehr höher als erwartet ist. Wenn mehrere Skalierungsrichtlinien aktiv sind, bestimmt jede Richtlinie die gewünschte Kapazität unabhängig, und die gewünschte Kapazität wird auf das Maximum dieser Richtlinien festgelegt. Wenn beispielsweise 10 Instances an der Zielauslastung in einer Skalierungsrichtlinie für Zielverfolgung verbleiben müssen und acht Instances in einer Richtlinie zur prädiktiven Skalierung an der Zielauslastung bleiben müssen, wird die gewünschte Kapazität der Gruppe auf zehn festgelegt. Wenn Sie mit dynamischer Skalierung noch nicht vertraut sind, empfehlen wir die Verwendung von Skalierungsrichtlinien für die Zielverfolgung. Weitere Informationen finden Sie unter Dynamische Skalierung für Amazon EC2 Auto Scaling.
- Eine zentrale Annahme der vorausschauenden Skalierung ist, dass die Auto-Scaling-Gruppe homogen ist und alle Instances von gleicher Kapazität sind. Wenn dies für Ihre Gruppe nicht zutrifft, kann die prognostizierte Kapazität ungenau sein. Seien Sie daher vorsichtig, wenn Sie Richtlinien zur vorausschauenden Skalierung für Gruppen mit gemischten Instanzen erstellen, da Instances verschiedener Typen mit ungleicher Kapazität bereitgestellt werden können. Im Folgenden finden Sie einige Beispiele, bei denen die prognostizierte Kapazität ungenau sein wird:
  - Ihre Richtlinie zur vorausschauenden Skalierung basiert auf der CPU-Auslastung, aber die Anzahl der vCPUs auf jeder Auto-Scaling-Instance variiert je nach Instance-Typen.
  - Ihre Richtlinie zur vorausschauenden Skalierung basiert auf einem Netzwerk-In- oder Netzwerkausgang, aber der Netzwerkbandbreitendurchsatz für jede Auto-Scaling-Instance variiert je nach Instance-Typen. Zum Beispiel ähneln sich die Instance-Typen M5 und M5n, der M5n-Instance-Typ liefert jedoch einen deutlich höheren Netzwerkdurchsatz.

Unterstützte Regionen

Amazon EC2 Auto Scaling unterstützt prädiktive Skalierungsrichtlinien in den folgenden Bereichen AWS-Regionen: USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Oregon), USA West (Nordkalifornien), Afrika (Kapstadt), Kanada (Zentral), EU (Frankfurt), EU (Irland), EU (London), EU (Mailand), EU (Paris), EU (Stockholm), Asien-Pazifik (Hongkong), Asien-Pazifik (Jakarta), Asien-Pazifik (Mumbai), Asien-Pazifik (Osaka), Asien-Pazifik (Tokio), Asien-Pazifik (Singapur), Asien-Pazifik (Seoul), Asien-Pazifik (Sydney), Naher Osten (Bahrain), Naher Osten (VAE), Südamerika (Sao Paulo), China (Peking), China (Ningxia), AWS GovCloud (US-Ost) und AWS GovCloud (US-West).

Erstellen Sie eine Richtlinie zur vorausschauenden Skalierung

Die folgenden Verfahren helfen Ihnen bei der Erstellung einer Richtlinie für vorausschauende Skalierung mithilfe von oder. AWS Management Console AWS CLI

Wenn die Auto-Scaling-Gruppe neu ist, muss es mindestens 24 Stunden an Daten liefern, bevor Amazon EC2 Auto Scaling eine Prognose erstellen kann.

Inhalt

Erstellen einer Richtlinie für die prädiktive Skalierung (Konsole)

Wenn Sie zum ersten Mal eine Richtlinie für vorausschauende Skalierung erstellen, empfehlen wir, die Konsole zu verwenden, um mehrere Richtlinien für die prädiktive Skalierung im Modus „Nur Prognose“ zu erstellen. Auf diese Weise können Sie die potenziellen Auswirkungen verschiedener Metriken und Zielwerte testen. Sie können mehrere Richtlinien für die prädiktive Skalierung für jede Auto-Scaling-Gruppe erstellen, aber nur eine der Richtlinien kann für die aktive Skalierung verwendet werden.

Erstellen einer Richtlinie für die prädiktive Skalierung (vordefinierte Metriken)

Führen Sie das folgende Verfahren aus, um eine Richtlinie für die prädiktive Skalierung unter Verwendung vordefinierter Metriken (CPU, Netzwerk-I/O oder Anzahl der Anfragen an den Application Load Balancer) zu erstellen. Der einfachste Weg, eine Richtlinie für die prädiktive Skalierung zu erstellen, besteht darin, vordefinierte Metriken zu verwenden. Wenn Sie stattdessen benutzerdefinierte Metriken verwenden möchten, siehe Erstellen einer Richtlinie für die prädiktive Skalierung in der Konsole (benutzerdefinierte Metriken).

So erstellen Sie eine Richtlinie für die prädiktive Skalierung

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Automatic scaling (Automatische Skalierung) unter Scaling policies (Skalierungsrichtlinien) die Option Create predictive scaling policy (Richtlinie für die prädiktive Skalierung erstellen) aus.
4. Geben Sie einen Namen für die Richtlinie ein.
5. Aktivieren Sie Skalierung basierend auf Prognose, um Amazon EC2 Auto Scaling die Berechtigung zu erteilen, sofort mit der Skalierung zu beginnen.

   Um die Richtlinie im Modus Nur Prognose zu belassen, bleibt Skalierung basierend auf Prognose deaktiviert.
6. Für Metriken wählen Sie Ihre Metriken aus der Liste der Optionen aus. Zu den Optionen gehören CPU, Netzwerkeingang, Netzwerkausgang, Anzahl der Application Load Balancer und Benutzerdefiniertes Metrikpaar.

   Wenn Sie Anzahl der Application Load Balancer pro Ziel auswählen, wählen Sie anschließend in Zielgruppe eine Zielgruppe aus. Anzahl der Application Load Balancer pro Ziel wird nur unterstützt, wenn Sie eine Application Load Balancer-Zielgruppe an Ihre Auto-Scaling-Gruppe angehängt haben.

   Wenn Sie Benutzerdefiniertes Metrikpaar auswählen, wählen Sie dann aus individuelle Metriken aus den Dropdown-Listen für Lastmetrik und Skalierungsmetrik aus.
7. Für Zielauslastung geben Sie den Zielwert ein, den Amazon EC2 Auto Scaling beibehalten werden soll. Amazon EC2 Auto Scaling skaliert Ihre Kapazität, bis die durchschnittliche Auslastung der Zielauslastung entspricht oder bis sie die von Ihnen angegebene maximale Anzahl von Instances erreicht.

Wenn Ihre Skalierungsmetrik ...	Dann ist die Zielauslastung ...
CPU	Der prozentuale CPU-Anteil, den jede Instance idealerweise verwenden sollte.
Netzwerkeingang	Die durchschnittliche Anzahl von Bytes pro Minute, die jede Instance idealerweise empfangen sollte.
Netzwerkausgang	Die durchschnittliche Anzahl von Bytes pro Minute, die jede Instance idealerweise senden sollte.
Anzahl der Application Load Balancer pro Ziel	Die durchschnittliche Anzahl von Anfragen pro Minute, die jede Instance idealerweise empfangen sollte.

8. (Optional) Für Vorabstarten von Instances wählen Sie aus, wie weit im Voraus Ihre Instances gestartet werden sollen, bevor die Prognose die Last erhöht.
9. (Optional) Für Verhalten bei max. Kapazität wählen Sie aus, ob Sie Amazon EC2 Auto Scaling höher als die maximale Kapazität der Gruppe skalieren lassen, wenn die prognostizierte Kapazität das definierte Maximum überschreitet. Wenn Sie diese Einstellung aktivieren, kann die Skalierung in Zeiten erfolgen, in denen der höchste Datenverkehr vorausgesagt wird.
10. (Optional) Für Maximale Pufferkapazität oberhalb der prognostizierten Kapazität wählen Sie die zusätzliche Kapazität aus, die Sie verwenden möchten, wenn die prognostizierte Kapazität bei der maximalen Kapazität liegt oder diese überschreitet. Der Wert wird als Prozentsatz relativ zur prognostizierten Kapazität angegeben. Beispiel: Wenn der Puffer 10 ist, bedeutet dies, dass ein Puffer von 10 Prozent vorhanden ist. Wenn daher die prognostizierte Kapazität 50 ist und die maximale Kapazität 40 ist, dann ist die effektive maximale Kapazität 55.

    Wenn die Option 0 ist, kann Amazon EC2 Auto Scaling eine Kapazität skalieren, die höher als die maximale Kapazität ist, um der prognostizierten Kapazität zu entsprechen, diese aber nicht zu überschreiten.
11. Klicken Sie auf Erstellen einer Richtlinie für die prädiktive Skalierung.

Erstellen einer Richtlinie für die prädiktive Skalierung in der Konsole (benutzerdefinierte Metriken)

Führen Sie das folgende Verfahren aus, um eine Richtlinie für die prädiktive Skalierung unter Verwendung benutzerdefiniertem Metriken zu erstellen. Benutzerdefinierte Metriken können andere

Messwerte enthalten, die von Ihnen bereitgestellt werden, CloudWatch oder Metriken, für die Sie veröffentlichen CloudWatch. Informationen zur Verwendung der CPU-, Netzwerk-I/O- oder Application Load Balancer Balancer-Anforderungsanzahl pro Ziel finden Sie unter Erstellen einer Richtlinie für die prädiktive Skalierung (vordefinierte Metriken).

Zur Erstellung einer Richtlinie für die prädiktive Skalierung unter Verwendung benutzerdefiniertem Metriken:

- Sie müssen die Rohabfragen angeben, die es Amazon EC2 Auto Scaling ermöglichen, mit den Metriken in CloudWatch zu interagieren. Weitere Informationen finden Sie unter Erweiterte prädiktive Skalierungsrichtlinienkonfigurationen mit benutzerdefinierten Metriken. Um sicherzustellen, dass Amazon EC2 Auto Scaling die Metrikdaten aus extrahieren kann CloudWatch, stellen Sie sicher, dass jede Abfrage Datenpunkte zurückgibt. Bestätigen Sie dies mithilfe der CloudWatch Konsole oder der CloudWatch GetMetricDataAPI-Operation.

  > **Note**
  >
  > Wir stellen Beispiel-JSON-Nutzlasten im JSON-Editor in der Amazon EC2 Auto Scaling Konsole zur Verfügung. Diese Beispiele geben Ihnen eine Referenz für die Schlüssel-Wert-Paare, die erforderlich sind, um andere CloudWatch Metriken hinzuzufügen, die von bereitgestellt werden AWS oder für die Sie zuvor veröffentlicht haben. CloudWatch Sie können sie als Ausgangspunkt verwenden und sie dann an Ihre Bedürfnisse anpassen.

- Wenn Sie metrische Berechnungen verwenden, müssen Sie den JSON manuell so erstellen, dass er zu Ihrem individuellen Szenario passt. Weitere Informationen finden Sie unter Metrikberechnungs-Ausdrücke verwenden. Bevor Sie metrische Berechnungen in Ihrer Richtlinie verwenden, stellen Sie sicher, dass Metrikabfragen, die auf metrischen mathematischen Ausdrücken basieren, gültig sind, und geben Sie eine einzelne Zeitreihe zurück. Bestätigen Sie dies mithilfe der CloudWatch Konsole oder der CloudWatch GetMetricDataAPI-Operation.

Wenn Sie in einer Abfrage einen Fehler machen, indem Sie falsche Daten angeben, z. B. den falschen Auto-Scaling-Gruppennamen, enthält die Prognose keine Daten. Informationen zur Behebung von Problemen mit benutzerdefinierten Metriken finden Sie unter Überlegungen und Fehlerbehebung.

So erstellen Sie eine Richtlinie für die prädiktive Skalierung

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Automatic scaling (Automatische Skalierung) unter Scaling policies (Skalierungsrichtlinien) die Option Create predictive scaling policy (Richtlinie für die prädiktive Skalierung erstellen) aus.
4. Geben Sie einen Namen für die Richtlinie ein.
5. Aktivieren Sie Skalierung basierend auf Prognose, um Amazon EC2 Auto Scaling die Berechtigung zu erteilen, sofort mit der Skalierung zu beginnen.

   Um die Richtlinie im Modus Nur Prognose zu belassen, bleibt Skalierung basierend auf Prognose deaktiviert.
6. Wählen Sie für Metriken die Option Benutzerdefiniertes Metrikpaar aus.

   a. Wählen Sie unter Metrik laden die Option Benutzerdefinierte CloudWatch Metrik aus, um eine benutzerdefinierte Metrik zu verwenden. Erstellen Sie die JSON-Nutzlast, die die Metrik der Richtlinie enthält, und fügen Sie diese in das JSON-Editorfeld ein. Ersetzen Sie dabei den Inhalt des Feldes.

   b. Wählen Sie für Skalierungsmetrik die Option Benutzerdefinierte CloudWatch Metrik aus, um eine benutzerdefinierte Metrik zu verwenden. Erstellen Sie die JSON-Nutzlast, die die Definition der Skalierungsmetrik für die Richtlinie enthält, und fügen Sie sie in das JSON-Editorfeld ein. Ersetzen Sie dabei den Inhalt des Feldes.

   c. (Optional) Um eine benutzerdefinierte Kapazitätsmetrik hinzuzufügen, aktivieren Sie das Kontrollkästchen Benutzerdefinierte Kapazitätsmetrik hinzufügen. Erstellen Sie die JSON-Nutzlast, die die Definition der Kapazitätsmetrik für die Richtlinie enthält, und fügen Sie sie in das JSON-Editorfeld ein. Ersetzen Sie dabei den Inhalt des Feldes.

      Sie müssen diese Option nur aktivieren, um eine neue Zeitreihe für Kapazität zu erstellen, wenn sich Ihre Kapazitätsmetrikdaten über mehrere Auto-Scaling-Gruppen erstrecken. In diesem Fall müssen Sie metrische Mathematik verwenden, um die Daten zu einer einzigen Zeitreihe zu aggregieren.
7. Für Zielauslastung geben Sie den Zielwert ein, den Amazon EC2 Auto Scaling beibehalten werden soll. Amazon EC2 Auto Scaling skaliert Ihre Kapazität, bis die durchschnittliche

Auslastung der Zielauslastung entspricht oder bis sie die von Ihnen angegebene maximale Anzahl von Instances erreicht.

8. (Optional) Für Vorabstarten von Instances wählen Sie aus, wie weit im Voraus Ihre Instances gestartet werden sollen, bevor die Prognose die Last erhöht.
9. (Optional) Für Verhalten bei max. Kapazität wählen Sie aus, ob Sie Amazon EC2 Auto Scaling höher als die maximale Kapazität der Gruppe skalieren lassen, wenn die prognostizierte Kapazität das definierte Maximum überschreitet. Wenn Sie diese Einstellung aktivieren, kann die Skalierung in Zeiten erfolgen, in denen der höchste Datenverkehr vorausgesagt wird.
10. (Optional) Für Maximale Pufferkapazität oberhalb der prognostizierten Kapazität wählen Sie die zusätzliche Kapazität aus, die Sie verwenden möchten, wenn die prognostizierte Kapazität bei der maximalen Kapazität liegt oder diese überschreitet. Der Wert wird als Prozentsatz relativ zur prognostizierten Kapazität angegeben. Beispiel: Wenn der Puffer 10 ist, bedeutet dies, dass ein Puffer von 10 Prozent vorhanden ist. Wenn daher die prognostizierte Kapazität 50 ist und die maximale Kapazität 40 ist, dann ist die effektive maximale Kapazität 55.

    Wenn die Option 0 ist, kann Amazon EC2 Auto Scaling eine Kapazität skalieren, die höher als die maximale Kapazität ist, um der prognostizierten Kapazität zu entsprechen, diese aber nicht zu überschreiten.
11. Klicken Sie auf Erstellen einer Richtlinie für die prädiktive Skalierung.

Erstellen einer Richtlinie für die prädiktive Skalierung (AWS CLI)

Gehen Sie AWS CLI wie folgt vor, um Predictive Scaling-Richtlinien für Ihre Auto Scaling Scaling-Gruppe zu konfigurieren. Ersetzen Sie jedes *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Weitere Informationen zu den CloudWatch Metriken, die Sie angeben können, finden Sie PredictiveScalingMetricSpecificationin der Amazon EC2 Auto Scaling API-Referenz.

Beispiel 1: Eine Richtlinie für die prädiktive Skalierung, die Prognosen erstellt, aber nicht skaliert

Die folgende Beispielrichtlinie zeigt eine vollständige Richtlinienkonfiguration, die CPU-Auslastungsmetriken für die prädiktive Skalierung mit einer Zielauslastung von `40` verwendet. `ForecastOnly` wird standardmäßig verwendet, es sei denn, Sie geben explizit an, welcher Modus verwendet werden soll. Speichern Sie diese Konfiguration in einer Datei mit dem Namen `config.json`.

```
{
```

```
    "MetricSpecifications": [
        {
            "TargetValue": 40,
            "PredefinedMetricPairSpecification": {
                "PredefinedMetricType": "ASGCPUUtilization"
            }
        }
    ]
}
```

Um die Richtlinie von der Befehlszeile aus zu erstellen, führen Sie den put-scaling-policyBefehl mit der angegebenen Konfigurationsdatei aus, wie im folgenden Beispiel gezeigt.

```
aws autoscaling put-scaling-policy --policy-name cpu40-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt er den Amazon-Ressourcennamen (ARN) der Richtlinie zurück.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/cpu40-predictive-scaling-
policy",
  "Alarms": []
}
```

Beispiel 2: Eine Richtlinie für die prädiktive Skalierung, die Prognosen erstellt und skaliert

Fügen Sie für eine Richtlinie, mit der Amazon EC2 Auto Scaling prognostiziert und skaliert werden kann, die Eigenschaft `Mode` mit einem Wert von`ForecastAndScale` hinzu. Das folgende Beispiel zeigt eine Richtlinienkonfiguration, die Anforderungsanzahlmetriken der Application Load Balancer verwendet. Die Zielauslastung ist `1000` und die prädiktive Skalierung ist aufden Modus `ForecastAndScale` eingestellt.

```
{
    "MetricSpecifications": [
        {
            "TargetValue": 1000,
            "PredefinedMetricPairSpecification": {
                "PredefinedMetricType": "ALBRequestCount",
```

```
                "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-
target-group/943f017f100becff"
            }
        }
    ],
    "Mode": "ForecastAndScale"
}
```

Um diese Richtlinie zu erstellen, führen Sie den put-scaling-policyBefehl mit der angegebenen Konfigurationsdatei aus, wie im folgenden Beispiel gezeigt.

```
aws autoscaling put-scaling-policy --policy-name alb1000-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt er den Amazon-Ressourcennamen (ARN) der Richtlinie zurück.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-
id:scalingPolicy:19556d63-7914-4997-8c81-d27ca5241386:autoScalingGroupName/my-
asg:policyName/alb1000-predictive-scaling-policy",
  "Alarms": []
}
```

Beispiel 3: Eine Richtlinie für die prädiktive Skalierung, die höher als die maximale Kapazität skaliert werden kann

Das folgende Beispiel zeigt, wie eine Richtlinie erstellt wird, die höher als die maximale Größenbeschränkung der Gruppe skaliert werden kann, wenn Sie eine höhere Last als die normale Last benötigen. Standardmäßig skaliert Amazon EC2 Auto Scaling Ihre EC2-Kapazität nicht höher als Ihre definierte maximale Kapazität. Es kann jedoch hilfreich sein, eine höhere Skalierung mit etwas mehr Kapazität zu ermöglichen, um Leistungs- oder Verfügbarkeitsprobleme zu vermeiden.

Um Amazon EC2 Auto Scaling Platz für die Bereitstellung zusätzlicher Kapazität zu bieten, wenn die Kapazität laut Prognose die maximale Größe Ihrer Gruppe erreichen oder ihr sehr nahe kommen wird, geben Sie die Eigenschaften `MaxCapacityBreachBehavior` und `MaxCapacityBuffer` wie im folgenden Beispiel gezeigt an. Sie müssen `MaxCapacityBreachBehavior` mit einem positiven Wert für `IncreaseMaxCapacity` angeben. Die maximale Anzahl von Instances, die Ihre Gruppe haben kann, hängt vom Wert von `MaxCapacityBuffer` ab.

```
{
    "MetricSpecifications": [
        {
            "TargetValue": 70,
            "PredefinedMetricPairSpecification": {
                "PredefinedMetricType": "ASGCPUUtilization"
            }
        }
    ],
    "MaxCapacityBreachBehavior": "IncreaseMaxCapacity",
    "MaxCapacityBuffer": 10
}
```

In diesem Beispiel ist die Richtlinie so konfiguriert, dass sie einen 10-Prozent-Puffer (`"MaxCapacityBuffer": 10`) verwendet. Wenn die prognostizierte Kapazität also 50 und die maximale Kapazität 40 ist, ist die effektive maximale Kapazität 55. Eine Richtlinie, die eine Kapazität skalieren kann, die höher als die maximale Kapazität ist, um der prognostizierten Kapazität zu entsprechen, diese aber nicht zu überschreiten, hätte einen Puffer von 0 (`"MaxCapacityBuffer": 0`).

Um diese Richtlinie zu erstellen, führen Sie den put-scaling-policyBefehl mit der angegebenen Konfigurationsdatei aus, wie im folgenden Beispiel gezeigt.

```
aws autoscaling put-scaling-policy --policy-name cpu70-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt er den Amazon-Ressourcennamen (ARN) der Richtlinie zurück.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:d02ef525-8651-4314-
bf14-888331ebd04f:autoScalingGroupName/my-asg:policyName/cpu70-predictive-scaling-
policy",
  "Alarms": []
}
```

Auswertung Ihrer Richtlinien für prädiktive Skalierung

Bevor Sie eine prädiktive Skalierungsrichtlinie zur Skalierung Ihrer Auto-Scaling-Gruppe verwenden, überprüfen Sie Ihre Empfehlungen und andere Daten zu Ihrer Richtlinie in der Konsole von Amazon EC2 Auto Scaling. Dies ist wichtig, denn eine Richtlinie für prädiktive Skalierung soll Ihre tatsächliche Kapazität erst dann skalieren, wenn Sie wissen, dass die Prognosen korrekt sind.

Wenn die Auto-Scaling-Gruppe neu ist, muss es mindestens 24 Stunden an Daten liefern, bevor Amazon EC2 Auto Scaling eine Prognose erstellen kann.

Wenn Amazon EC2 Auto Scaling eine Prognose erstellt, verwendet es Verlaufsdaten. Wenn Ihre Auto-Scaling-Gruppe nicht über ausreichend aktuelle Verlaufsdaten verfügt, füllt Amazon EC2 Auto Scaling die Prognose möglicherweise vorübergehend mit Aggregaten auf, die aus den aktuell verfügbaren Verlaufsaggregaten erstellt wurden. Prognosen werden bis zu zwei Wochen vor dem Erstellungsdatum einer Richtlinie aufgefüllt.

Inhalt

Anzeigen Ihrer Richtlinien für prädiktive Skalierung

Für eine effektive Analyse sollte Amazon EC2 Auto Scaling über mindestens zwei Richtlinien für prädiktive Skalierung zum Vergleich verfügen. (Sie können die Ergebnisse jedoch weiterhin für eine einzelne Richtlinie überprüfen.) Wenn Sie mehrere Richtlinien erstellen, können Sie eine Richtlinie, die eine Metrik verwendet, gegen eine Richtlinie auswerten, die eine andere Metrik verwendet. Sie können auch die Auswirkungen verschiedener Zielwert- und Metrikkombinationen bewerten. Nachdem die Richtlinien für prädiktive Skalierung erstellt wurden, beginnt Amazon EC2 Auto Scaling sofort mit der Auswertung, welche Richtlinie für die Skalierung Ihrer Gruppe besser geeignet wäre.

So zeigen Sie Ihre Empfehlungen in der Konsole von Amazon EC2 Auto Scaling an

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.

3. Auf der Registerkarte Automatische Skalierung finden Sie unter Richtlinien für prädiktive Skalierung Details zu einer Richtlinie sowie unsere Empfehlung. In der Empfehlung erfahren Sie, ob es besser wäre, die Richtlinie für prädiktive Skalierung zu verwenden, oder nicht.

   Wenn Sie sich nicht sicher sind, ob eine prädiktive Skalierungsrichtlinie für Ihre Gruppe geeignet ist, prüfen Sie die Spalten Auswirkungen auf die Verfügbarkeit und Auswirkungen auf die Kosten, um die richtige Richtlinie auszuwählen. Die Informationen in den einzelnen Spalten geben Aufschluss über die Auswirkungen der jeweiligen Richtlinie.

   - Auswirkungen auf die Verfügbarkeit: Beschreibt, ob die Richtlinie negative Auswirkungen auf die Verfügbarkeit vermeiden würde, indem genügend Instances zur Bewältigung des Workloads bereitgestellt werden, und zieht einen Vergleich für den Fall, dass die Richtlinie nicht verwendet wird.
   - Auswirkungen auf die Kosten: Beschreibt, ob die Richtlinie negative Auswirkungen auf Ihre Kosten vermeiden würde, indem Instances nicht übermäßig bereitgestellt werden, und zieht einen Vergleich für den Fall, dass die Richtlinie nicht verwendet wird. Eine zu hohe Bereitstellung führt dazu, dass Ihre Instances nicht ausgelastet sind oder sich im Leerlauf befinden, was die Kosten nur noch weiter erhöht.

   Wenn Sie über mehrere Richtlinien verfügen, wird neben dem Namen der Richtlinie, die die meisten Verfügbarkeitsvorteile zu geringeren Kosten bietet, ein Tag für die Beste Prognose angezeigt. Die Auswirkungen auf die Verfügbarkeit werden stärker gewichtet.

4. (Optional) Um den gewünschten Zeitraum für die Empfehlungsergebnisse auszuwählen, wählen Sie den gewünschten Wert aus der Dropdown-Liste Auswertungszeitraum: 2 Tage, 1 Woche, 2 Wochen, 4 Wochen, 6 Wochen oder 8 Wochen. Standardmäßig wird der Auswertungszeitraum auf die letzten zwei Wochen festgelegt. Ein längerer Auswertungszeitraum liefert mehr Datenpunkte für die Empfehlungsergebnisse. Das Hinzufügen weiterer Datenpunkte verbessert die Ergebnisse jedoch möglicherweise nicht, wenn sich Ihre Lastmuster geändert haben, z. B. nach einer Phase außergewöhnlich hoher Nachfrage. In diesem Fall können Sie eine gezieltere Empfehlung erhalten, indem Sie sich aktuellere Daten ansehen.

> **Note**
>
> Empfehlungen werden nur für Richtlinien generiert, die sich im Modus Nur Prognose befinden. Die Empfehlungsfunktion liefert bessere Ergebnisse, wenn sich eine Richtlinie während des gesamten Bewertungszeitraums im Modus Nur Prognose befindet. Wenn Sie

eine Richtlinie im Prognose- und Skalierungsmodus starten und diese später in den Modus Nur Prognose wechselt, sind die Ergebnisse für diese Richtlinie wahrscheinlich verzerrt. Dies liegt daran, dass die Richtlinie bereits zur tatsächlichen Kapazität beigetragen hat.

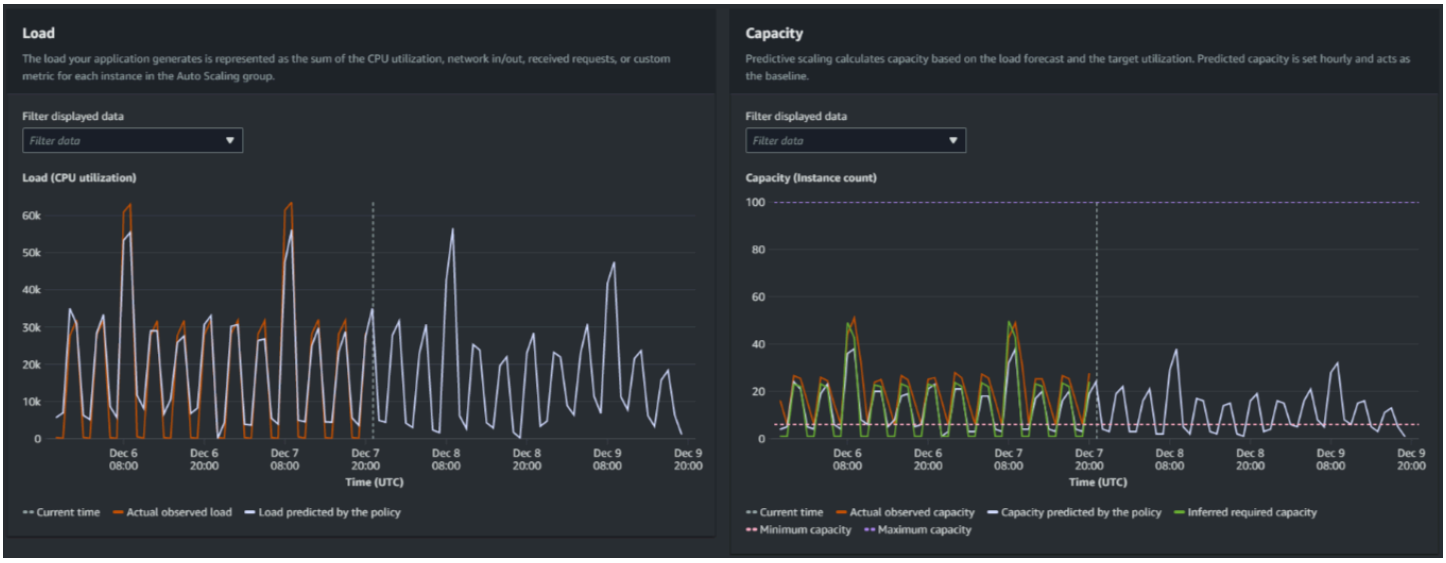
Anzeigen von Diagrammen zur Überwachung der prädiktiven Skalierung

In der Konsole von Amazon EC2 Auto Scaling können Sie die Prognose der vergangenen Tage, Wochen oder Monate überprüfen, um zu visualisieren, wie gut die Richtlinie im Laufe der Zeit funktioniert. Sie können diese Informationen auch zur Auswertung der Genauigkeit von Vorhersagen verwenden, wenn Sie entscheiden, ob Sie Ihre tatsächliche Kapazität durch eine Richtlinie skalieren lassen möchten.

So überprüfen Sie Überwachungsdiagramme für die prädiktive Skalierung in der Konsole von Amazon EC2 Auto Scaling

1. Wählen Sie eine Richtlinie aus der Liste Richtlinien für prädiktive Skalierung aus.
2. Im Abschnitt Überwachung können Sie die vergangenen und zukünftigen Prognosen Ihrer Richtlinie für Last und Kapazität im Vergleich zu tatsächlichen Werten anzeigen. Das Diagramm für Last zeigt Auslastungsprognosen und tatsächliche Werte für die ausgewählte Auslastungsmetrik. Das Diagramm zur Kapazität zeigt die Anzahl der von der Richtlinie vorhergesagten Instances. Es enthält auch die tatsächliche Anzahl der gestarteten Instances. Die vertikale Linie trennt Verlaufswerte von zukünftigen Prognosen. Diese Diagramme stehen kurz nach der Erstellung der Richtlinie zur Verfügung.
3. (Optional) Um die Menge der im Diagramm angezeigten Verlaufsdaten zu ändern, wählen Sie Ihren bevorzugten Wert aus der Dropdown-Liste Auswertungszeitraum oben auf der Seite aus. Der Auswertungszeitraum verändert die Daten auf dieser Seite in keiner Weise. Er ändert nur die Menge der angezeigten Verlaufsdaten.

Das folgende Image zeigt die Diagramme für Last und Kapazität, wenn Prognosen mehrfach angewendet wurden. Die prädiktive Skalierung prognostiziert die Last basierend auf Ihren historischen Lastdaten. Die von Ihrer Anwendung erzeugte Last wird als Summe der CPU-Auslastung, des Netzwerkeingangs/-ausgangs, der empfangenen Anfragen oder der benutzerdefinierten Metrik für jede Instance in der Auto-Scaling-Gruppe dargestellt. Die prädiktive Skalierung berechnet den zukünftigen Kapazitätsbedarf basierend auf der Lastprognose und der Zielauslastung, die Sie für die Skalierungsmetrik erreichen möchten.



Vergleichen von Daten im Diagramm für Last

Jede horizontale Linie stellt einen anderen Satz von Datenpunkten dar, die in einstündigen Intervallen gemeldet werden:

1. Tatsächliche beobachtete Last verwendet die SUM-Statistik für die von Ihnen gewählte Lastmetrik, um die gesamte stündliche Last im Verlauf anzuzeigen.
2. Von der Richtlinie prognostizierte Last zeigt die stündliche Lastprognose. Diese Prognose basiert auf den tatsächlichen Lastbeobachtungen der letzten zwei Wochen.

Vergleichen von Daten im Diagramm zur Kapazität

Jede horizontale Linie stellt einen anderen Satz von Datenpunkten dar, die in einstündigen Intervallen gemeldet werden:

1. Tatsächliche beobachtete Kapazität zeigt die tatsächliche Kapazität Ihrer Auto-Scaling-Gruppe in der Vergangenheit an, die von Ihren anderen Skalierungsrichtlinien und der für den ausgewählten Zeitraum geltenden Mindestgruppengröße abhängt.
2. Von der Richtlinie prognostizierte Kapazität zeigt die Basiskapazität an, die Sie zu Beginn jeder Stunde erwarten können, wenn sich die Richtlinie im Modus Prognose und Skalierung befindet.
3. Abgeleitete erforderliche Kapazität zeigt die ideale Kapazität, um die Skalierungsmetrik auf dem von Ihnen gewählten Zielwert zu halten.
4. Mindestkapazität zeigt die Mindestkapazität der Auto-Scaling-Gruppe an.
5. Maximale Kapazität zeigt die maximale Kapazität der Auto-Scaling-Gruppe.

Um die abgeleitete erforderliche Kapazität zu berechnen, gehen wir zunächst davon aus, dass jede Instance bei einem bestimmten Zielwert gleichmäßig ausgelastet ist. In der Praxis werden Instances nicht gleichmäßig ausgelastet. Wenn wir jedoch davon ausgehen, dass die Auslastung gleichmäßig auf die Instances verteilt ist, können wir eine wahrscheinliche Schätzung der benötigten Kapazität vornehmen. Der Kapazitätsbedarf wird dann umgekehrt proportional zu der Skalierungsmetrik berechnet, die Sie für Ihre Richtlinie für prädiktive Skalierung verwendet haben. Mit anderen Worten heißt das: Wenn die Kapazität zunimmt, nimmt die Skalierungsmetrik im gleichen Maß ab. Wenn sich beispielsweise die Kapazität verdoppelt, muss die Skalierungsmetrik um die Hälfte verringert werden.

Die Formel für die abgeleitete erforderliche Kapazität lautet wie folgt:

```
sum of (actualCapacityUnits*scalingMetricValue)/(targetUtilization)
```

Als Beispiel nehmen wir den `actualCapacityUnits` (`10`) und den `scalingMetricValue` (`30`) für eine bestimmte Stunde her. Wir nehmen dann die `targetUtilization`, die Sie in Ihrer Richtlinie für prädiktive Skalierung (`60`) angegeben haben, und berechnen die abgeleitete erforderliche Kapazität für dieselbe Stunde. Dies gibt den Wert `5` zurück. Das bedeutet, dass fünf die abgeleitete Kapazität ist, die erforderlich ist, um die Kapazität im direkt umgekehrten Verhältnis zum Zielwert der Skalierungsmetrik zu erhalten.

> **Note**
>
> Es stehen Ihnen verschiedene Möglichkeiten zur Verfügung, mit denen Sie die Kosteneinsparungen und die Verfügbarkeit Ihrer Anwendung verbessern können.
>
> - Sie verwenden die prädiktive Skalierung für die Basiskapazität und die dynamische Skalierung für den Umgang mit zusätzlicher Kapazität. Die dynamische Skalierung funktioniert unabhängig von der prädiktiven Skalierung, indem sie basierend auf der aktuellen Auslastung ab- und aufskaliert. Zunächst berechnet Amazon EC2 Auto Scaling die empfohlene Anzahl von Instances für jede dynamische Skalierungsrichtlinie. Anschließend skaliert die Lösung basierend auf der Richtlinie, die die größte Anzahl von Instances bereitstellt.
> - Damit bei sinkender Last eine Abskalierung erfolgen kann, sollte Ihre Auto-Scaling-Gruppe immer über mindestens eine dynamische Skalierungsrichtlinie verfügen, bei der das Abskalieren aktiviert ist.
> - Sie können die Skalierungsleistung verbessern, indem Sie sicherstellen, dass Ihre Mindest- und Höchstkapazität nicht zu restriktiv ist. Eine Richtlinie mit einer empfohlenen Anzahl von

Instances, die nicht innerhalb des Mindest- und Höchstkapazitätsbereichs liegt, wird an der Ab- und Aufskalierung gehindert.

Überwachen Sie Metriken zur vorausschauenden Skalierung mit CloudWatch

Je nach Ihren Anforderungen ziehen Sie es möglicherweise vor, auf Überwachungsdaten für vorausschauende Skalierung von Amazon zuzugreifen, CloudWatch anstatt auf die Amazon EC2 Auto Scaling Scaling-Konsole zuzugreifen. Nach Erstellung einer prädiktiven Skalierungsrichtlinie werden Daten gesammelt, um Ihre zukünftige Last und Kapazität zu prognostizieren. Nachdem diese Daten erfasst wurden, werden sie automatisch in regelmäßigen CloudWatch Abständen gespeichert. Anschließend können Sie visualisieren, wie gut CloudWatch sich die Richtlinie im Laufe der Zeit entwickelt. Sie können auch CloudWatch Alarme erstellen, um Sie zu benachrichtigen, wenn sich Leistungsindikatoren über die von Ihnen definierten Grenzwerte hinaus ändern CloudWatch.

Themen

Visualisieren historischer Prognosedaten

Die Last- und Kapazitätsprognosedaten für eine Richtlinie zur vorausschauenden Skalierung finden Sie unter CloudWatch. Dies kann nützlich sein, wenn Sie Prognosen im Vergleich zu anderen CloudWatch Kennzahlen in einem einzigen Diagramm visualisieren möchten. Es kann auch hilfreich sein, wenn Sie einen größeren Zeitraum anzeigen, um Trends im Zeitverlauf zu erkennen. Ihnen stehen historische Metriken von bis zu 15 Monaten zur Verfügung, um die Leistung Ihrer Richtlinie besser analysieren zu können.

Weitere Informationen finden Sie unter Metriken und Dimensionen für die prädiktive Skalierung.

Um historische Prognosedaten mit der CloudWatch Konsole anzuzeigen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus.
3. Wählen Sie Metrik-Namespace Auto Scaling aus.
4. Wählen Sie eine der folgenden Optionen aus, um entweder die Lastprognose- oder die Kapazitätsprognosemetriken anzuzeigen:

- Prädiktive Skalierung: Lastprognosen
- Prädiktive Skalierung: Kapazitätsprognosen

5. Geben Sie im Suchfeld den Namen der prädiktiven Skalierungsrichtlinie oder den Namen der Auto-Scaling-Gruppe ein, und drücken Sie dann die Eingabetaste, um die Ergebnisse zu filtern.
6. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren. Wenn Sie den Namen des Diagramms ändern möchten, wählen Sie das Bleistiftsymbol. Wenn Sie den Zeitraum ändern möchten, müssen Sie einen der vordefinierten Werte oder custom (benutzerdefiniert) auswählen. Weitere Informationen finden Sie unter Grafische Darstellung einer Metrik im CloudWatch Amazon-Benutzerhandbuch.
7. Wenn Sie die Statistik ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert und anschließend eine andere Statistik aus. Sie können zwar für jede Metrik eine beliebige Statistik wählen, aber nicht alle Statistiken sind für PredictiveScalingLoadForecastMetriken PredictiveScalingCapacityForecastnützlich. So sind zum Beispiel die Statistiken Durchschnitt, Minimum und Maximum hilfreich, die Statistik Summe jedoch nicht.
8. Wenn Sie dem Diagramm eine weitere Metrik hinzufügen möchten, wählen Sie unter Browse (Durchsuchen) die Option All (Alle) aus, suchen Sie nach der spezifischen Metrik, und aktivieren Sie dann das zugehörige Kontrollkästchen. Sie können bis zu 10 Metriken hinzufügen.

   Wenn Sie dem Diagramm beispielsweise die tatsächlichen Werte für die CPU-Auslastung hinzufügen möchten, wählen Sie den Namespace EC2 und anschließend By Auto Scaling Group (Nach Auto-Scaling-Gruppe) aus. Aktivieren Sie dann das Kontrollkästchen für die Metrik CPUUtilization und die spezifische Auto-Scaling-Gruppe.
9. (Optional) Um das Diagramm zu einem CloudWatch Dashboard hinzuzufügen, wählen Sie Aktionen, Zum Dashboard hinzufügen aus.

Erstellen von Genauigkeitsmetriken mithilfe von Metrikberechnungen

Mit 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 auf 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.

Mithilfe von Metrikberechnungen können Sie die Daten, die Amazon EC2 Auto Scaling für die prädiktive Skalierung generiert, auf unterschiedliche Weise grafisch darstellen. So können Sie die

Leistung von Richtlinien im Zeitverlauf überwachen und erkennen, ob Ihre Kombination von Metriken möglicherweise verbessert werden kann.

Sie können beispielsweise einen Metrikberechnungsausdruck verwenden, um den Mean Absolute Percentage Error (MAPE) zu überwachen. Die MAPE-Metrik hilft bei der Überwachung der Differenz zwischen den prognostizierten Werten und den tatsächlichen Werten eines bestimmten Prognosefensters. Änderungen des MAPE-Werts können Aufschluss darüber geben, ob sich die Leistung der Richtlinie im Laufe der Zeit verschlechtert, wenn sich Ihre Anwendung verändert. Eine Erhöhung des MAPE-Werts bedeutet eine größere Diskrepanz zwischen prognostizierten und tatsächlichen Werten.

Beispiel: Metrikberechnungsausdruck

Für die ersten Schritte mit dieser Art von Diagramm können Sie beispielsweise den Metrikberechnungsausdruck aus dem folgenden Beispiel erstellen.

```
{
  "MetricDataQueries": [
    {
      "Expression": "TIME_SERIES(AVG(ABS(m1-m2)/m1))",
      "Id": "e1",
      "Period": 3600,
      "Label": "MeanAbsolutePercentageError",
      "ReturnData": true
    },
    {
      "Id": "m1",
      "Label": "ActualLoadValues",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "CPUUtilization",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg"
            }
          ]
        },
        "Period": 3600,
        "Stat": "Sum"
      },
```

```
      "ReturnData": false
    },
    {
      "Id": "m2",
      "Label": "ForecastedLoadValues",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/AutoScaling",
          "MetricName": "PredictiveScalingLoadForecast",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg"
            },
            {
              "Name": "PolicyName",
              "Value": "my-predictive-scaling-policy"
            },
            {
              "Name": "PairIndex",
              "Value": "0"
            }
          ]
        },
        "Period": 3600,
        "Stat": "Average"
      },
      "ReturnData": false
    }
  ]
}
```

Anstelle einer einzelnen Metrik gibt es für `MetricDataQueries` ein Array von Abfragestrukturen für Metrikdaten. Jedes Element in `MetricDataQueries` ruft eine Metrik ab oder wendet einen mathematischen Ausdruck an. Das erste Element (`e1`) ist der mathematische Ausdruck. Der angegebene Ausdruck legt den Parameter `ReturnData` auf `true` fest, was letztendlich eine einzelne Zeitreihe generiert. Für alle anderen Metriken hat `ReturnData` den Wert `false`.

In diesem Beispiel verwendet der angegebene Ausdruck die tatsächlichen und prognostizierten Werte als Eingabe und gibt die neue Metrik (MAPE) zurück. `m1`ist die CloudWatch Metrik, die die tatsächlichen Lastwerte enthält (vorausgesetzt, die CPU-Auslastung ist die Lastmetrik, die ursprünglich für die genannte `my-predictive-scaling-policy` Richtlinie angegeben wurde).

`m2`ist die CloudWatch Metrik, die die prognostizierten Lastwerte enthält. Die mathematische Syntax für die MAPE-Metrik lautet wie folgt:

Durchschnitt von (abs ((tatsächlicher Wert - prognostizierter Wert)/(tatsächlichen Wert)))

Visualisieren Ihrer Genauigkeitsmetriken und Festlegen von Alarmen

Um die Genauigkeitsmetrikdaten zu visualisieren, wählen Sie in der CloudWatch Konsole die Registerkarte Metriken aus. Von dort aus können Sie die Daten grafisch darstellen. Weitere Informationen finden Sie unter Hinzufügen eines mathematischen Ausdrucks zu einem CloudWatch Diagramm im CloudWatch Amazon-Benutzerhandbuch.

Im Abschnitt Metrics (Metriken) können Sie auch einen Alarm für eine von Ihnen überwachte Metrik festlegen. Wählen Sie auf der Registerkarte Graphed metrics (Grafisch dargestellte Metriken) unter der Spalte Actions (Aktionen) das Symbol Create alarm (Alarm erstellen) aus. Das Symbol Create alarm (Alarm erstellen) wird als kleine Glocke dargestellt. Weitere Informationen und Benachrichtigungsoptionen finden Sie unter Erstellen eines CloudWatch Alarms auf der Grundlage eines metrischen mathematischen Ausdrucks und Benachrichtigung von Benutzern über Alarmänderungen im CloudWatch Amazon-Benutzerhandbuch.

Alternativ können Sie GetMetricDataund verwenden, PutMetricAlarmum Berechnungen mithilfe metrischer Mathematik durchzuführen und Alarme auf der Grundlage der Ausgabe zu erstellen.

Überschreiben von Prognosewerten mithilfe geplanter Aktionen

Manchmal haben Sie möglicherweise zusätzliche Informationen zu Ihren zukünftigen Anwendungsanforderungen, die bei der Prognoseberechnung nicht berücksichtigt werden können. Prognoseberechnungen können beispielsweise die Kapazität unterschätzen, die für eine bevorstehende Marketingveranstaltung benötigt wird. Sie können geplante Aktionen verwenden, um die Prognose in zukünftigen Zeiträumen vorübergehend zu überschreiben. Die geplanten Aktionen können auf einer wiederkehrenden Basis oder zu einem bestimmten Zeitpunkt ausgeführt werden, wenn einmalige Nachfrageschwankungen auftreten.

Sie können beispielsweise eine geplante Aktion mit einer höheren Mindestkapazität als die prognostizierte Aktion erstellen. Zur Laufzeit aktualisiert Amazon EC2 Auto Scaling die minimale Kapazität Ihrer Auto-Scaling-Gruppe. Da die prädiktive Skalierung für die Kapazität optimiert wird, wird eine geplante Aktion mit einer minimalen Kapazität, die höher als die Prognosewerte ist, berücksichtigt. Dadurch wird verhindert, dass die Kapazität geringer ist als erwartet. Um das Überschreiben der Prognose zu beenden, setzen Sie über eine zweite geplante Aktion die minimale Kapazität auf ihre ursprüngliche Einstellung zurück.

Im folgenden Verfahren werden die Schritte zum Überschreiben der Prognose in zukünftigen Zeiträumen erläutert.

Themen

- Schritt 1: (Optional) Analysieren von Zeitreihendaten
- Schritt 2: Erstellen von zwei geplanten Aktionen

> **Important**
>
> In diesem Thema wird davon ausgegangen, dass Sie versuchen, die Prognose zu überschreiben, um auf eine höhere Kapazität als die prognostizierte Kapazität zu skalieren. Wenn Sie die Kapazität vorübergehend verringern müssen, ohne dass dies durch eine Richtlinie zur vorausschauenden Skalierung beeinträchtigt wird, verwenden Sie stattdessen den Modus „Nur Prognose“. Im Modus „Nur Prognose“ generiert die vorausschauende Skalierung zwar weiterhin Prognosen, erhöht aber nicht automatisch die Kapazität. Anschließend können Sie die Ressourcennutzung überwachen und die Größe Ihrer Gruppe nach Bedarf manuell verringern. Weitere Informationen zur manuellen Skalierung finden Sie unterManuelle Skalierung für Amazon EC2 Auto Scaling.

Schritt 1: (Optional) Analysieren von Zeitreihendaten

Beginnen Sie mit der Analyse der Prognose-Zeitreihendaten. Dies ist ein optionaler Schritt, aber es ist hilfreich, wenn Sie die Details der Prognose verstehen möchten.

1. Rufen Sie die Prognose ab

   Nachdem die Prognose erstellt wurde, können Sie einen bestimmten Zeitraum in der Prognose abfragen. Ziel der Abfrage ist es, einen vollständigen Überblick über die Zeitreihendaten für einen bestimmten Zeitraum zu erhalten.

   Ihre Abfrage kann Prognosedaten bis zwei Tage in die Zukunft enthalten. Wenn Sie die prädiktive Skalierung eine Weile verwenden, können Sie auch auf Ihre früheren Prognosedaten zugreifen. Der maximale Zeitraum zwischen der Start- und Endzeit beträgt jedoch 30 Tage.

   Um die Prognose mithilfe des get-predictive-scaling-forecast AWS CLI Befehls abzurufen, geben Sie im Befehl die folgenden Parameter an:

- Geben Sie den Namen der Auto-Scaling-Gruppe im Feld `--auto-scaling-group-name`-Parameter an.
- Geben Sie den Namen der Richtlinie im `--policy-name`-Parameter an.
- Geben Sie die Startzeit im `--start-time`-Parameter an, um nur prognostizierte Daten für den angegebenen Zeitpunkt oder danach zurückzugeben.
- Geben Sie die Endzeit im `--end-time`-Parameter an, um nur prognostizierte Daten für den angegebenen Zeitpunkt oder davor zurückzugeben.

```
aws autoscaling get-predictive-scaling-forecast --auto-scaling-group-name my-asg \
    --policy-name cpu40-predictive-scaling-policy \
    --start-time "2021-05-19T17:00:00Z" \
    --end-time "2021-05-19T23:00:00Z"
```

Bei erfolgreicher Ausführung gibt der Befehl Daten zurück, die in etwa wie folgt aussehen:

```
{
    "LoadForecast": [
        {
            "Timestamps": [
                "2021-05-19T17:00:00+00:00",
                "2021-05-19T18:00:00+00:00",
                "2021-05-19T19:00:00+00:00",
                "2021-05-19T20:00:00+00:00",
                "2021-05-19T21:00:00+00:00",
                "2021-05-19T22:00:00+00:00",
                "2021-05-19T23:00:00+00:00"
            ],
            "Values": [
                153.0655799339254,
                128.8288551285919,
                107.1179447150675,
                197.3601844551528,
                626.4039934516954,
                596.9441277518481,
                677.9675713779869
            ],
            "MetricSpecification": {
                "TargetValue": 40.0,
                "PredefinedMetricPairSpecification": {
```

```
                    "PredefinedMetricType": "ASGCPUUtilization"
                }
            }
        }
    ],
    "CapacityForecast": {
        "Timestamps": [
            "2021-05-19T17:00:00+00:00",
            "2021-05-19T18:00:00+00:00",
            "2021-05-19T19:00:00+00:00",
            "2021-05-19T20:00:00+00:00",
            "2021-05-19T21:00:00+00:00",
            "2021-05-19T22:00:00+00:00",
            "2021-05-19T23:00:00+00:00"
        ],
        "Values": [
            2.0,
            2.0,
            2.0,
            2.0,
            4.0,
            4.0,
            4.0
        ]
    },
    "UpdateTime": "2021-05-19T01:52:50.118000+00:00"
}
```


Die Antwort enthält zwei Prognosen: `LoadForecast` und `CapacityForecast`. `LoadForecast` zeigt die stündliche Lastprognose an. `CapacityForecast` zeigt Prognosewerte für die Kapazität an, die stündlich benötigt wird, um die prognostizierte Last zu verarbeiten, während ein `TargetValue` von 40,0 (40 % durchschnittliche CPU-Auslastung) aufrechterhalten bleibt.

2. Identifizieren des Zielzeitraums

   Ermitteln Sie die Stunde oder die Stunden, zu der/zu denen die einmalige Nachfrageschwankung stattfinden soll. Denken Sie daran, dass die in der Prognose angezeigten Datumsangaben und Uhrzeiten in UTC angegeben sind.

Schritt 2: Erstellen von zwei geplanten Aktionen

Erstellen Sie als Nächstes zwei geplante Aktionen für einen bestimmten Zeitraum, in dem Ihre Anwendung eine höhere Last aufweist als die prognostizierte Last. Wenn Sie beispielsweise während eines Marketing-Ereignisses für einen begrenzten Zeitraum ein erhöhtes Datenvolumen erwarten, können Sie eine einmalige Aktion planen, um die Mindestkapazität bei deren Beginn zu aktualisieren. Planen Sie dann eine weitere Aktion, um die Mindestkapazität auf die ursprüngliche Einstellung zurückzusetzen, wenn das Ereignis endet.

Erstellen von zwei geplanten Aktionen für einmalige Ereignisse (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Automatic scaling (Automatische Skalierung) unter Scheduled actions (Geplante Aktionen) die Option Geplante Aktion erstellen (Create scheduled action) aus.
4. Geben Sie die folgenden Einstellungen für die geplante Aktion ein:

   a. Geben Sie einen Namen für die geplante Aktion ein.

   b. Für Min geben Sie die neue Mindestkapazität für Ihre Auto-Scaling-Gruppe ein. Der Min-Wert darf maximal so groß sein wie die Höchstgröße der Gruppe. Wenn Ihr Wert für Min größer ist als die Höchstgröße der Gruppe, müssen Sie Max aktualisieren.

   c. Wählen Sie für Recurrence (Wiederholung) Once (Einmal) aus.

   d. Wählen Sie für Zeitzone eine Zeitzone aus. Wenn keine Zeitzone gewählt ist, wird standardmäßig `ETC/UTC` verwendet.

   e. Definieren Sie eine Spezifische Startzeit.
5. Wählen Sie Erstellen.

   Die Konsole zeigt die geplanten Aktionen der Auto-Scaling-Gruppe an.
6. Konfigurieren Sie eine zweite geplante Aktion, damit die Mindestkapazität am Ende des Ereignisses wieder auf die ursprüngliche Einstellung zurückkehrt. Die prädiktive Skalierung kann die Kapazität nur skalieren, wenn der Wert, den Sie für Min angeben, niedriger ist als die Prognosewerte.

Erstellen von zwei geplanten Aktionen für einmalige Ereignisse (AWS CLI)

Verwenden Sie den Befehl put-scheduled-update-group-action, AWS CLI um die geplanten Aktionen zu erstellen.

Lassen Sie uns als Beispiel einen Zeitplan definieren, der am 19. Mai um 17:00 Uhr acht Stunden lang eine Mindestkapazität von drei Instances beibehält. Die folgenden Befehle veranschaulichen die Implementierung dieses Szenarios.

Der erste Befehl put-scheduled-update-group-action weist Amazon EC2 Auto Scaling an, die Mindestkapazität der angegebenen Auto Scaling Scaling-Gruppe am 19. Mai 2021 um 17:00 Uhr UTC zu aktualisieren.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-
start \
  --auto-scaling-group-name my-asg --start-time "2021-05-19T17:00:00Z" --minimum-
capacity 3
```

Der zweite Befehl weist Amazon EC2 Auto Scaling an, die Mindestkapazität der Gruppe am 20. Mai 2021 um 1:00 Uhr UTC auf eins zu setzen.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-end
 \
  --auto-scaling-group-name my-asg --start-time "2021-05-20T01:00:00Z" --minimum-
capacity 1
```

Nachdem Sie der Auto-Scaling-Gruppe diese geplanten Aktionen hinzugefügt haben, führt Amazon EC2 Auto Scaling folgende Schritte aus:

- Um 17:00 Uhr UTC am 19. Mai 2021 wird die erste geplante Aktion ausgeführt. Wenn die Gruppe derzeit weniger als drei Instances hat, wird die Gruppe auf drei Instances skaliert. Während dieser Zeit und für die Dauer der nächsten acht Stunden kann Amazon EC2 Auto Scaling weiterhin skaliert werden, wenn die prognostizierte Kapazität höher als die tatsächliche Kapazität ist oder wenn eine Richtlinie für dynamische Skalierung in Kraft ist.
- Um 1:00 Uhr UTC am 20. Mai 2021 wird die zweite geplante Aktion ausgeführt. Dadurch wird die Mindestkapazität am Ende des Ereignisses auf die ursprüngliche Einstellung zurückgesetzt.

Skalierung basierend auf wiederkehrenden Zeitplänen

Um die Prognose jede Woche während des gleichen Zeitraums zu überschreiben, erstellen Sie zwei geplante Aktionen und stellen die Zeit- und Datumslogik mithilfe eines Cron-Ausdrucks bereit.

Der Cron-Ausdruck besteht aus fünf Feldern, getrennt durch Leerzeichen: [Minute] [Stunde] [Tag_des_Monats] [Monat_des_Jahres] [Wochentag]. Felder können alle zulässigen Werte enthalten, einschließlich Sonderzeichen.

Beispielsweise führt der folgende Cron-Ausdruck jeden Dienstag um 6:30 Uhr die Aktion aus. Das Sternchen wird als Platzhalter verwendet, um alle Werte für ein Feld abzugleichen.

```
30 6 * * 2
```

Weitere Informationen finden Sie auch unter

Weitere Informationen zum Erstellen, Auflisten, Bearbeiten und Löschen von geplanten Aktionen finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling.

Erweiterte prädiktive Skalierungsrichtlinienkonfigurationen mit benutzerdefinierten Metriken

In einer prädiktiven Skalierungsrichtlinie können Sie vordefinierte oder benutzerdefinierte Metriken verwenden. Benutzerdefinierte Metriken sind nützlich, wenn die vordefinierten Metriken (CPU, Netzwerk-I/O und Anzahl der Anfragen an den Application Load Balancer) Ihre Anwendungslast nicht ausreichend beschreiben.

Wenn Sie eine Richtlinie für vorausschauende Skalierung mit benutzerdefinierten Metriken erstellen, können Sie andere CloudWatch Messwerte angeben, die von bereitgestellt werden AWS, oder Sie können Metriken angeben, die Sie selbst definieren und veröffentlichen. Sie können auch metrische Mathematik verwenden, um bestehende Metriken zu aggregieren und in eine neue Zeitreihe umzuwandeln, die AWS nicht automatisch erfasst wird. Wenn Sie Werte in Ihren Daten kombinieren, indem Sie z.B. neue Summen oder Durchschnittswerte berechnen, nennt man das Aggregieren. Die resultierenden Daten werden als Aggregat bezeichnet.

Der folgende Abschnitt enthält bewährte Verfahren und Beispiele für die Erstellung der JSON-Struktur für die Richtlinie.

Themen

Bewährte Methoden

Die folgenden bewährten Methoden können Ihnen helfen, benutzerdefinierte Metriken effektiver zu nutzen:

- Für die Spezifikation der Lastmetrik ist die nützlichste Metrik eine Metrik, die die Last einer Auto-Scaling-Gruppe als Ganzes darstellt, unabhängig von der Kapazität der Gruppe.
- Bei der Angabe der Skalierungsmetrik ist die sinnvollste Metrik für die Skalierung ein durchschnittlicher Durchsatz oder eine durchschnittliche Auslastung pro Instance.
- Die Skalierungsmetrik muss umgekehrt proportional zur Kapazität sein. Das heißt, wenn die Anzahl der Instances in der Auto-Scaling-Gruppe steigt, sollte die Skalierungsmetrik in etwa im gleichen Verhältnis sinken. Um sicherzustellen, dass sich die prädiktive Skalierung wie erwartet verhält, müssen die Lastmetrik und die Skalierungsmetrik auch stark miteinander korrelieren.
- Die Zielauslastung muss mit der Art der Skalierungsmetrik übereinstimmen. Bei einer Richtlinienkonfiguration, die die CPU-Auslastung verwendet, ist dies ein Zielprozentsatz. Bei einer Richtlinienkonfiguration, die den Durchsatz verwendet, wie z.B. die Anzahl der Anfragen oder Nachrichten, ist dies die angestrebte Anzahl von Anfragen oder Nachrichten pro Instance während eines einminütigen Intervalls.
- Wenn diese Empfehlungen nicht befolgt werden, werden die prognostizierten zukünftigen Werte der Zeitreihen wahrscheinlich falsch sein. Um zu überprüfen, ob die Daten korrekt sind, können Sie die prognostizierten Werte in der Amazon EC2 Auto Scaling-Konsole einsehen. Alternativ können Sie nach der Erstellung Ihrer Richtlinie für vorausschauende Skalierung die `CapacityForecast` Objekte `LoadForecast` und überprüfen, die durch einen GetPredictiveScalingForecastAPI-Aufruf zurückgegeben wurden.
- Wir empfehlen Ihnen dringend, die prädiktive Skalierung im Modus "Nur Prognose" zu konfigurieren, damit Sie die Prognose auswerten können, bevor die prädiktive Skalierung mit der aktiven Skalierung der Kapazität beginnt.

Voraussetzungen

Um benutzerdefinierte Metriken zu Ihrer prädiktiven Skalierungsrichtlinie hinzuzufügen, müssen Sie über entsprechende `cloudwatch:GetMetricData`-Berechtigungen verfügen.

Wenn Sie Ihre eigenen Metriken anstelle der bereitgestellten Metriken angeben möchten, müssen Sie Ihre Metriken zunächst auf CloudWatch veröffentlichen. AWS Weitere Informationen finden Sie unter Veröffentlichen benutzerdefinierter Metriken im CloudWatch Amazon-Benutzerhandbuch.

Sollten Sie Ihre eigenen Metriken veröffentlichen, achten Sie darauf, dass Sie die Datenpunkte mindestens alle fünf Minuten veröffentlichen. Amazon EC2 Auto Scaling ruft die Datenpunkte CloudWatch basierend auf der Länge des benötigten Zeitraums ab. Beispielsweise verwendet die Lastmetrikspezifikation stündliche Metriken, um die Auslastung Ihrer Anwendung zu messen. CloudWatch verwendet Ihre veröffentlichten Metrikdaten, um einen einzelnen Datenwert für einen beliebigen Zeitraum von einer Stunde bereitzustellen, indem alle Datenpunkte mit Zeitstempeln aggregiert werden, die in jeden Zeitraum von einer Stunde fallen.

Konstruieren von JSON für benutzerdefinierte Metriken

Der folgende Abschnitt enthält Beispiele für die Konfiguration der prädiktiven Skalierung für die Abfrage von Daten. CloudWatch Es gibt zwei verschiedene Methoden, um diese Option zu konfigurieren, und die von Ihnen gewählte Methode wirkt sich darauf aus, welches Format Sie verwenden, um den JSON für Ihre prädiktive Skalierungsrichtlinie zu erstellen. Wenn Sie metrische Berechnungen verwenden, variiert das Format von JSON je nach der durchgeführten metrischen Berechnung weiter.

1. Informationen zum Erstellen einer Richtlinie, mit der Daten direkt aus anderen CloudWatch Metriken abgerufen werden, die von bereitgestellt werden AWS oder für die Sie Daten veröffentlichen CloudWatch, finden Beispiel einer prädiktiven Skalierungsrichtlinie mit benutzerdefinierten Last- und Skalierungsmetriken (AWS CLI) Sie unter.
2. Informationen zum Erstellen einer Richtlinie, mit der mehrere CloudWatch Messwerte abgefragt und mithilfe mathematischer Ausdrücke neue Zeitreihen auf der Grundlage dieser Messwerte erstellt werden können, finden Sie unterMetrikberechnungs-Ausdrücke verwenden.

Beispiel einer prädiktiven Skalierungsrichtlinie mit benutzerdefinierten Last- und Skalierungsmetriken (AWS CLI)

Um eine prädiktive Skalierungsrichtlinie mit benutzerdefinierten Last- und Skalierungsmetriken mit dem zu erstellen AWS CLI, speichern Sie die Argumente für `--predictive-scaling-configuration` in einer JSON-Datei mit dem Namen`config.json`.

Sie beginnen mit dem Hinzufügen benutzerdefinierter Metriken, indem Sie die ersetzbaren Werte im folgenden Beispiel durch die Werte Ihrer Metriken und Ihrer Zielauslastung ersetzen.

```
{
  "MetricSpecifications": [
    {
      "TargetValue": 50,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "scaling_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "MyUtilizationMetric",
                "Namespace": "MyNameSpace",
                "Dimensions": [
                  {
                    "Name": "MyOptionalMetricDimensionName",
                    "Value": "MyOptionalMetricDimensionValue"
                  }
                ]
              },
              "Stat": "Average"
            }
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "MyLoadMetric",
                "Namespace": "MyNameSpace",
                "Dimensions": [
```

```
                {
                  "Name": "MyOptionalMetricDimensionName",
                  "Value": "MyOptionalMetricDimensionValue"
                }
              ]
            },
            "Stat": "Sum"
          }
        }
      ]
    }
  }
]
}
```

Weitere Informationen finden Sie MetricDataQueryin der Amazon EC2 Auto Scaling API-Referenz.

> **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.

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

```
aws autoscaling put-scaling-policy --policy-name my-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt er den Amazon-Ressourcennamen (ARN) der Richtlinie zurück.

```
{
```

```
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-predictive-scaling-policy",
  "Alarms": []
}
```

Metrikberechnungs-Ausdrücke verwenden

Der folgende Abschnitt enthält Informationen und Beispiele für Richtlinien zur vorausschauenden Skalierung, die zeigen, wie Sie metrische Berechnungen in Ihrer Richtlinie verwenden können.

Themen

- Metrikberechnung verstehen
- Beispiel für eine prädiktive Skalierungspolitik, die Metriken mit metrischer Mathematik kombiniert (AWS CLI)
- Beispiel für eine prädiktive Skalierungsrichtlinie in einem blau/grünen Einsatzszenario (AWS CLI)

Metrikberechnung verstehen

Wenn Sie lediglich vorhandene Metrikdaten aggregieren möchten, erspart Ihnen CloudWatch Metric Math den Aufwand und die Kosten für die Veröffentlichung einer weiteren Metrik in CloudWatch. Sie können jede verfügbare Metrik verwenden AWS , und Sie können auch Metriken verwenden, die Sie als Teil Ihrer Anwendungen definieren. Sie könnten zum Beispiel den Rückstand der Amazon SQS-Warteschlange pro Instance berechnen wollen. Dazu nehmen Sie die ungefähre Anzahl der Nachrichten, die für den Abruf aus der Warteschlange zur Verfügung stehen, und dividieren diese Zahl durch die laufende Kapazität der Auto-Scaling-Gruppe.

Weitere Informationen finden Sie unter Verwenden von metrischer Mathematik im CloudWatch Amazon-Benutzerhandbuch.

Wenn Sie sich für die Verwendung eines metrischen mathematischen Ausdrucks in Ihrer prädiktiven Skalierungsrichtlinie entscheiden, sollten Sie die folgenden Punkte beachten:

- Metrische Rechenoperationen verwenden die Datenpunkte der eindeutigen Kombination aus Metrikname, Namespace und Dimensionsschlüssel/Wertpaaren von Metriken.
- 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.

- Ihre metrischen mathematischen Ausdrücke können aus verschiedenen Aggregationen zusammengesetzt sein. Für das endgültige Aggregationsergebnis ist es jedoch eine bewährte Methode, `Average` für die Skalierungsmetrik und `Sum` für die Lastmetrik zu verwenden.
- Alle Ausdrücke, die in einer metrischen Spezifikation verwendet werden, müssen letztendlich eine einzige Zeitreihe ergeben.

Um metrische Mathematik zu verwenden, gehen Sie wie folgt vor:

- Wählen Sie eine oder mehrere CloudWatch Metriken aus. Erstellen Sie dann den Ausdruck. Weitere Informationen finden Sie unter Verwenden von metrischer Mathematik im CloudWatch Amazon-Benutzerhandbuch.
- Stellen Sie mithilfe der CloudWatch Konsole oder der CloudWatch GetMetricDataAPI sicher, dass der metrische mathematische Ausdruck gültig ist.

Beispiel für eine prädiktive Skalierungspolitik, die Metriken mit metrischer Mathematik kombiniert (AWS CLI)

Manchmal müssen Sie die Metrik nicht direkt angeben, sondern die Daten erst auf irgendeine Weise verarbeiten. Sie könnten zum Beispiel eine Anwendung haben, die Arbeit aus einer Amazon SQS-Warteschlange abruft, und Sie könnten die Anzahl der Objekte in der Warteschlange als Kriterium für die prädiktive Skalierung verwenden wollen. Die Anzahl der Nachrichten in der Warteschlange bestimmt nicht allein die Anzahl der Instances, die Sie benötigen. Daher ist weitere Arbeit erforderlich, um eine Metrik zu erstellen, die zur Berechnung des Rückstands pro Instance verwendet werden kann. Weitere Informationen finden Sie unter Skalierung basierend auf Amazon SQS.

Im Folgenden finden Sie ein Beispiel für eine prädiktive Skalierungsrichtlinie für dieses Szenario. Sie legt Skalierungs- und Auslastungsmetriken fest, die auf der Amazon SQS `ApproximateNumberOfMessagesVisible`-Metrik basieren, d.h. der Anzahl der Nachrichten, die für den Abruf aus der Warteschlange verfügbar sind. Es verwendet auch die Amazon EC2 Auto Scaling `GroupInServiceInstances`-Metrik und einen mathematischen Ausdruck, um den Rückstand pro Instance für die Skalierungsmetrik zu berechnen.

```
aws autoscaling put-scaling-policy --policy-name my-sqs-custom-metrics-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
{
  "MetricSpecifications": [
```

```
    {
      "TargetValue": 100,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "Label": "Get the queue size (the number of messages waiting to be
processed)",
            "Id": "queue_size",
            "MetricStat": {
              "Metric": {
                "MetricName": "ApproximateNumberOfMessagesVisible",
                "Namespace": "AWS/SQS",
                "Dimensions": [
                  {
                    "Name": "QueueName",
                    "Value": "my-queue"
                  }
                ]
              },
              "Stat": "Sum"
            },
            "ReturnData": false
          },
          {
            "Label": "Get the group size (the number of running instances)",
            "Id": "running_capacity",
            "MetricStat": {
              "Metric": {
                "MetricName": "GroupInServiceInstances",
                "Namespace": "AWS/AutoScaling",
                "Dimensions": [
                  {
                    "Name": "AutoScalingGroupName",
                    "Value": "my-asg"
                  }
                ]
              },
              "Stat": "Sum"
            },
            "ReturnData": false
          },
          {
            "Label": "Calculate the backlog per instance",
            "Id": "scaling_metric",
```

```
            "Expression": "queue_size / running_capacity",
            "ReturnData": true
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "ApproximateNumberOfMessagesVisible",
                "Namespace": "AWS/SQS",
                "Dimensions": [
                  {
                    "Name": "QueueName",
                    "Value": "my-queue"
                  }
                ],
              },
              "Stat": "Sum"
            },
            "ReturnData": true
          }
        ]
      }
    }
  ]
}
```

Das Beispiel gibt den ARN der Richtlinie zurück.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-sqs-custom-metrics-policy",
  "Alarms": []
}
```

Beispiel für eine prädiktive Skalierungsrichtlinie in einem blau/grünen Einsatzszenario (AWS CLI)

Ein Suchausdruck bietet eine erweiterte Option, mit der Sie eine Metrik aus mehreren Auto-Scaling-Gruppen abfragen und mathematische Ausdrücke auf sie anwenden können. Dies ist besonders nützlich für blau/grüne Bereitstellungen.

> **Note**
>
> Eine blau/grüne Bereitstellung ist eine Bereitstellungsmethode, bei der Sie zwei separate, aber identische Auto-Scaling-Gruppen erstellen. Nur eine der Gruppen empfängt den Produktionsverkehr. Der Benutzerverkehr wird zunächst auf die frühere ("blaue") Auto-Scaling-Gruppe geleitet, während eine neue Gruppe („grün“) zum Testen und Evaluieren einer neuen Version einer Anwendung oder eines Dienstes verwendet wird. Der Benutzerverkehr wird auf die grüne Auto-Scaling-Gruppe verlagert, nachdem eine neue Bereitstellung getestet und akzeptiert wurde. Sie können die blaue Gruppe dann löschen, nachdem die Bereitstellung erfolgreich war.

Wenn neue Auto-Scaling-Gruppen als Teil einer blau/grünen Bereitstellung erstellt werden, kann die Metrik-Historie jeder Gruppe automatisch in die prädiktive Skalierungsrichtlinie aufgenommen werden, ohne dass Sie ihre Metrik-Spezifikationen ändern müssen. Weitere Informationen finden Sie im Compute-Blog unter Verwenden von Richtlinien zur vorausschauenden Skalierung von EC2 Auto Scaling mit Blue/Green-Bereitstellungen. AWS

Die folgende Beispielrichtlinie zeigt, wie dies geschehen kann. In diesem Beispiel verwendet die Richtlinie die von Amazon EC2 ausgegebene `CPUUtilization`-Metrik. Es verwendet die Amazon EC2 Auto Scaling `GroupInServiceInstances`-Metrik und einen mathematischen Ausdruck, um den Wert der Skalierungsmetrik pro Instance zu berechnen. Sie gibt auch eine Kapazitätsmetrik an, um die `GroupInServiceInstances`-Metrik zu erhalten.

Der Suchausdruck findet das `CPUUtilization` von Instances in mehreren Auto-Scaling-Gruppen anhand der angegebenen Suchkriterien. Wenn Sie zu einem späteren Zeitpunkt eine neue Auto-Scaling-Gruppe erstellen, die denselben Suchkriterien entspricht, werden die `CPUUtilization` der Instances in der neuen Auto-Scaling-Gruppe automatisch einbezogen.

```
aws autoscaling put-scaling-policy --policy-name my-blue-green-predictive-scaling-
policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
{
  "MetricSpecifications": [
    {
      "TargetValue": 25,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
```

```
          {
            "Id": "load_sum",
            "Expression": "SUM(SEARCH('{AWS/EC2,AutoScalingGroupName} MetricName=
\"CPUUtilization\" ASG-myapp', 'Sum', 300))",
            "ReturnData": false
          },
          {
            "Id": "capacity_sum",
            "Expression": "SUM(SEARCH('{AWS/AutoScaling,AutoScalingGroupName}
 MetricName=\"GroupInServiceInstances\" ASG-myapp', 'Average', 300))",
            "ReturnData": false
          },
          {
            "Id": "weighted_average",
            "Expression": "load_sum / capacity_sum",
            "ReturnData": true
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_sum",
            "Expression": "SUM(SEARCH('{AWS/EC2,AutoScalingGroupName} MetricName=
\"CPUUtilization\" ASG-myapp', 'Sum', 3600))"
          }
        ]
      },
      "CustomizedCapacityMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "capacity_sum",
            "Expression": "SUM(SEARCH('{AWS/AutoScaling,AutoScalingGroupName}
 MetricName=\"GroupInServiceInstances\" ASG-myapp', 'Average', 300))"
          }
        ]
      }
    }
  ]
}
```

Das Beispiel gibt den ARN der Richtlinie zurück.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-blue-green-predictive-
scaling-policy",
  "Alarms": []
}
```

Überlegungen und Fehlerbehebung

Wenn bei der Verwendung von benutzerdefinierten Metriken ein Problem auftritt, empfehlen wir Ihnen, wie folgt vorzugehen:

- Wenn eine Fehlermeldung angezeigt wird, lesen Sie die Nachricht und beheben Sie das gemeldete Problem, falls möglich.
- Wenn ein Problem auftritt, wenn Sie versuchen, einen Suchausdruck in einem blau/grünen Bereitstellungsszenario zu verwenden, vergewissern Sie sich zunächst, dass Sie wissen, wie Sie einen Suchausdruck erstellen, der nach einer teilweisen Übereinstimmung anstelle einer genauen Übereinstimmung sucht. Vergewissern Sie sich außerdem, dass Ihre Abfrage nur die Auto-Scaling-Gruppen findet, in denen die betreffende Anwendung ausgeführt wird. Weitere Informationen zur Syntax von Suchausdrücken finden Sie unter Syntax von CloudWatch Suchausdrücken im CloudWatch Amazon-Benutzerhandbuch.
- Wenn Sie einen Ausdruck nicht im Voraus validiert haben, validiert ihn der put-scaling-policyBefehl, wenn Sie Ihre Skalierungsrichtlinie erstellen. Es besteht jedoch die Möglichkeit, dass dieser Befehl die genaue Ursache der erkannten Fehler nicht identifizieren kann. Um die Probleme zu beheben, beheben Sie die Fehler, die Sie als Antwort auf eine Anfrage an den get-metric-dataBefehl erhalten. Sie können den Ausdruck auch von der CloudWatch Konsole aus beheben.
- Wenn Sie Ihre Load (Last)- und Capacity (Kapazitäts)-Diagramme in der Konsole betrachten, zeigt das Capacity (Kapazitäts)-Diagramm möglicherweise keine Daten an. Um sicherzustellen, dass die Diagramme vollständige Daten enthalten, stellen Sie sicher, dass Sie die Gruppenmetriken für Ihre Auto-Scaling-Gruppen konsequent aktivieren. Weitere Informationen finden Sie unter Aktivieren der Auto-Scaling-Metriken (Konsole).
- Die Angabe der Kapazitätsmetrik ist nur für blau/grüne Bereitstellungen sinnvoll, wenn Sie Anwendungen haben, die während ihrer Lebensdauer in verschiedenen Auto-Scaling-Gruppen laufen. Mit dieser benutzerdefinierten Metrik können Sie die Gesamtkapazität mehrerer Auto-Scaling-Gruppen angeben. Die prädiktive Skalierung nutzt dies, um historische Daten in den Capacity (Kapazitäts)-Diagrammen in der Konsole anzuzeigen.

- Sie müssen `false` für `ReturnData` angeben, wenn `MetricDataQueries` die Funktion SEARCH() allein ohne eine mathematische Funktion wie SUM() angibt. Das liegt daran, dass Suchausdrücke mehrere Zeitreihen zurückgeben können, während eine auf einem Ausdruck basierende Metrikspezifikation nur eine Zeitreihe zurückgeben kann.
- Alle an einem Suchausdruck beteiligten Metriken sollten die gleiche Auflösung haben.

Einschränkungen

- Sie können Datenpunkte von bis zu 10 Metriken in einer Metrikspezifikation abfragen.
- Für die Zwecke dieses Limits zählt ein Ausdruck als eine Metrik.

Steuern welche Auto-Scaling-Instances beim Abskalieren beendet werden

Amazon EC2 Auto Scaling verwendet Kündigungsrichtlinien, um die Reihenfolge für das Beenden von Instances festzulegen. Sie können eine vordefinierte Richtlinie verwenden oder eine benutzerdefinierte Richtlinie erstellen, um Ihre spezifischen Anforderungen zu erfüllen. Durch die Verwendung einer benutzerdefinierten Richtlinie oder eines Instance Scale-In-Schutzes können Sie auch verhindern, dass Ihre Auto Scaling Scaling-Gruppe Instances beendet, die noch nicht zum Beenden bereit sind.

Inhalt

Wenn Amazon EC2 Auto Scaling Kündigungsrichtlinien verwendet

In den folgenden Abschnitten werden die Szenarien beschrieben, in denen Amazon EC2 Auto Scaling Beendigungsrichtlinien verwendet.

Inhalt

Scale-In Ereignisse

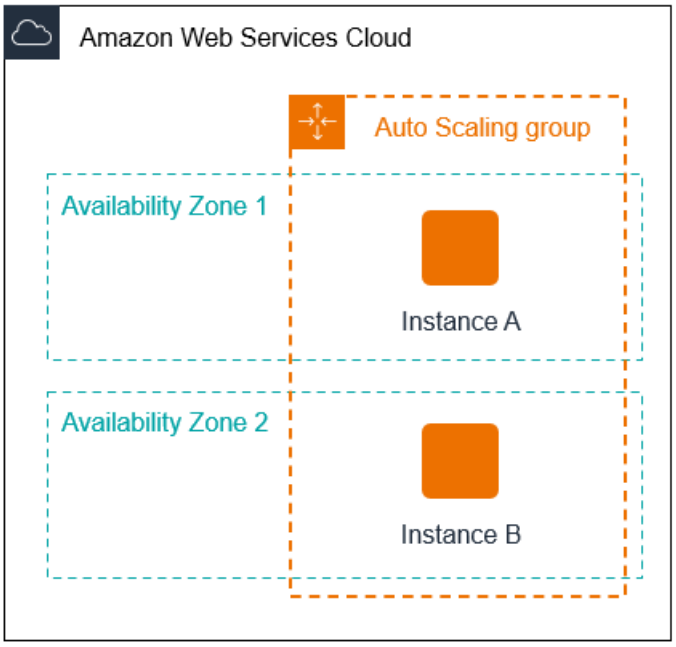
Ein Skale-in-Ereignis tritt auf, wenn ein neuer Wert für die gewünschte Kapazität einer Auto-Scaling-Gruppe vorhanden ist, der niedriger ist als die aktuelle Kapazität der Gruppe.

Scale-in-Ereignisse treten in den folgenden Szenarien auf:

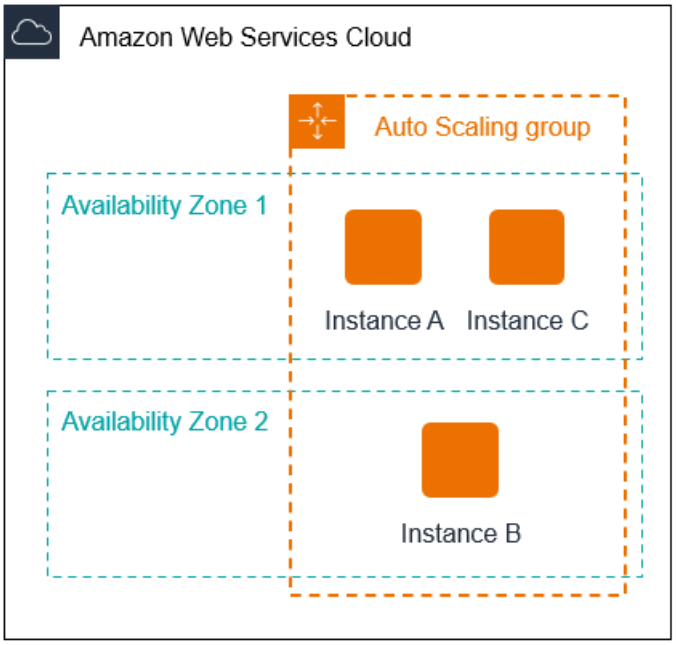
- Wenn dynamische Skalierungsrichtlinien verwendet werden, nimmt die Größe der Gruppe aufgrund von Änderungen des Werts einer Metrik ab
- Bei Verwendung der geplanten Skalierung nimmt die Größe der Gruppe infolge einer geplanten Aktion ab
- Wenn Sie die Gruppengröße manuel verkleinern

Das folgende Beispiel zeigt, wie Beendigungsrichtlinien funktionieren, wenn ein Scale-In-Ereignis vorliegt.

1. Die Auto-Scaling-Gruppe in diesem Beispiel hat einen Instance-Typ, zwei Availability Zones und eine gewünschte Kapazität von zwei Instances. Es verfügt auch über eine dynamische Skalierungsrichtlinie, die Instances hinzufügt und entfernt, wenn die Ressourcenauslastung zunimmt oder abnimmt. Die zwei Instances in dieser Gruppe sind auf die zwei Availability Zones verteilt, wie im folgenden Diagramm dargestellt.

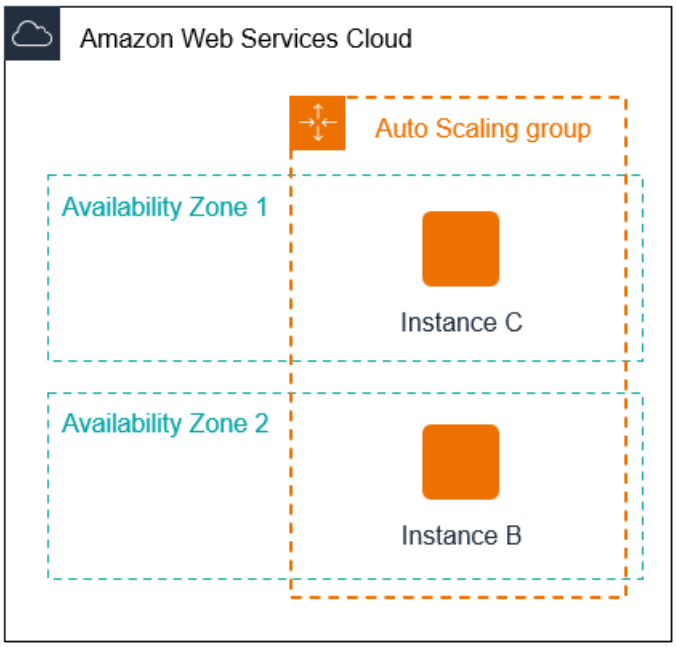


2. Wenn die Auto-Scaling-Gruppe skaliert wird, startet Amazon EC2 Auto Scaling eine neue Instance. Die Auto-Scaling-Gruppe verfügt nun über drei Instances, die auf die beiden Availability Zones verteilt sind, wie im folgenden Diagramm dargestellt.



3. Wenn die Auto-Scaling-Gruppe skaliert wird, beendet Amazon EC2 Auto Scaling eine der Instances.
4. Wenn Sie der Gruppe keine bestimmte Beendigungsrichtlinie zugewiesen haben, verwendet Amazon EC2 Auto Scaling die Standardbeendigungsrichtlinie. Sie wählt die Availability Zone mit zwei Instances aus und beendet die Instance, die über eine Startkonfiguration, eine andere

Startvorlage oder die älteste Version der aktuellen Startvorlage gestartet wurde. Wenn die Instances mit derselben Startvorlage und Version gestartet wurden, wählt Amazon EC2 Auto Scaling die Instance aus, die der nächsten Abrechnungsstunde am nächsten ist, und beendet sie.



Instance-Aktualisierung

Sie können eine Instance-Aktualisierung starten, um die Instances in Ihrer Auto Scaling Scaling-Gruppe zu aktualisieren. Während einer Instance-Aktualisierung beendet Amazon EC2 Auto Scaling Instances in der Gruppe und startet dann Ersetzungen für die beendeten Instances. Die Beendigungsrichtlinie für die Auto-Scaling-Gruppe steuert, welche Instances zuerst ersetzt werden.

Neuausgleich der Availability Zone

Amazon EC2 Auto Scaling gleicht Ihre Kapazität gleichmäßig über die Availability Zones aus, die für Ihre Auto-Scaling-Gruppe aktiviert sind. Dies trägt dazu bei, die Auswirkungen eines Ausfalls der Availability Zone zu reduzieren. Wenn die Verteilung der Kapazität über Availability Zones nicht ausgeglichen ist, gleicht Amazon EC2 Auto Scaling die Auto-Scaling-Gruppe neu aus, indem Instances in den aktivierten Availability Zones mit den wenigsten Instances gestartet und Instances an anderer Stelle beendet werden. Die Beendigungsrichtlinie steuert, welche Instances zuerst für die Beendigung priorisiert werden.

Es gibt eine Reihe von Gründen, warum die Verteilung von Instances über Availability Zones aus dem Gleichgewicht geraten kann.

Entfernen von Instances

Wenn Sie Instances von Ihrer Auto-Scaling-Gruppe trennen, setzen Sie Instances in den Standby-Modus oder beenden Instances explizit und verringern die gewünschte Kapazität, wodurch das Starten von Ersatz-Instances verhindert wird. Dadurch kann die Gruppe unausgewogen sein. Wenn dies auftritt, kann Amazon EC2 Auto Scaling dies kompensieren und das Gleichgewicht der Availability Zones wiederherstellen.

Verwenden anderer Availability Zones als ursprünglich angegeben

Wenn Sie Ihre Auto-Scaling-Gruppe erweitern, um zusätzliche Availability Zones einzuschließen oder Sie ändern, welche Availability Zones verwendet werden, startet Amazon EC2 Auto Scaling Instances in den neuen Availability Zones, und Instances in den anderen Zonen werden beendet, um sicherzustellen, dass sich Ihre Auto-Scaling-Gruppe gleichmäßig über Availability Zones erstreckt.

Ausfall der Verfügbarkeit

Verfügbarkeitsausfälle sind selten. Wenn jedoch eine Availability Zone nicht verfügbar ist und später wiederhergestellt wird, kann die Auto-Scaling-Gruppe zwischen Availability Zones unausgewogen sein. Amazon EC2 Auto Scaling versucht, die Gruppe schrittweise neu auszugleichen, und durch den Neuausgleich können Instances in anderen Zonen beendet werden.

Nehmen wir das Beispiel mit einer Auto-Scaling-Gruppe mit einem Instance-Typ, zwei Availability Zones und einer gewünschten Kapazität von zwei Instances. In einer Situation, in der eine Availability Zone fehlschlägt, startet Amazon EC2 Auto Scaling automatisch eine neue Instance in der fehlerfreien Availability Zone, um die Instance in der instabilen Availability Zone zu ersetzen. Wenn dann die instabile Availability Zone in einen fehlerfreien Zustand zurückkehrt, startet Amazon EC2 Auto Scaling automatisch eine neue Instance in dieser Zone, wodurch wiederum eine Instance in der nicht betroffenen Zone beendet wird.

> **Note**
>
> Beim Wiederherstellen des Gleichgewichts startet Amazon EC2 Auto Scaling vor dem Beenden der alten Instances neue, damit Leistung und Verfügbarkeit Ihrer Anwendung nicht beeinträchtigt werden.
> Da Amazon EC2 Auto Scaling vor dem Beenden der alten Instances versucht, neue zu starten, kann das Wiederherstellen des Gleichgewichts beeinträchtigt und sogar gänzlich unterbrochen werden, falls die angegebene maximale Kapazität nahezu oder gänzlich

erreicht ist. Um dieses Problem zu vermeiden, kann das System beim Wiederherstellen des Gleichgewichts die angegebene maximale Kapazität einer Gruppe vorübergehend um 10 % (oder um die Marge einer Instance, je nachdem, welcher Wert größer ist) überschreiten. Dieser Wert kann nur erhöht werden, wenn die Gruppe die maximale Kapazität erreicht hat oder kurz davor ist und das Wiederherstellen des Gleichgewichts aufgrund einer vom Benutzer angeforderten Neuverteilung der Availability Zones oder zum Kompensieren von Verfügbarkeitsproblemen der Availability Zones erforderlich ist. Die Kapazität wird nur für die Dauer der Wiederherstellung des Gleichgewichts in der Gruppe erhöht.

Kündigungsrichtlinien für Amazon EC2 Auto Scaling konfigurieren

Eine Kündigungsrichtlinie legt die Kriterien fest, nach denen Amazon EC2 Auto Scaling Instances in einer bestimmten Reihenfolge beendet.

Standardmäßig verwendet Amazon EC2 Auto Scaling eine Kündigungsrichtlinie, die darauf ausgelegt ist, zuerst Instances zu beenden, die veraltete Konfigurationen verwenden. Sie können die Kündigungsrichtlinie ändern, um zu kontrollieren, welche Instances am wichtigsten zuerst beendet werden müssen.

Wenn Amazon EC2 Auto Scaling Instances beendet, versucht es, das Gleichgewicht zwischen den Availability Zones aufrechtzuerhalten, die für Ihre Auto Scaling Scaling-Gruppe aktiviert sind. Die Aufrechterhaltung des zonalen Gleichgewichts hat Vorrang vor der Kündigungsrichtlinie. Wenn eine Availability Zone mehr Instances als andere hat, wendet Amazon EC2 Auto Scaling die Kündigungsrichtlinie zuerst auf die unausgeglichene Zone an. Wenn die Availability Zones ausgeglichen sind, wendet es die Kündigungsrichtlinie auf alle Zonen an.

Themen

So funktioniert die standardmäßige Kündigungsrichtlinie

Wenn Amazon EC2 Auto Scaling eine Instance beenden muss, identifiziert es zunächst, welche Availability Zone (oder Zonen) die meisten Instances und mindestens eine Instance hat, die nicht

vor einer Skalierung geschützt ist. Anschließend bewertet es ungeschützte Instances innerhalb der identifizierten Availability Zone wie folgt:

Instanzen, die veraltete Konfigurationen verwenden

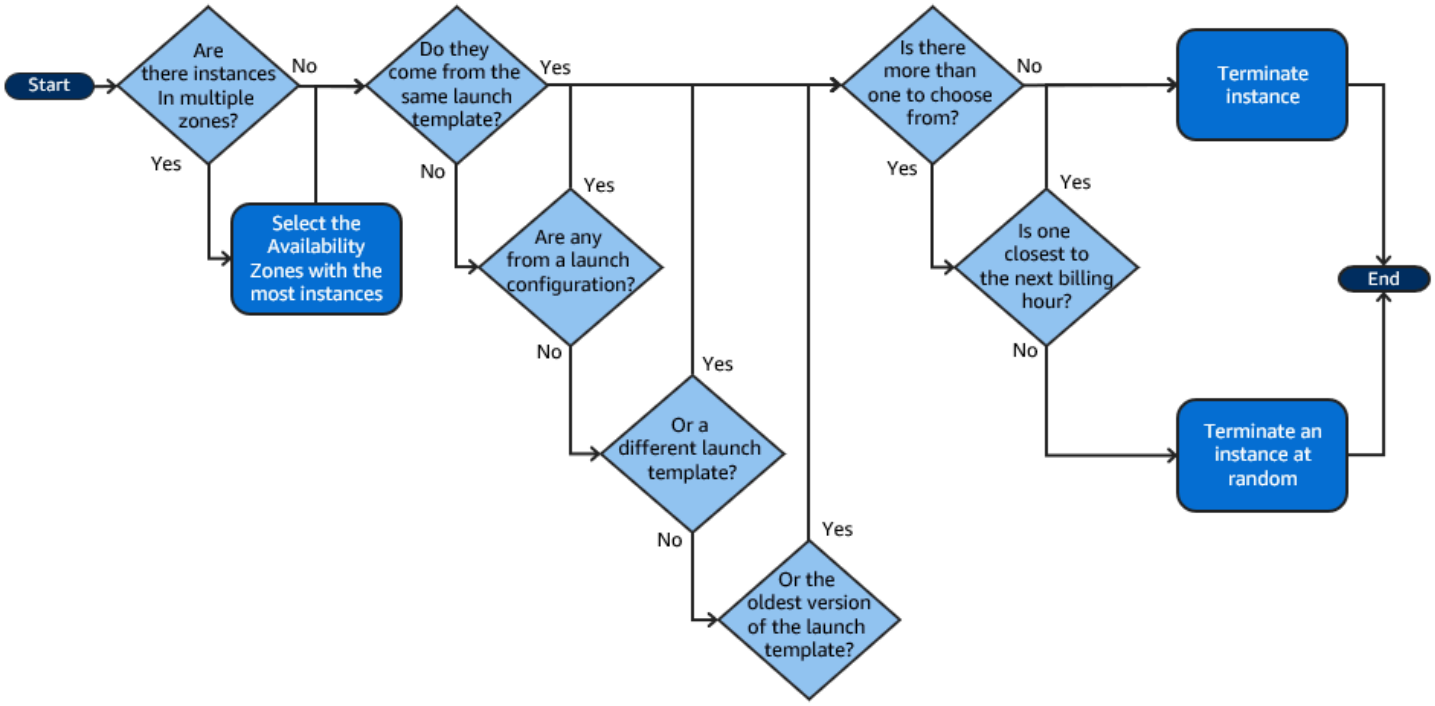
- Für Gruppen, die eine Startvorlage verwenden — Stellen Sie fest, ob eine der Instances veraltete Konfigurationen verwendet, und priorisieren Sie dabei in dieser Reihenfolge:
  1. Suchen Sie zunächst nach Instances, die mit einer Startkonfiguration gestartet wurden.
  2. Suchen Sie dann nach Instances, die mit einer anderen Startvorlage als mit der aktuellen Startvorlage gestartet wurden.
  3. Suchen Sie abschließend nach Instances, die die älteste Version der aktuellen Startvorlage verwenden.
- Für Gruppen, die eine Startkonfiguration verwenden — Stellen Sie fest, ob eine der Instances die älteste Startkonfiguration verwendet.

Wenn keine Instances mit veralteten Konfigurationen gefunden werden oder mehrere Instances zur Auswahl stehen, berücksichtigt Amazon EC2 Auto Scaling die nächsten Kriterien für Instances, die sich ihrer nächsten Abrechnungsstunde nähern.

Instances, die sich der nächsten Abrechnungsstunde nähern

Stellen Sie fest, ob eine der Instanzen, die die vorherigen Kriterien erfüllen, der nächsten Abrechnungsstunde am nächsten kommt. Wenn mehrere Instanzen gleich nah beieinander liegen, beenden Sie eine nach dem Zufallsprinzip. Auf diese Weise können Sie die Nutzung Ihrer Instances, die stündlich abgerechnet werden, maximieren. Der Großteil der EC2-Nutzung wird jetzt jedoch pro Sekunde abgerechnet, sodass diese Optimierung weniger Vorteile bietet. Weitere Informationen dazu finden Sie unter Amazon EC2 – Preise.

Das folgende Flussdiagramm zeigt, wie die standardmäßige Kündigungsrichtlinie für Gruppen funktioniert, die eine Startvorlage verwenden.



Standard-Beendigungsrichtlinie und Gruppe mit gemischten Instances

Amazon EC2 Auto Scaling wendet beim Beenden von Instances in gemischten Instance-Gruppen zusätzliche Kriterien an.

Wenn Amazon EC2 Auto Scaling eine Instance beenden muss, wird zunächst anhand der Gruppeneinstellungen ermittelt, welche Kaufoption (Spot oder On-Demand) beendet werden soll. Dadurch wird sichergestellt, dass sich die Gruppe im Laufe der Zeit dem angegebenen Verhältnis von Spot- und On-Demand-Instances nähert.

Anschließend wendet sie die Kündigungsrichtlinie unabhängig innerhalb jeder Availability Zone an. Sie bestimmt, welche Spot- oder On-Demand-Instance in welcher Availability Zone beendet werden soll, um die Availability Zones im Gleichgewicht zu halten. Dieselbe Logik gilt für eine gemischte Instance-Gruppe mit definierten Gewichtungen für die Instance-Typen.

In jeder Zone funktioniert die standardmäßige Kündigungsrichtlinie wie folgt, um zu bestimmen, welche ungeschützte Instance innerhalb der identifizierten Kaufoption gekündigt werden kann:

1. Ermitteln Sie, ob eine der Instances beendet werden kann, um die Abstimmung mit der angegebenen Zuweisungsstrategie für die Auto Scaling Scaling-Gruppe zu verbessern. Wenn keine Instanzen für die Optimierung identifiziert wurden oder mehrere Instanzen zur Auswahl stehen, wird die Bewertung fortgesetzt.

2. Stellen Sie fest, ob eine der Instanzen veraltete Konfigurationen verwendet, und priorisieren Sie dabei in dieser Reihenfolge:
   a. Suchen Sie zunächst nach Instances, die mit einer Startkonfiguration gestartet wurden.
   b. Suchen Sie dann nach Instances, die mit einer anderen Startvorlage als mit der aktuellen Startvorlage gestartet wurden.
   c. Suchen Sie abschließend nach Instances, die die älteste Version der aktuellen Startvorlage verwenden.

   Wenn keine Instanzen mit veralteten Konfigurationen gefunden werden oder mehrere Instanzen zur Auswahl stehen, wird die Bewertung fortgesetzt.
3. Stellen Sie fest, ob eine der Instanzen der nächsten Abrechnungsstunde am nächsten ist. Wenn mehrere Instanzen gleich nah beieinander liegen, wählen Sie nach dem Zufallsprinzip eine aus.

Vordefinierte Kündigungsrichtlinien

Sie wählen aus den folgenden vordefinierten Kündigungsrichtlinien:

- **`Default`**— Beenden Sie Instances gemäß der Standard-Kündigungsrichtlinie.
- **`AllocationStrategy`**— Beenden Sie Instances in der Auto Scaling Scaling-Gruppe, um die verbleibenden Instances an der Zuweisungsstrategie für den Instance-Typ auszurichten, der beendet wird (entweder eine Spot-Instance oder eine On-Demand-Instance). Diese Richtlinie ist nützlich, wenn Ihre bevorzugte Instance-Typen sich geändert haben. Wenn die Spot-Zuweisungsstrategie `lowest-price` lautet, können Sie Spot-Instances allmählich auf die N günstigsten Spot-Instance-Pools neu verteilen. Wenn die Spot-Zuweisungsstrategie `capacity-optimized` lautet, können Sie Spot-Instances allmählich auf mehrere Spot-Pools verteilen, in denen mehr Spot-Kapazität verfügbar ist. Sie können auch schrittweise On-Demand-Instances einer niedrigeren Priorität durch On-Demand-Instances einer höheren Priorität ersetzen.
- **`OldestLaunchTemplate`**— Beendet Instances mit der ältesten Startvorlage. Mit dieser Richtlinie werden Instances, die eine langfristige Startvorlage verwenden, zuerst beendet, gefolgt von Instances, die eine älteste Version der aktuellen Startvorlage verwenden. Diese Richtlinie ist nützlich, wenn Sie eine Gruppe aktualisieren und die Instances einer früheren Konfiguration auslaufen lassen.
- **`OldestLaunchConfiguration`**— Beendet Instances mit der ältesten Startkonfiguration. Diese Richtlinie ist nützlich, wenn Sie eine Gruppe aktualisieren und die Instances einer früheren Konfiguration auslaufen lassen. Mit dieser Richtlinie werden Instances, welche die nicht aktuelle Startkonfiguration verwenden, zuerst beendet.

- **`ClosestToNextInstanceHour`**— Beendet Instances, die der nächsten Abrechnungsstunde am nächsten sind. Diese Richtlinie hilft Ihnen, die Nutzung Ihrer Instances mit Stundengebühr zu maximieren.
- **`NewestInstance`**— Beendet die neueste Instanz in der Gruppe. Diese Richtlinie ist nützlich, wenn Sie eine neue Startkonfiguration testen, sie aber in der Produktionsumgebung nicht weiter verwenden möchten.
- **`OldestInstance`**— Beendet die älteste Instanz in der Gruppe. Diese Option ist nützlich, wenn Sie die Instances in der Auto-Scaling-Gruppe auf einen neuen EC2-Instance-Typ upgraden. Sie können nach und nach Instances des alten Typs durch Instances des neuen Typs ersetzen.

> **Note**
>
> Amazon EC2 Auto Scaling gleicht Instances immer zuerst über Availability Zones aus, unabhängig davon, welche Beendigungsrichtlinie verwendet wird. Daher können Situationen auftreten, in denen neuere Instances vor älteren Instances beendet werden. Beispielsweise, wenn eine kürzlich hinzugefügte Availability Zone vorhanden ist oder eine Availability Zone über mehr Instances verfügt als die anderen Availability Zones, die von der Gruppe verwendet werden.

Ändern Sie die Kündigungsrichtlinie für eine Auto Scaling Scaling-Gruppe

Verwenden Sie eine der folgenden Methoden, um die Kündigungsrichtlinie für Ihre Auto Scaling Scaling-Gruppe zu ändern.

Console

Sie können die Kündigungsrichtlinie nicht ändern, wenn Sie zum ersten Mal eine Auto Scaling Scaling-Gruppe in der Amazon EC2 Auto Scaling Scaling-Konsole erstellen. Die standardmäßige Beendigungsrichtlinie wird automatisch verwendet. Nachdem Ihre Auto Scaling Scaling-Gruppe erstellt wurde, können Sie die Standardrichtlinie durch eine andere Kündigungsrichtlinie oder durch mehrere Kündigungsrichtlinien ersetzen, die in der Reihenfolge aufgeführt sind, in der sie gelten sollen.

So ändern Sie die Kündigungsrichtlinie für eine Auto Scaling Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.

2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option Erweiterte Konfigurationen, Bearbeiten.
4. Wählen Sie für Beendigungsrichtlinien eine oder mehrere Beendigungsrichtlinien aus. Wenn Sie mehrere Richtlinien auswählen, geben Sie diese in der Reihenfolge an, in der sie ausgewertet werden sollen.

   Sie können optional die Option Benutzerdefinierte Beendigungsrichtlinie wählen und dann eine Lambda-Funktion auswählen, die Ihren Anforderungen entspricht. Wenn Sie Versionen und Aliase für Ihre Lambda-Funktion erstellt haben, können Sie eine Version oder einen Alias aus der Dropdown-Liste Version/Alias auswählen. Um die unveröffentlichte Version Ihrer Lambda-Funktion zu verwenden, lassen Sie Version/Alias auf den Standardwert eingestellt. Weitere Informationen finden Sie unter Eine benutzerdefinierte Beendigungsrichtlinie mit Lambda erstellen.

   > **Note**
   >
   > Wenn Sie mehrere Richtlinien verwenden, muss deren Reihenfolge korrekt festgelegt werden:
   >
   > - Wenn Sie die Standardrichtlinie verwenden, muss sie die letzte Richtlinie in der Liste sein.
   > - Wenn Sie eine Custom termination policy (benutzerdefinierte Beendigungsrichtlinie) verwenden, muss diese die erste Richtlinie in der Liste sein.
5. Wählen Sie Aktualisieren.

AWS CLI

Die Standardbeendigungsrichtlinie wird automatisch verwendet, es sei denn, es wird eine andere Richtlinie angegeben.

So ändern Sie die Kündigungsrichtlinie für eine Auto Scaling Scaling-Gruppe

Verwenden Sie einen der folgenden Befehle:

- create-auto-scaling-group
- update-auto-scaling-group

Sie können diese Beendigungsrichtlinien einzeln verwenden oder sie zu einer Liste von Richtlinien zusammenführen. Nutzen Sie z. B. den folgenden Befehl, um eine Auto-Scaling-Gruppe zu aktualisieren, damit sie zuerst die Richtlinie `OldestLaunchConfiguration` und dann die Richtlinie `ClosestToNextInstanceHour` verwendet.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --
termination-policies "OldestLaunchConfiguration" "ClosestToNextInstanceHour"
```

Falls Sie die `Default`-Beendigungsrichtlinie verwenden, setzen Sie sie ans Ende der Liste der Beendigungsrichtlinien. z. B. `--termination-policies "OldestLaunchConfiguration" "Default"`.

Um eine benutzerdefinierte Kündigungsrichtlinie zu verwenden, müssen Sie zunächst Ihre Kündigungsrichtlinie mithilfe von erstellen AWS Lambda. Um die Lambda-Funktion zur Verwendung als Beendigungsrichtlinie anzugeben, setzen Sie sie an die erste Stelle in der Liste der Beendigungsrichtlinien. z. B. `--termination-policies "arn:aws:lambda:us-west-2:123456789012:function:HelloFunction:prod" "OldestLaunchConfiguration"`. Weitere Informationen finden Sie unter Eine benutzerdefinierte Beendigungsrichtlinie mit Lambda erstellen.

Eine benutzerdefinierte Beendigungsrichtlinie mit Lambda erstellen

Amazon EC2 Auto Scaling verwendet Beendigungsrichtlinien, um zu priorisieren, welche Instances zuerst beendet werden sollen, wenn die Größe Ihrer Auto-Scaling-Gruppe verringert (abskaliert) wird. Ihre Auto-Scaling-Gruppe verwendet eine Standard-Beendigungsrichtlinie, Sie können jedoch optional eigene Beendigungsrichtlinien auswählen oder erstellen. Weitere Informationen zur Auswahl einer vordefinierten Beendigungsrichtlinie finden Sie unter Kündigungsrichtlinien für Amazon EC2 Auto Scaling konfigurieren.

In diesem Thema erfahren Sie, wie Sie eine benutzerdefinierte Beendigungsrichtlinie mithilfe einer AWS Lambda -Funktion erstellen, die Amazon EC2 Auto Scaling als Reaktion auf bestimmte Ereignisse aufruft. Die von Ihnen erstellte Lambda-Funktion verarbeitet die Informationen in den Eingabedaten, die von Amazon EC2 Auto Scaling gesendet werden, und gibt eine Liste von Instances zurück, die zum Beenden bereit sind.

Eine benutzerdefinierte Beendigungsrichtlinie bietet eine bessere Kontrolle darüber, welche Instances wann beendet werden. Wenn beispielsweise Ihre Auto-Scaling-Gruppe abskaliert, kann Amazon EC2

Auto Scaling nicht ermitteln, ob Workloads ausgeführt werden, die nicht unterbrochen werden sollten. Mit einer Lambda-Funktion können Sie die Beendigungsanforderung validieren und warten, bis der Workload abgeschlossen ist, bevor Sie die Instance-ID zur Beendigung an Amazon EC2 Auto Scaling zurückgeben.

Inhalt

Eingabedaten

Amazon EC2 Auto Scaling generiert eine JSON-Nutzlast für Scale-In-Ereignisse und tut dies auch dann, wenn Instances aufgrund der maximalen Instance-Lebensdauer oder Instance-Aktualisierungsfunktionen beendet werden. Außerdem wird eine JSON-Nutzlast für die Scale-In-Ereignisse generiert, die beim Neuausgleich Ihrer Gruppe über Availability Zones ausgelöst werden können.

Diese Nutzlast enthält Informationen über die Kapazität, die Amazon EC2 Auto Scaling beendet werden muss, eine Liste der Instances, die es für die Beendigung vorschlägt, und das Ereignis, das die Beendigung ausgelöst hat.

Es folgt ein Beispiel einer Nutzlast:

```
{
  "AutoScalingGroupARN": "arn:aws:autoscaling:us-east-1:<account-
id>:autoScalingGroup:d4738357-2d40-4038-ae7e-b00ae0227003:autoScalingGroupName/my-asg",
  "AutoScalingGroupName": "my-asg",
  "CapacityToTerminate": [
    {
      "AvailabilityZone": "us-east-1b",
      "Capacity": 2,
      "InstanceMarketOption": "on-demand"
    },
    {
      "AvailabilityZone": "us-east-1b",
```

```
      "Capacity": 1,
      "InstanceMarketOption": "spot"
    },
    {
      "AvailabilityZone": "us-east-1c",
      "Capacity": 3,
      "InstanceMarketOption": "on-demand"
    }
  ],
  "Instances": [
    {
      "AvailabilityZone": "us-east-1b",
      "InstanceId": "i-0056faf8da3e1f75d",
      "InstanceType": "t2.nano",
      "InstanceMarketOption": "on-demand"
    },
    {
      "AvailabilityZone": "us-east-1c",
      "InstanceId": "i-02e1c69383a3ed501",
      "InstanceType": "t2.nano",
      "InstanceMarketOption": "on-demand"
    },
    {
      "AvailabilityZone": "us-east-1c",
      "InstanceId": "i-036bc44b6092c01c7",
      "InstanceType": "t2.nano",
      "InstanceMarketOption": "on-demand"
    },
    ...
  ],
  "Cause": "SCALE_IN"
}
```

Die Nutzlast enthält den Namen der Auto-Scaling-Gruppe, ihren Amazon-Ressourcennamen (ARN) und die folgenden Elemente:

- `CapacityToTerminate` beschreibt, wie viel Ihrer Spot- oder On-Demand-Kapazität in einer bestimmten Availability Zone beendet wird.
- `Instances` stellt die Instances dar, die Amazon EC2 Auto Scaling basierend auf den Informationen in `CapacityToTerminate` vorschlägt.
- `Cause` beschreibt das Ereignis, das die Beendigung ausgelöst hat: `SCALE_IN`, `INSTANCE_REFRESH`, `MAX_INSTANCE_LIFETIME` oder `REBALANCE`.

In den folgenden Informationen werden die wichtigsten Faktoren erläutert, wie Amazon EC2 Auto Scaling die `InstancesIn` den Eingabedaten generiert:

- Die Aufrechterhaltung des Gleichgewichts über Availability Zones hinweg hat Vorrang, wenn eine Instance aufgrund von Scale-In-Ereignissen und Instance-Aktualisierungs-basierten Beendigungen beendet wird. Wenn daher eine Availability Zone über mehr Instances verfügt als die anderen Availability Zones, die von der Gruppe verwendet werden, umfassen die Eingabedaten nur Instances zur Beendigung, die aus der unausgewogenen Availability Zone stammen. Wenn die von der Gruppe verwendeten Availability Zones ausgeglichen sind, enthalten die Eingabedaten Instances aus allen Availability Zones der Gruppe.
- Wenn Sie eine Richtlinie für gemischte Instances verwenden, hat die Aufrechterhaltung Ihrer Spot- und On-Demand-Kapazitäten auf der Grundlage Ihrer gewünschten Prozentsätze für jede Kaufoption ebenfalls Vorrang. Wir identifizieren zunächst, welcher der beiden Typen (Spot oder On-Demand) beendet werden soll. Außerdem ermitteln wir dann, welche Instances (innerhalb der identifizierten Kaufoption) in welchen Availability Zones beendet werden sollen, was dazu führt, dass die Availability Zones am stärksten ausgeglichen sind.

Antwortdaten

Die Eingabedaten und Antwortdaten arbeiten zusammen, um die Liste der zu beendenden Instances einzugrenzen.

Mit der angegebenen Eingabe sollte die Antwort Ihrer Lambda-Funktion wie im folgenden Beispiel aussehen:

```
{
  "InstanceIDs": [
    "i-02e1c69383a3ed501",
    "i-036bc44b6092c01c7",
    ...
  ]
}
```

Die `InstanceIDs` in der Antwort stellen die Instances dar, die zum Beenden bereit sind.

Alternativ können Sie einen anderen Satz von Instances zurückgeben, die zum Beenden bereit sind, wodurch die Instances in den Eingabedaten außer Kraft gesetzt werden. Wenn beim Aufruf Ihrer Lambda-Funktion keine Instances beendet werden können, können Sie auch keine Instances zurückgeben.

Wenn keine Instances zum Beenden bereit sind, sollte die Antwort Ihrer Lambda-Funktion wie im folgenden Beispiel aussehen:

```
{
  "InstanceIDs": [ ]
}
```

Überlegungen

Beachten Sie die folgenden Überlegungen bei der Verwendung einer benutzerdefinierten Beendigungsrichtlinie:

- Wenn Sie eine Instance zuerst in den Antwortdaten zurückgeben, wird die Beendigung nicht garantiert. Wenn beim Aufruf Ihrer Lambda-Funktion mehr als die erforderliche Anzahl von Instances zurückgegeben wird, wertet Amazon EC2 Auto Scaling jede Instance anhand der anderen Beendigungsrichtlinien aus, die Sie für Ihre Auto-Scaling-Gruppe angegeben haben. Wenn mehrere Beendigungsrichtlinien vorhanden sind, wird versucht, die nächste Beendigungsrichtlinie in der Liste anzuwenden. Wenn mehr Instances vorhanden sind, als zum Beenden erforderlich sind, wird die nächste Beendigungsrichtlinie fortgesetzt usw. Wenn keine anderen Beendigungsrichtlinien angegeben sind, wird die Standardbeendigungsrichtlinie verwendet, um zu bestimmen, welche Instances beendet werden sollen.
- Wenn keine Instances zurückgegeben werden oder Ihre Lambda-Funktion eine Zeitüberschreitung auftritt, wartet Amazon EC2 Auto Scaling kurz, bevor Sie Ihre Funktion erneut aufrufen. Bei jedem Scale-In-Ereignis versucht es weiter, solange die gewünschte Kapazität der Gruppe geringer ist als die aktuelle Kapazität. Zum Beispiel versucht es bei Instance-Aktualisierungsbasierten Beendigungen eine Stunde lang. Wenn danach weiterhin Instances nicht beendet werden, schlägt der Instance-Aktualisierungsvorgang fehl. Bei maximaler Instance-Lebensdauer versucht Amazon EC2 Auto Scaling weiterhin, die Instance zu beenden, die als Überschreitung ihrer maximalen Lebensdauer identifiziert wird.
- Da Ihre Funktion wiederholt erneut versucht wird, stellen Sie sicher, dass Sie permanente Fehler im Code testen und beheben, bevor Sie eine Lambda-Funktion als benutzerdefinierte Beendigungsrichtlinie verwenden.
- Wenn Sie die Eingabedaten mit Ihrer eigenen Liste der zu beendenden Instances überschreiben und das Beenden dieser Instances die Availability Zones aus dem Gleichgewicht bringen, passt Amazon EC2 Auto Scaling die Kapazitätsverteilung über Availability Zones schrittweise neu aus. Zuerst ruft es Ihre Lambda-Funktion auf, um zu sehen, ob es Instances gibt, die beendet werden können, damit sie bestimmen kann, ob mit dem Neuausgleich begonnen werden soll. Wenn

Instances vorhanden sind, die beendet werden können, werden zuerst neue Instances gestartet. Wenn die Instances gestartet werden, wird dann erkannt, dass die aktuelle Kapazität Ihrer Gruppe höher ist als die gewünschte Kapazität, und ein Scale-In-Ereignis wird initiiert.

- Eine benutzerdefinierte Beendigungsrichtlinie beeinträchtigt nicht Ihre Fähigkeit, auch den Abskalierungsschutz zu verwenden, um bestimmte Instances vor dem Beenden zu schützen. Weitere Informationen finden Sie unter Instance-Abskalierungsschutz verwenden.

So erstellen Sie die Lambda-Funktion:

Erstellen Sie zunächst die Lambda-Funktion, damit Sie ihren Amazon-Ressourcennamen (ARN) in den Beendigungsrichtlinien für Ihre Auto-Scaling-Gruppe angeben können.

Erstellen einer Lambda-Funktion (Konsole)

1. Öffnen Sie die Funktions-Seite in der Lambda-Konsole.
2. Wählen Sie in der Navigationsleiste oben dieselbe Region aus, die Sie beim Erstellen der Auto-Scaling-Gruppe verwendet haben.
3. Wählen Sie Funktion erstellen und Von Grund auf neu erstellen aus.
4. Geben Sie unter Basic information (Grundlegende Informationen) bei Function name (Funktionsname) den Namen für Ihre Funktion ein.
5. Wählen Sie Funktion erstellen. Sie kehren zum Code und zur Konfiguration der Funktion zurück.
6. Wenn Ihre Funktion noch in der Konsole geöffnet ist, fügen Sie unter Funktionscode Ihren Code in den Editor ein.
7. Wählen Sie Bereitstellen.
8. Optional können Sie eine veröffentlichte Version der Lambda-Funktion erstellen, indem Sie die Registerkarte Versionen und dann Eine neue Version veröffentlichen auswählen. Weitere Informationen zur Versionierung in Lambda finden Sie unterVersionen der Lambda-Funktion im AWS Lambda -Entwicklerhandbuch.
9. Wenn Sie eine Version veröffentlichen möchten, wählen Sie die Registerkarte Aliasnamen aus, wenn Sie einen Alias mit dieser Version der Lambda-Funktion verbinden möchten. Weitere Informationen zu Aliassen in Lambda finden Sie unter Versionen der Lambda-Funktion im AWS Lambda -Entwicklerhandbuch.
10. Wählen Sie als Nächstes die Registerkarte Konfiguration und dann Berechtigungen aus.
11. Scrollen Sie nach unten bis zu Ressourcenbasierte Richtlinie und wählen Sie dann Hinzufügen von Berechtigungen aus. Eine ressourcenbasierte Richtlinie wird verwendet, um dem Prinzipal,

der in der Richtlinie angegeben ist, Berechtigungen zum Aufrufen Ihrer Funktion zu erteilen. In diesem Fall wird der Prinzipal die serviceverknüpfte Amazon EC2 Auto Scaling-Rolle sein, die der Auto-Scaling-Gruppe zugeordnet ist.

12. In der Richtlinienanweisung konfigurieren Sie Ihre Berechtigungen:

    a. Wählen Sie AWS-Konto.

    b. Für Prinzipal geben Sie den ARN der aufrufenden serviceverknüpften Rolle ein, z. B. **`arn:aws:iam::<aws-account-id>:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling`**.

    c. Wählen Sie für Aktion die Option lambda: InvokeFunction aus.

    d. Für Anweisungs-ID geben Sie eine eindeutige Anweisungs-ID ein, wie z. B. **`AllowInvokeByAutoScaling`**.

    e. Wählen Sie Speichern.

13. Nachdem Sie diese Anweisungen befolgt haben, fahren Sie als nächsten Schritt damit fort, den ARN Ihrer Funktion in den Beendigungsrichtlinien für Ihre Auto-Scaling-Gruppe anzugeben. Weitere Informationen finden Sie unter Ändern Sie die Kündigungsrichtlinie für eine Auto Scaling Scaling-Gruppe.

> **Note**
>
> Beispiele, die Sie als Referenz für die Entwicklung Ihrer Lambda-Funktion verwenden können, finden Sie im GitHub Repository für Amazon EC2 Auto Scaling.

Einschränkungen

- Sie können nur eine Lambda-Funktion in den Beendigungsrichtlinien für eine Auto-Scaling-Gruppe angeben. Wenn mehrere Beendigungsrichtlinien angegeben sind, muss zuerst die Lambda-Funktion angegeben werden.
- Sie können auf Ihre Lambda-Funktion verweisen, indem Sie entweder einen unqualifizierten ARN (ohne Suffix) oder einen qualifizierten ARN verwenden, der entweder eine Version oder einen Alias als Suffix hat. Wenn ein unqualifizierter ARN verwendet wird (z. B. `function:my-function`), muss die ressourcenbasierte Richtlinie für die unveröffentlichte Version Ihrer Funktion erstellt werden. Wenn ein unqualifizierter ARN verwendet wird (z. B. `function:my-function:1` oder

`function:my-function:prod`), muss die ressourcenbasierte Richtlinie für die spezifische veröffentlichte Version Ihrer Funktion erstellt werden.

- Sie können einen qualifizierten ARN nicht mit dem `$LATEST`-Suffix verwenden. Wenn Sie versuchen, eine benutzerdefinierte Beendigungsrichtlinie hinzuzufügen, die sich auf einen qualifizierten ARN mit dem `$LATEST`-Suffix bezieht, führt dies zu einem Fehler.
- Die Anzahl der in den Eingabedaten angegebenen Instances ist auf 30.000 Instances begrenzt. Wenn es mehr als 30.000 Instances gibt, die beendet werden könnten, nehmen die Eingabedaten `"HasMoreInstances": true` auf, umanzugehen, dass die maximale Anzahl von Instances zurückgegeben wird.
- Die maximale Laufzeit für Ihre Lambda-Funktion beträgt zwei Sekunden (2.000 Millisekunden). Als bewährte Methode sollten Sie den Zeitüberschreitungswert Ihrer Lambda-Funktion auf der Grundlage Ihrer erwarteten Laufzeit festlegen. Lambda-Funktionen haben eine Standard-Zeitüberschreitung von drei Sekunden, dies kann jedoch verringert werden.
- Wenn Ihre Laufzeit das 2-Sekunden-Limit überschreitet, wird jede Scale-In-Aktion angehalten, bis die Laufzeit unter diesen Schwellenwert fällt. Suchen Sie für Lambda-Funktionen mit durchweg längeren Laufzeiten nach einer Möglichkeit, die Laufzeit zu reduzieren, z. B. indem Sie die Ergebnisse zwischenspeichern, sodass sie bei nachfolgenden Lambda-Aufrufen abgerufen werden können.

Instance-Abskalierungsschutz verwenden

Mit dem Instance Scale-In Protection haben Sie die Kontrolle darüber, welche Instances Amazon EC2 Auto Scaling beenden kann. Ein häufiger Anwendungsfall für diese Funktion ist die Skalierung containerbasierter Workloads. Weitere Informationen finden Sie unter Entwerfen Sie Ihre Anwendungen auf Amazon EC2 Auto Scaling, um die Instance-Beendigung ordnungsgemäß zu handhaben.

Standardmäßig ist der Instanz-Scale-In-Schutz deaktiviert, wenn Sie eine Auto Scaling Scaling-Gruppe erstellen. Das bedeutet, dass Amazon EC2 Auto Scaling jede Instance in der Gruppe beenden kann.

Sie können Instances schützen, sobald sie gestartet werden, indem Sie die Instance-Abskalierungsschutz-Einstellung für Ihre Auto-Scaling-Gruppe aktivieren. Der Instance-Skalierungsschutz tritt in Kraft, sobald der Instance-Status `InService` lautet. Um dann zu kontrollieren, welche Instances beendet werden können, deaktivieren Sie die Abskalierungsschutz-

Einstellung für einzelne Instances innerhalb der Auto-Scaling-Gruppe. Auf diese Weise können Sie bestimmte Instances weiterhin vor dem ungewollten Beenden schützen.

Themen

Überlegungen

Bei der Verwendung von Instance Scale-In Protection sollten Sie Folgendes beachten:

- Wenn alle Instances in einer Auto-Scaling-Gruppe vor Scale-In geschützt sind und ein Scale-In-Ereignis eintritt, wird die gewünschte Kapazität verringert. Allerdings kann die Auto-Scaling-Gruppe die erforderliche Anzahl von Instances erst beenden, wenn der Instance-Skalierungsschutz deaktiviert wurde. In der AWS Management Console enthält der Aktivitätsverlauf für die Auto Scaling Scaling-Gruppe die folgende Meldung, wenn alle Instances in einer Auto Scaling Scaling-Gruppe vor dem Einskalieren geschützt sind, wenn ein Scale-In-Ereignis eintritt: `Could not scale to desired capacity because all remaining instances are protected from scale-in.`
- Wird eine Instance getrennt, die vor Abskalierung geschützt ist, verliert sie den Instance-Skalierungsschutz. Wird die Instance erneut der Gruppe zugewiesen, übernimmt sie die aktuelle Instance-Skalierungsschutzeinstellung der Gruppe. Wenn Amazon EC2 Auto Scaling eine neue Instanz startet oder eine Instanz aus einem warmen Pool in die Auto-Scaling-Gruppe verschiebt, erbt die Instanz die Einstellung für den Instance-Scale-in-Schutz der Auto-Scaling-Gruppe.
- Der Instance-Skalierungsschutz schützt die Auto-Scaling-Instances nicht vor Folgendem:
  - Ersetzung im Zuge von Zustandsprüfungen, falls die Instance Zustandsprüfungen nicht besteht. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.
  - Spot-Instance-Unterbrechungen Eine Spot-Instance wird beendet, wenn keine Kapazität mehr verfügbar ist oder der Spot-Preis Ihren Höchstpreis übersteigt.
  - Eine Kapazitätsblock-Reservierung endet. Amazon EC2 fordert die Capacity Block-Instances zurück, auch wenn sie vor Skalierung geschützt sind.

- Manuelles Beenden mit dem Befehl. `terminate-instance-in-auto-scaling-group` Weitere Informationen finden Sie unter Beenden einer Instance in Ihrer Auto-Scaling-Gruppe (AWS CLI).
- Manuelles Beenden über die Amazon EC2 EC2-Konsole, CLI-Befehle und API-Operationen. Aktivieren Sie den Amazon EC2-Beendigungsschutz, um Auto-Scaling-Instances vor manueller Beendigung zu schützen. (Dies verhindert nicht, dass Amazon EC2 Auto Scaling Instances beendet oder manuell über den `terminate-instance-in-auto-scaling-group` Befehl beendet.) Informationen zur Aktivierung des Amazon EC2 EC2-Kündigungsschutzes in einer Startvorlage finden Sie unterErstellen einer Startvorlage mithilfe erweiterter Einstellungen.

Ändern Sie den Scale-In-Schutz für eine Auto Scaling Scaling-Gruppe

Sie können den Instance-Skalierungsschutz für eine Auto-Scaling-Gruppe aktivieren oder deaktivieren. Wenn Sie ihn aktivieren, ist für alle neuen Instances, die von der Gruppe gestartet werden, der Instanz-Scale-In-Schutz aktiviert.

Das Aktivieren oder Deaktivieren dieser Einstellung für eine Auto Scaling Scaling-Gruppe hat keine Auswirkungen auf bestehende Instances.

Console

So aktivieren Sie den Scale-In-Schutz für eine neue Auto Scaling Scaling-Gruppe

Wenn Sie die Auto Scaling Scaling-Gruppe erstellen, aktivieren Sie auf der Seite Gruppengröße und Skalierungsrichtlinien konfigurieren unter Instance Scale-In Protection das Kontrollkästchen Instance Scale-In Protection aktivieren.

Um den Scale-in-Schutz für eine bestehende Gruppe zu aktivieren oder zu deaktivieren

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option Erweiterte Konfigurationen, Bearbeiten.
4. Aktivieren oder deaktivieren Sie für Instance Scale-In Protection das Kontrollkästchen Instance Scale-In Protection aktivieren, um diese Option nach Bedarf zu aktivieren oder zu deaktivieren.

5. Wählen Sie Aktualisieren.

AWS CLI

So aktivieren Sie den Scale-In-Schutz für eine neue Auto Scaling Scaling-Gruppe

Verwenden Sie den folgenden create-auto-scaling-group-Befehl, um den Skalierungsschutz-Instance zu aktivieren:

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --new-instances-protected-from-scale-in ...
```

Um den Scale-In-Schutz für eine bestehende Gruppe zu aktivieren

Verwenden Sie den folgenden update-auto-scaling-group-Befehl, um den Instance-Skalierungsschutz einer bestimmten Auto-Scaling-Gruppe zu aktivieren.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --new-instances-protected-from-scale-in
```

Um den Scale-In-Schutz für eine bestehende Gruppe zu deaktivieren

Verwenden Sie den folgenden Befehl, um den Instance-Skalierungsschutz für die angegebene Gruppe zu deaktivieren:

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --no-new-instances-protected-from-scale-in
```

Ändern Sie den Scale-In-Schutz für eine Instanz

Standardmäßig übernehmen Instances die Instance-Skalierungsschutzeinstellung der Auto-Scaling-Gruppe, der sie angehören. Sie können den Instanz-Scale-In-Schutz jedoch für einzelne Instances nach deren Start aktivieren oder deaktivieren.

Console

Um den Scale-in-Schutz für eine Instance zu aktivieren oder zu deaktivieren

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.

2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Instance management (Instance-Verwaltung) unter Instances eine Instance aus.
4. Um den Instance-Skalierungsschutz zu aktivieren, wählen Sie Actions (Aktionen), Set Scale In Protection (Skalierungsschutz festlegen) aus. Wählen Sie nach Aufforderung Set Scale In Protection (Skalierungsschutz einrichten) aus.
5. Um den Instance-Abwärtsskalierungsschutz zu deaktivieren, wählen Sie Actions (Aktionen), Remove Scale In Protection (Skalierungsschutz entfernen) aus. Wählen Sie nach Aufforderung Remove Scale In Protection (Skalierungsschutz entfernen) aus.

AWS CLI

Um den Scale-In-Schutz für eine Instance zu aktivieren

Verwenden Sie den folgenden set-instance-protection-Befehl, um den Skalierungsschutz der angegebenen Instance zu aktivieren:

```
aws autoscaling set-instance-protection --instance-ids i-5f2e8a0d --auto-scaling-
group-name my-asg --protected-from-scale-in
```

Um den Scale-In-Schutz für eine Instance zu deaktivieren

Verwenden Sie den folgenden Befehl, um den Instance-Skalierungsschutz der angegebenen Instance zu deaktivieren:

```
aws autoscaling set-instance-protection --instance-ids i-5f2e8a0d --auto-scaling-
group-name my-asg --no-protected-from-scale-in
```

> **Note**
>
> Denken Sie daran, dass der Instance Scale-In Protection nicht garantiert, dass Instances im Falle eines menschlichen Fehlers nicht beendet werden, z. B. wenn jemand eine Instance manuell über die Amazon EC2 EC2-Konsole beendet oder. AWS CLI Wenn Sie Ihre Instance davor schützen möchten, dass sie versehentlich beendet wird, verwenden Sie den Amazon EC2-Beendigungsschutz. Selbst bei aktiviertem Beendigungsschutz und

Instance-Scale-In-Schutz können Daten, die im Instance-Speicher gespeichert werden, verloren gehen, wenn eine Zustandsprüfung feststellt, dass eine Instance fehlerhaft ist oder wenn die Gruppe selbst versehentlich gelöscht wurde. Wie bei jeder Umgebung ist es eine bewährte Vorgehensweise, Ihre Daten häufig zu sichern bzw. zu für Ihre Business Continuity-Anforderungen geeigneten Intervallen.

Entwerfen Sie Ihre Anwendungen auf Amazon EC2 Auto Scaling, um die Instance-Beendigung ordnungsgemäß zu handhaben

Dieses Thema befasst sich mit den verschiedenen Ansätzen, die Sie verfolgen können, wenn Sie Anwendungen auf Instances laufen haben, die idealerweise nicht unerwartet beendet werden sollten, sobald Amazon EC2 Auto Scaling auf ein Abskalierungsereignis reagiert.

Nehmen wir zum Beispiel an, Sie haben eine Amazon-SQS-Warteschlange, die eingehende Nachrichten für Aufträge mit langer Laufzeit sammelt. Wenn eine neue Nachricht eingeht, ruft eine Instanz in der Auto-Scaling-Gruppe die Nachricht ab und beginnt mit der Verarbeitung. Die Verarbeitung jeder Nachricht dauert 3 Stunden. Wenn die Anzahl der Nachrichten zunimmt, werden der Auto-Scaling-Gruppe automatisch neue Instances hinzugefügt. Wenn die Anzahl der Nachrichten abnimmt, werden vorhandene Instanzen automatisch beendet. In diesem Fall muss Amazon EC2 Auto Scaling entscheiden, welche Instance beendet werden soll. Standardmäßig ist es möglich, dass Amazon EC2 Auto Scaling eine Instance beendet, die seit 2,9 Stunden mit der Verarbeitung eines dreistündigen Jobs beschäftigt ist, und nicht eine Instance, die sich derzeit im Leerlauf befindet. Um Probleme mit unerwarteten Beendigungen bei der Verwendung von Amazon EC2 Auto Scaling zu vermeiden, müssen Sie Ihre Anwendung so gestalten, dass sie auf dieses Szenario reagiert.

Sie können die folgenden Funktionen verwenden, um zu verhindern, dass Ihre Auto-Scaling-Gruppe Instances beendet, die noch nicht zum Beenden bereit sind, oder dass Instances schneller beendet werden, als sie ihre zugewiesenen Jobs erledigen können. Alle drei Funktionen können in Kombination oder einzeln verwendet werden.

Inhalt

Important

Wenn Sie Ihre Anwendungen auf Amazon EC2 Auto Scaling entwerfen, um die Instance-Beendigung ordentlich zu handhaben, denken Sie an diese Punkte.

- Wenn eine Instance fehlerhaft ist, ersetzt Amazon EC2 Auto Scaling sie unabhängig davon, welches Feature Sie verwenden (es sei denn, Sie unterbrechen den `ReplaceUnhealthy`-Prozess). Sie können einen Lebenszyklus-Hook verwenden, um es der Anwendung zu ermöglichen, ordnungsgemäß herunterzufahren oder alle Daten zu kopieren, die Sie wiederherstellen müssen, bevor die Instance beendet wird.
- Es kann nicht garantiert werden, dass ein Beendigungs-Lebenszyklus-Hook ausgeführt oder beendet wird, bevor eine Instance beendet wird. Wenn etwas fehlschlägt, beendet Amazon EC2 Auto Scaling die Instance trotzdem.

Instance-Abskalierungsschutz

Sie können den Instance-Abskalierungsschutz in vielen Situationen verwenden, in denen das Beenden von Instances eine kritische Aktion ist, die standardmäßig verweigert und nur für bestimmte Instances ausdrücklich zugelassen werden sollte. Bei der Ausführung containerisierter Workloads ist es beispielsweise üblich, alle Instances zu schützen und den Schutz nur für Instances aufzuheben, die keine aktuellen oder geplanten Aufgaben haben. Dienste wie Amazon ECS haben den Instance-Abskalierungsschutz in ihre Produkte integriert.

Sie können den Abskalierungsschutz in der Auto-Scaling-Gruppe aktivieren, um den Abskalierungsschutz auf Instances anzuwenden, wenn diese erstellt werden, und ihn für vorhandene Instances zu aktivieren. Wenn eine Instance keine Arbeit mehr zu erledigen hat, kann sie den Schutz ausschalten. Die Instance kann weiterhin nach neuen Jobs suchen und den Schutz wieder aktivieren, wenn neue Jobs zugewiesen werden.

Anwendungen können den Schutz entweder von einer zentralen Steuerebene aus einrichten, die verwaltet, ob eine Instance beendet werden kann oder nicht, oder von den Instances selbst aus. Bei einer großen Flotte kann es jedoch zu Drosselungsproblemen kommen, wenn eine große Anzahl von Instances ihren Abskalierungsschutz ständig umschaltet.

Weitere Informationen finden Sie unter Instance-Abskalierungsschutz verwenden.

Benutzerdefinierte Beendigungsrichtlinie

Wie beim Instance-Abskalierungsschutz können Sie mit einer benutzerdefinierten Beendigungsrichtlinie verhindern, dass Ihre Auto-Scaling-Gruppe bestimmte Instances beendet.

Standardmäßig verwendet Ihre Auto-Scaling-Gruppe eine Standardbeendigungs-Richtlinie, um zu bestimmen, welche Instances zuerst beendet werden. Wenn Sie mehr Kontrolle darüber haben möchten, welche Instances zuerst beendet werden, können Sie mithilfe einer Lambda-Funktion Ihre eigene benutzerdefinierte Beendigungsrichtlinie implementieren. Amazon EC2 Auto Scaling ruft die Funktion immer dann auf, wenn entschieden werden muss, welche Instance beendet werden soll. Es wird nur eine Instance beendet, die von der Funktion zurückgegeben wird. Wenn die Funktion fehlerhaft ist, ein Timeout auftritt oder eine leere Liste erzeugt wird, beendet Amazon EC2 Auto Scaling keine Instances.

Eine benutzerdefinierte Beendigungsrichtlinie ist nützlich, wenn bekannt ist, wann eine Instance ausreichend redundant oder nicht ausgelastet ist, sodass sie beendet werden kann. Um dies zu unterstützen, müssen Sie Ihre Anwendung mit einer Steuerebene implementieren, die die Workload in der gesamten Gruppe überwacht. Auf diese Weise weiß die Lambda-Funktion, dass sie eine Instance, die immer noch Jobs verarbeitet, nicht einbeziehen soll.

Weitere Informationen finden Sie unter Eine benutzerdefinierte Beendigungsrichtlinie mit Lambda erstellen.

Beendigungs-Lebenszyklus-Hooks

Ein Beendigungs-Lebenszyklus-Hook verlängert die Lebensdauer einer Instance, die bereits für die Beendigung ausgewählt wurde. Er bietet zusätzliche Zeit, um alle Nachrichten oder Anfragen zu bearbeiten, die der Instance derzeit zugewiesen sind, oder um den Fortschritt zu speichern und die Arbeit auf eine andere Instance zu übertragen.

Bei vielen Workloads kann ein Lebenszyklus-Hook ausreichen, um eine Anwendung auf einer Instance, die zur Beendigung ausgewählt wurde, ordnungsgemäß herunterzufahren. Dies ist ein Best-Effort-Ansatz und kann nicht verwendet werden, um eine Beendigung im Falle eines Fehlers zu verhindern.

Um einen Lebenszyklus-Hook verwenden zu können, müssen Sie wissen, wann eine Instance zum Beenden ausgewählt wurde. Sie haben zwei Möglichkeiten, dies herauszufinden:

Option	Beschreibung	Am besten verwendet für	Link zur Dokumentation
Innerhalb der Instance	Der Instance Metadata Service (IMDS) ist ein sicherer Endpunkt, bei dem Sie direkt von der Instance aus den Status einer Instance abfragen können. Wenn die Metadaten mit `Terminated` zurückgegeben werden, ist die Beendigung Ihrer Instance geplant.	Anwendungen, bei denen Sie eine Aktion auf der Instance ausführen müssen, bevor die Instance beendet wird.	Abrufen des Zielleben szyklus-S tatus
Außerhalb der Instance	Wenn eine Instance beendet wird, wird eine Ereignisb enachrichtigung generiert. Sie können Regeln mithilfe von Amazon EventBridge, Amazon SQS oder Amazon SNS erstellen, um diese Ereignisse zu erfassen, und eine Antwort aufrufen, z. B. mit einer Lambda-Funktion.	Anwendungen, die außerhalb der Instance Aktionen durchführen müssen.	Konfiguri eren eines Benachric htigungsziels

Um einen Lebenszyklus-Hook verwenden zu können, müssen Sie auch wissen, wann eine Instance für die vollständige Beendigung bereit ist. Amazon EC2 Auto Scaling weist Amazon EC2 nicht an, die Instance zu beenden, bis sie einen CompleteLifecycleAktionsaufruf erhält oder das Timeout abgelaufen ist, je nachdem, was zuerst eintritt.

Standardmäßig kann eine Instance aufgrund eines Beendigungs-Lebenszyklus-Hook eine Stunde lang weiterlaufen (Heartbeat-Timeout). Sie können den standardmäßigen Timeout konfigurieren, falls eine Stunde nicht ausreicht, um die Lebenszyklus-Aktion abzuschließen. Wenn tatsächlich eine Lebenszyklusaktion ausgeführt wird, können Sie das Timeout mit API-Aufrufen verlängern. RecordLifecycleActionHeartbeat

Weitere Informationen finden Sie unter Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling.

Amazon EC2 Auto Scaling Scaling-Prozesse aussetzen und fortsetzen

In diesem Thema wird beschrieben, wie Sie einen oder mehrere Prozesse für Ihre Auto Scaling Scaling-Gruppe aussetzen und dann wieder aufnehmen können, um bestimmte Vorgänge vorübergehend zu deaktivieren.

Das Aussetzen von Prozessen kann nützlich sein, wenn Sie ein Problem untersuchen oder beheben müssen, ohne dass es zu Störungen durch Skalierungsrichtlinien oder geplante Aktionen kommt. Es verhindert auch, dass Amazon EC2 Auto Scaling Instances als fehlerhaft markiert und ersetzt, während Sie Änderungen an Ihrer Auto Scaling Scaling-Gruppe vornehmen.

Themen

> **Note**
>
> Zusätzlich zu den Unterbrechungen, die Sie initiieren, kann Amazon EC2 Auto Scaling auch Prozesse für Auto-Scaling-Gruppen unterbrechen, für die wiederholt keine Instances gestartet werden können. Dies wird als administrative Unterbrechung bezeichnet. Eine administrative Unterbrechung wird meistens für Auto-Scaling-Gruppen verwendet, die seit über 24 Stunden ohne Erfolg versuchen, Instances zu starten. Sie können Prozesse fortsetzen, die aus administrativen Gründen von Amazon EC2 Auto Scaling unterbrochen wurden.

Arten von Prozessen

Die Anhalten-Fortsetzen-Funktion unterstützt die folgenden Prozesse:

- `Launch`— Fügt Instances zur Auto Scaling-Gruppe hinzu, wenn die Gruppe horizontal skaliert wird oder wenn Amazon EC2 Auto Scaling Instances aus anderen Gründen startet, z. B. wenn Instances zu einem warmen Pool hinzugefügt werden.
- `Terminate`— Entfernt Instances aus der Auto Scaling-Gruppe, wenn die Gruppe skaliert wird oder wenn Amazon EC2 Auto Scaling beschließt, Instances aus anderen Gründen zu beenden, z. B. wenn eine Instance wegen Überschreitung ihrer maximalen Lebensdauer beendet wird oder wenn eine Zustandsprüfung nicht bestanden hat.
- `AddToLoadBalancer`— Fügt Instances der angehängten Load Balancer-Zielgruppe oder dem Classic Load Balancer hinzu, wenn sie gestartet werden. Weitere Informationen finden Sie unter Um den Datenverkehr über die Instances in Ihrer Auto-Scaling-Gruppe zu verteilen, verwenden Sie Elastic-Load-Balancing..
- `AlarmNotification`— Akzeptiert Benachrichtigungen von CloudWatch Alarmen, die mit dynamischen Skalierungsrichtlinien verknüpft sind. Weitere Informationen finden Sie unter Dynamische Skalierung für Amazon EC2 Auto Scaling.
- `AZRebalance`— Verteilt die Anzahl der EC2-Instances in der Gruppe gleichmäßig auf alle angegebenen Availability Zones, wenn die Gruppe aus dem Gleichgewicht gerät, z. B. wenn eine zuvor nicht verfügbare Availability Zone wieder in einen fehlerfreien Zustand zurückkehrt. Weitere Informationen finden Sie unter Wiederherstellen des Gleichgewichts von Aktivitäten.
- `HealthCheck`— Überprüft den Zustand der Instances und markiert eine Instance als fehlerhaft, wenn Amazon EC2 oder Elastic Load Balancing Amazon EC2 Auto Scaling mitteilt, dass die Instance fehlerhaft ist. Dieser Vorgang kann den manuell festgelegten Zustand einer Instance außer Kraft setzen. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.
- `InstanceRefresh`— Beendet und ersetzt Instances mithilfe der Instance-Aktualisierungsfunktion. Weitere Informationen finden Sie unter Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren.
- `ReplaceUnhealthy`— Beendet Instanzen, die als fehlerhaft markiert sind, und erstellt dann neue Instanzen, um sie zu ersetzen. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.
- `ScheduledActions`— Führt die geplanten Skalierungsaktionen aus, die Sie erstellen oder die für Sie erstellt werden, wenn Sie einen AWS Auto Scaling Skalierungsplan erstellen und die prädiktive Skalierung aktivieren. Weitere Informationen finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling.

Überlegungen

Berücksichtigen Sie Folgendes, bevor Sie Prozesse anhalten:

- Durch das Aussetzen `AlarmNotification` können Sie die Zielverfolgungs-, Step- und Simple Scaling-Richtlinien der Gruppe vorübergehend beenden, ohne die Skalierungsrichtlinien oder die zugehörigen CloudWatch Alarme zu löschen. Informationen zum vorübergehenden Beenden einzelner Skalierungsrichtlinien finden Sie unter Eine Skalierungsrichtlinie für eine Auto-Scaling-Gruppe deaktivieren.
- Sie können sich dafür entscheiden, die `HealthCheck` `ReplaceUnhealthy` Prozesse zum Neustarten von Instances auszusetzen, ohne dass Amazon EC2 Auto Scaling die Instances aufgrund seiner Zustandsprüfungen beendet. Wenn Sie jedoch Amazon EC2 Auto Scaling benötigen, um weiterhin Zustandsprüfungen für die verbleibenden Instances durchzuführen, verwenden Sie stattdessen die Standby-Funktion. Weitere Informationen finden Sie unter Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe.
- Wenn Sie die Prozesse `Launch` und `Terminate` oder `AZRebalance` aussetzen und dann Änderungen an Ihrer Auto-Scaling-Gruppe vornehmen, zum Beispiel indem Sie Instances abziehen oder die angegebenen Availability Zones ändern, kann Ihre Gruppe zwischen den Availability Zones unausgewogen werden. Wenn das vorkommt, verteilt Amazon EC2 Auto Scaling nach dem Fortsetzen der angehaltenen Prozesse die Instances schrittweise gleichmäßig auf die Availability Zones.
- Wenn Sie den `Terminate` Prozess unterbrechen, können Sie trotzdem die Beendigung von Instances erzwingen, indem Sie den Befehl delete-auto-scaling-group mit der Option force delete verwenden.
- Das Anhalten des `Terminate` Prozesses gilt nur für Instanzen, die sich derzeit im Status befinden. `InService` Es verhindert nicht die Beendigung von Instances in anderen Staaten, z. B., oder von Instanzen`Pending`, die nicht ordnungsgemäß aus dem Standby-Modus wieder aufgenommen werden können.
- Der `RemoveFromLoadBalancerLowPriority` Prozess kann ignoriert werden, wenn er in Aufrufen zur Beschreibung von Auto Scaling Scaling-Gruppen mit den AWS CLI oder SDKs vorkommt. Dieser Prozess ist veraltet und wird nur aus Gründen der Abwärtskompatibilität beibehalten.

Prozess anhalten

Verwenden Sie eine der folgenden Methoden, um einen Prozess für eine Auto Scaling Scaling-Gruppe anzuhalten:

Console

Setzen Sie einen Prozess wie folgt aus:

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option Erweiterte Konfigurationen, Bearbeiten.
4. Wählen Sie für Suspended Processes (Unterbrochene Prozesse) den zu unterbrechenden Prozess aus.
5. Wählen Sie Aktualisieren.

AWS CLI

Verwenden Sie den folgenden suspend-processes-Befehl, um einzelne Prozesse anzuhalten.

```
aws autoscaling suspend-processes --auto-scaling-group-name my-asg --scaling-
processes HealthCheck ReplaceUnhealthy
```

Um alle Prozesse anzuhalten, lassen Sie die `--scaling-processes`-Option wie folgt aus.

```
aws autoscaling suspend-processes --auto-scaling-group-name my-asg
```

Prozesse fortsetzen

Verwenden Sie eine der folgenden Methoden, um einen unterbrochenen Prozess für eine Auto Scaling Scaling-Gruppe wieder aufzunehmen:

Console

Setzen Sie einen ausgesetzten Prozess wie folgt fort:

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option Erweiterte Konfigurationen, Bearbeiten.
4. Entfernen Sie für Suspended Processes (Unterbrochene Prozesse) den unterbrochenen Prozess.
5. Wählen Sie Aktualisieren.

AWS CLI

Verwenden Sie den folgenden Befehl resume-processes, um einen unterbrochenen Prozess wieder aufzunehmen.

```
aws autoscaling resume-processes --auto-scaling-group-name my-asg --scaling-
processes HealthCheck
```

Um alle Prozesse fortzusetzen, lassen Sie die `--scaling-processes`-Option wie folgt aus.

```
aws autoscaling resume-processes --auto-scaling-group-name my-asg
```

Wie sich unterbrochene Prozesse auf andere Prozesse auswirken

In den folgenden Abschnitten wird beschrieben, was passiert, wenn verschiedene Prozesse einzeln angehalten werden.

Themen

- Launchist suspendiert
- Terminateist suspendiert
- AddToLoadBalancerist suspendiert
- AlarmNotificationist suspendiert

Launchist suspendiert

- `AlarmNotification` ist immer noch aktiv, aber Ihre Auto-Scaling-Gruppe kann keine Aufskalierungs-Aktivitäten für Alarme initiieren, die verletzt sind.
- `ScheduledActions` ist aktiv, aber Ihre Auto-Scaling-Gruppe kann keine Aufskalierungs-Aktivitäten für geplante Aktionen initiieren, die auftreten.
- `AZRebalance` hört auf, die Gruppe auszugleichen.
- `ReplaceUnhealthy` beendet weiterhin fehlerhafte Instances, startet jedoch keine Ersetzungen. Wenn Sie den Prozess `Launch` wieder aufnehmen, ersetzt Amazon EC2 Auto Scaling sofort alle Instances, die es während der Zeit, in der `Launch` ausgesetzt war, beendet hat.
- `InstanceRefresh` ersetzt keine Instances.

Terminateist suspendiert

- `AlarmNotification` ist immer noch aktiv, aber Ihre Auto-Scaling-Gruppe kann keine Abskalierungs-Aktivitäten für Alarme initiieren, die verletzt sind.
- `ScheduledActions` ist aktiv, aber Ihre Auto-Scaling-Gruppe kann keine Abskalierungs-Aktivitäten für geplante Aktionen initiieren, die auftreten.
- `AZRebalance` ist noch aktiv, funktioniert jedoch nicht ordnungsgemäß. Es können neue Instances gestartet werden, ohne dass alte beendet werden. Dies kann dazu führen, dass Ihre Auto-Scaling-Gruppe auf eine Größe anwächst, die ihre Höchstgröße um bis zu 10 Prozent übersteigt, da dies während Aktivitäten zur Wiederherstellung des Gleichgewichts vorübergehend zulässig ist. Ihre Auto-Scaling-Gruppe könnte diese die Höchstgröße überschreitende Größe beibehalten, bis Sie den `Terminate`-Prozess fortsetzen.
- `ReplaceUnhealthy` ist inaktiv, jedoch nicht `HealthCheck`. Wenn `Terminate` fortgesetzt wird, wird der Prozess `ReplaceUnhealthy` sofort ausgeführt. Wenn Instances als fehlerhaft markiert wurden, während `Terminate` unterbrochen war, werden sie sofort ersetzt.

- `InstanceRefresh` ersetzt keine Instances.

`AddToLoadBalancer`ist suspendiert

- Amazon EC2 Auto Scaling startet die Instances, fügt sie aber nicht zur Load Balancer-Zielgruppe oder Classic Load Balancer hinzu. Wird der Prozess `AddToLoadBalancer` fortgesetzt, fügt es Instances beim Start wieder zum Load Balancer hinzu. Allerdings fügt es keine Instances hinzu, die gestartet wurden, als der Prozess ausgesetzt war. Diese Instances müssen Sie manuell anmelden.

`AlarmNotification`ist suspendiert

- Amazon EC2 Auto Scaling ruft keine Skalierungsrichtlinien auf, wenn ein CloudWatch Alarmschwellenwert überschritten wird. Wenn Sie `AlarmNotification` fortsetzen, berücksichtigt Amazon EC2 Auto Scaling Richtlinien mit Alarmschwellenwerten aus, die derzeit überschritten werden.

`AZRebalance`ist suspendiert

- Amazon EC2 Auto Scaling versucht nicht, Instances nach bestimmten Ereignissen neu zu verteilen. Wenn jedoch ein Ereignis zu einer horizontalen Skalierung nach oben oder nach unten eintritt, versucht der Prozess weiterhin, die Availability Zones auszugleichen. Beispielsweise startet er bei einer horizontalen Skalierung nach oben die Instance in der Availability Zone mit den wenigsten Instances. Wenn die Gruppe aus dem Gleichgewicht gerät, während `AZRebalance` unterbrochen ist, und Sie dies fortsetzen, versucht Amazon EC2 Auto Scaling, die Gruppe wieder auszugleichen. Zuerst wird `Launch` und anschließend `Terminate` aufgerufen.

`HealthCheck`ist suspendiert

- Amazon EC2 Auto Scaling beendet das Markieren von Instances als fehlerhaft infolge der Zustandsprüfungen durch EC2 und Elastic Load Balancing. Ihre benutzerdefinierten Zustandsprüfungen funktionieren weiterhin ordnungsgemäß. Wenn Sie `HealthCheck` unterbrochen haben, können Sie den Zustand von Instances in Ihrer Gruppe bei Bedarf manuell festlegen und sie durch `ReplaceUnhealthy` ersetzen lassen.

`InstanceRefresh`ist suspendiert

- Amazon EC2 Auto Scaling beendet das Ersetzen von Instances als Ergebnis einer Instance-Aktualisierung. Wenn eine Instance-Aktualisierung ausgeführt wird, unterbricht dies den Vorgang, ohne ihn abzubrechen.

`ReplaceUnhealthy`ist suspendiert

- Amazon EC2 Auto Scaling stoppt das Ersetzen von Instances, die als fehlerhaft markiert wurden. Instances, die Zustandsprüfungen von EC2 oder Elastic Load Balancing nicht bestehen, werden weiterhin als fehlerhaft markiert. Sobald Sie den Prozess `ReplaceUnhealthy` fortsetzen, ersetzt Amazon EC2 Auto Scaling die Instances, die als fehlerhaft markiert wurden, während dieser Prozess unterbrochen war. Der `ReplaceUnhealthy`-Prozess ruft zuerst `Terminate` und dann `Launch` auf.

`ScheduledActions`ist suspendiert

- Amazon EC2 Auto Scaling führt keine geplanten Aktionen aus, die während des Aussetzungszeitraums geplant sind. Wenn Sie `ScheduledActions` wieder aufnehmen, berücksichtigt Amazon EC2 Auto Scaling nur geplante Aktionen, deren geplante Zeit noch nicht abgelaufen ist.

Weitere Überlegungen

Darüber hinaus, wenn `Launch` oder `Terminate` ausgesetzt sind, funktionieren die folgenden Funktionen möglicherweise nicht richtig:

- Maximale Instanzlebensdauer — Wenn `Launch` oder gesperrt `Terminate` sind, kann die Funktion zur maximalen Instanzlebensdauer keine Instanzen ersetzen.
- Spot-Instance-Unterbrechungen — Wenn `Terminate` die Spot-Instances ausgesetzt sind und Ihre Auto Scaling Scaling-Gruppe über Spot-Instances verfügt, können diese trotzdem beendet werden, falls Spot-Kapazitäten nicht mehr verfügbar sind. Während `Launch` ausgesetzt ist, kann Amazon EC2 Auto Scaling keine Ersatz-Instances aus einem anderen Spot-Instance-Pool oder aus demselben Spot-Instance-Pool starten, wenn dieser wieder verfügbar ist.
- Kapazitätsausgleich — Wenn die Einstellung unterbrochen `Terminate` ist und Sie Capacity Rebalancing verwenden, um Spot-Instance-Unterbrechungen zu beheben, kann der Amazon EC2

Spot-Service Instances trotzdem beenden, falls Spot-Kapazität nicht mehr verfügbar ist. Wenn `Launch` ausgesetzt ist, kann Amazon EC2 Auto Scaling keine Ersatz-Instances aus einem anderen Spot-Instance-Pool oder aus demselben Spot-Instance-Pool starten, wenn dieser wieder verfügbar ist.

- Instances anhängen und trennen — Wenn `Launch` und suspendiert `Terminate` sind, können Sie Instances trennen, die an Ihre Auto Scaling Scaling-Gruppe angehängt sind, aber solange gesperrt `Launch` ist, können Sie keine neuen Instances an die Gruppe anhängen.
- Standby-Instances — Wenn `Launch` und suspendiert `Terminate` sind, können Sie eine Instance in den `Standby` Status versetzen, aber solange sie suspendiert `Launch` ist, können Sie eine Instance im `Standby` Status nicht wieder in Betrieb nehmen.

Überwachen Sie Ihre Amazon EC2 Auto Scaling Scaling-Gruppen

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon EC2 Auto Scaling und Ihrer AWS Cloud Lösungen. AWS bietet die folgenden Überwachungstools, um Amazon EC2 Auto Scaling zu beobachten, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

Health checks (Zustandsprüfungen)

Amazon EC2 Auto Scaling führt regelmäßig Zustandsprüfungen für die Instances in Ihrer Auto-Scaling-Gruppe durch. Wenn eine Instance diese Zustandsprüfungen nicht besteht, wird sie als fehlerhaft markiert und beendet, während Amazon EC2 Auto Scaling eine neue Instance als Ersatz startet. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

AWS Health Dashboard

Das AWS Health Dashboard zeigt Informationen an und bietet auch Benachrichtigungen, die bei Änderungen im Zustand von AWS Ressourcen ausgelöst 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 Amazon EC2 Auto Scaling.

CloudTrail

Mit AWS CloudTrail können Sie die Aufrufe der Amazon EC2 Auto Scaling Scaling-API von oder in Ihrem AWS-Konto Namen verfolgen. CloudTrail speichert die Informationen in Protokolldateien im Amazon S3 S3-Bucket, den Sie angeben. Mit diesen Protokolldateien können Sie die Aktivitäten Ihrer Auto-Scaling-Gruppen überwachen. Protokolle enthalten Informationen dazu, welche Anforderungen erfolgt sind, zu den Quell-IP-Adressen, von denen die Anforderungen kamen, wer die Anforderung gestellt hat, wann die Anforderung erfolgt ist usw. Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling API-Aufrufe protokollieren mit AWS CloudTrail.

> **Protokollerfassung für Ihre Amazon EC2 Instances**
>
> Sie können CloudWatch es verwenden, um Protokolle von den Betriebssystemen für Ihre EC2-Instances zu sammeln. Weitere Informationen finden Sie unter Erfassung von Metriken und Protokollen von Amazon EC2 EC2-Instances und lokalen Servern mit dem CloudWatch Agenten und An CloudWatch Logs gesendete Protokolldaten anzeigen im CloudWatch Amazon-Benutzerhandbuch.
> Informationen zu anderen AWS Diensten, die Ihnen bei der Protokollierung und Erfassung von Daten über Ihre Workloads helfen können, finden Sie im Leitfaden zur Protokollierung und Überwachung für Anwendungsbesitzer in der AWS Prescriptive Guidance.

Amazon CloudWatch

Amazon CloudWatch unterstützt Sie bei der Analyse von Protokollen und bei der Überwachung der Kennzahlen Ihrer AWS Ressourcen und gehosteten Anwendungen in Echtzeit. Sie können Kennzahlen 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. Beispielsweise können Sie benachrichtigt werden, wenn die Netzwerkaktivität plötzlich höher oder niedriger als der erwartete Wert einer Metrik ist. Weitere Informationen über die Verwendung dieses Services zur Überwachung der Metriken Ihrer Auto-Scaling-Gruppen und -Instances finden Sie unter Überwachen Sie CloudWatch Metriken für Ihre Auto Scaling Scaling-Gruppen und -Instances.

CloudWatch verfolgt auch AWS API-Nutzungsmetriken für Amazon EC2 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 CloudWatch Amazon-Benutzerhandbuch.

AWS Compute Optimizer

Compute Optimizer bietet Empfehlungen für Amazon-EC2-Instances, die Ihnen bei der Entscheidung helfen können, zu einem neuen Instance-Typ zu wechseln. Er analysiert, ob der Instance-Typ einer Auto-Scaling-Gruppe optimal ist und generiert Empfehlungen, um die Kosten zu senken und die Leistung Ihrer Workloads zu verbessern. Weitere Informationen finden Sie unter Wird verwendet AWS Compute Optimizer , um Empfehlungen für den Instance-Typ für eine Auto Scaling Scaling-Gruppe abzurufen.

Amazon EventBridge

Amazon EventBridge ist ein serverloser Event-Bus-Service, der es einfach macht, Ihre Anwendungen mit Daten aus einer Vielzahl von Quellen zu verbinden. EventBridge liefert einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, Software-as-a-Service (SaaS) - Anwendungen und AWS Diensten und leitet diese Daten 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 Wird EventBridge zur Behandlung von Auto Scaling Scaling-Ereignissen verwendet.

AWS Security Hub

Verwenden Sie AWS Security Hub, um Ihre Nutzung von Amazon EC2 Auto Scaling im Hinblick auf bewährte Sicherheitsmethoden zu überwachen. Security Hub verwendet aufdeckende Sicherheitskontrollen für die Bewertung von Ressourcenkonfigurationen und Sicherheitsstandards, um Sie bei der Einhaltung verschiedener Compliance-Frameworks zu unterstützen. Weitere Informationen zum Security Hub zur Bewertung von Amazon EC2 Auto Scaling-Ressourcen finden Sie unter Amazon EC2 Auto Scaling-Steuerelemente im AWS Security Hub -Benutzerhandbuch.

Amazon Simple Notification Service

Sie können Auto-Scaling-Gruppen so konfigurieren, dass sie Amazon SNS Benachrichtigungen senden, wenn Amazon-EC2-Auto-Scaling startet. Weitere Informationen finden Sie unter Amazon SNS-Benachrichtigungsoptionen für Amazon EC2 Auto Scaling.

Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe

Amazon EC2 Auto Scaling überwacht kontinuierlich den Integritätsstatus von Instances in einer Auto Scaling Scaling-Gruppe, um die gewünschte Kapazität aufrechtzuerhalten.

Alle Instances in einer Auto Scaling Scaling-Gruppe beginnen mit einem `Healthy` Status. Instances werden als fehlerfrei betrachtet, bis Amazon EC2 Auto Scaling benachrichtigt wird, dass sie fehlerhaft sind. Sie kann Benachrichtigungen von verschiedenen Quellen empfangen, wenn eine Instanz defekt ist und ersetzt werden muss. Diese Quellen umfassen u. a. folgende:

- Amazon EC2
- Elastic Load Balancing
- VPC Lattice

- Benutzerdefinierte Integritätsprüfungen, die Sie definieren

Wenn Amazon EC2 Auto Scaling feststellt, dass eine `InService` Instance fehlerhaft ist, wird sie durch eine neue Instance ersetzt, um die gewünschte Kapazität der Gruppe aufrechtzuerhalten. Die neue Instance wird mit den aktuellen Einstellungen der Auto-Scaling-Gruppe und der zugehörigen Startvorlage oder Startkonfiguration gestartet.

Fehlerhafte Instances können auch auftreten, wenn eine Instance unerwartet beendet wird, z. B. aufgrund einer Unterbrechung der Spot-Instance oder einer manuellen Kündigung durch einen Benutzer. Auch in diesen Fällen startet Amazon EC2 Auto Scaling automatisch eine Ersatz-Instance, um die gewünschte Kapazität aufrechtzuerhalten.

Inhalt

Über Zustandsprüfungen Ihrer Auto-Scaling-Gruppe

Dieses Thema bietet einen Überblick über die verfügbaren Zustandsprüfungstypen und beschreibt die wichtigsten Überlegungen zur Integration von Amazon EC2 Auto Scaling Scaling-Zustandsprüfungen in Ihre Anwendungen.

Inhalt

Health check type (Typ der Zustandsprüfung)

Amazon EC2 Auto Scaling kann den Integritätsstatus einer `InService` Instance mithilfe einer oder mehrerer der folgenden Zustandsprüfungen ermitteln:

Health check type (Typ der Zustandsprüfung)	Was es überprüft
Amazon-EC2-Zustand sprüfungen und geplante Ereignisse	• Überprüft, ob die Instance läuft. • Prüft auf zugrunde liegende Hardware- oder Softwarep robleme, die die Instanz beeinträchtigen könnten. Dies ist der standardmäßige Zustandsprüfungstyp für eine Auto-Scaling-Gruppe.
Elastic Load Balancing-Zustandsprüfungen	• Überprüft, ob der Load Balancer die Instance als fehlerfrei meldet, und bestätigt, ob die Instance für die Bearbeitung von Anfragen verfügbar ist. Um diesen Integritätsprüfungstyp auszuführen, müssen Sie ihn für Ihre Auto Scaling Scaling-Gruppe aktivieren.
VPC-Lattice-Zustandsprüfung en	• Überprüft, ob VPC Lattice die Instance als fehlerfrei meldet, und bestätigt, ob die Instance für die Bearbeitung von Anfragen verfügbar ist. Um diesen Integritätsprüfungstyp auszuführen, müssen Sie ihn für Ihre Auto Scaling Scaling-Gruppe aktivieren.
Benutzerdefinierte Zustandsp rüfungen	• Sucht gemäß Ihren benutzerdefinierten Integritätsprüfung en nach anderen Problemen, die auf Integritätsprobleme der Instance hinweisen könnten.

Zustandsprüfungen von Amazon EC2

Nachdem eine Instance gestartet wurde, wird sie an die Auto-Scaling-Gruppe angefügt und wechselt in den Zustand `InService`. Weitere Informationen über die verschiedenen Lebenszyklus-Statusse der Instances in einer Auto-Scaling-Gruppe finden Sie unter Instance-Lebenszyklus bei Amazon EC2 Auto Scaling.

Amazon EC2 Auto Scaling überprüft durch eine regelmäßige Zustandsprüfung aller Instances innerhalb der Auto-Scaling-Gruppe, ob sie ausgeführt werden und sich in einem guten Zustand befinden.

Statusüberprüfungen

Amazon EC2 Auto Scaling verwendet standardmäßig die Ergebnisse der Amazon EC2-Instance-Statusprüfungen und System-Statusprüfungen, um den Zustand einer Instance zu bestimmen. Wenn sich die Instance in einem anderen Amazon-EC2-Status als `running` befindet oder der Systemstatus `impaired` ist, betrachtet Amazon EC2 Auto Scaling die Instance als fehlerhaft und ersetzt sie. Dies gilt auch, wenn die Instance einen der folgenden Zustände aufweist:

- `stopping`
- `stopped`
- `shutting-down`
- `terminated`

Die Amazon EC2-Statusprüfungen erfordern keine spezielle Konfiguration und sind stets aktiviert. Weitere Informationen finden Sie unter Arten von Statusprüfungen im Amazon EC2 EC2-Benutzerhandbuch.

> **Important**
>
> Amazon EC2 Auto Scaling lässt die Statusprüfungen gelegentlich ausfallen, ohne Maßnahmen zu ergreifen. Wenn eine Statusüberprüfung fehlschlägt, wartet Amazon EC2 Auto Scaling einige Minuten, AWS bis das Problem behoben ist. Es markiert eine Instance nicht sofort als `Unhealthy`, wenn ihr Status für die Zustandsprüfungen `impaired` wird. Wenn Amazon EC2 Auto Scaling jedoch erkennt, dass sich eine Instance nicht mehr im Status `running` befindet, wird diese Situation umgehend als Fehler behandelt. In diesem Fall markiert es die Instance sofort als `Unhealthy` und ersetzt sie.

Geplante Ereignisse

Amazon EC2 kann gelegentlich Ereignisse auf Ihren Instances planen, die nach einem bestimmten Zeitstempel ausgeführt werden. Weitere Informationen finden Sie unter Geplante Ereignisse für Ihre Instances im Amazon EC2 EC2-Benutzerhandbuch.

Wenn eine Ihrer Instances von einem geplanten Ereignis betroffen ist, betrachtet Amazon EC2 Auto Scaling die Instance als fehlerhaft und ersetzt sie. Die Instance wird erst heruntergefahren, wenn das im Zeitstempel angegebene Datum und die Uhrzeit erreicht sind.

Elastic Load Balancing-Zustandsprüfungen

Wenn Sie Elastic Load Balancing Health Checks für Ihre Auto Scaling-Gruppe aktivieren, kann Amazon EC2 Auto Scaling die Ergebnisse dieser Zustandsprüfungen verwenden, um den Integritätsstatus einer Instance zu ermitteln.

Bevor Sie Elastic Load Balancing Health Checks für Ihre Auto Scaling Scaling-Gruppe aktivieren können, müssen Sie einen Elastic Load Balancing Load Balancer konfigurieren und dafür eine Integritätsprüfung konfigurieren, um festzustellen, ob Ihre Instances fehlerfrei sind. Weitere Informationen finden Sie unter Bereiten Sie sich darauf vor, Ihrer Auto Scaling Scaling-Gruppe einen Elastic Load Balancing-Load Balancer hinzuzufügen.

Nachdem Sie den Load Balancer Ihrer Auto Scaling Scaling-Gruppe hinzugefügt haben, passiert Folgendes:

- Amazon EC2 Auto Scaling registriert die Instances in der Auto-Scaling-Gruppe beim Load Balancer.
- Nachdem eine Instance die Registrierung beendet hat, wechselt sie in den Status `InService` und wird für die Verwendung mit dem Load Balancer verfügbar.

Standardmäßig ignoriert Amazon EC2 Auto Scaling die Ergebnisse der Zustandsprüfungen des Elastic Load Balancing. Wenn Sie diese Zustandsprüfungen für Ihre Auto Scaling-Gruppe aktiviert haben und Elastic Load Balancing eine registrierte Instance als meldet`Unhealthy`, markiert Amazon EC2 Auto Scaling die Instance `Unhealthy` bei der nächsten regelmäßigen Integritätsprüfung und ersetzt sie.

Wenn der Connection Draining (Verzögerte Deregistrierung) für Ihren Load Balancer aktiviert ist, wartet Amazon EC2 Auto Scaling, bis die laufenden Anforderungen abgeschlossen werden oder das maximale Zeitlimit abgelaufen ist, bevor die fehlerhaften Instances beendet werden.

> **Note**
>
> Anweisungen, wie Sie den Load Balancer anhängen und Elastic Load Balancing Health Checks für Ihre Auto Scaling Scaling-Gruppe aktivieren, finden Sie unterFügen Sie Ihrer Auto Scaling Scaling-Gruppe einen Elastic Load Balancing Load Balancer hinzu.
> Wenn Sie Elastic Load Balancing Health Checks für eine Gruppe aktivieren, kann Amazon EC2 Auto Scaling Instances ersetzen, die Elastic Load Balancing als fehlerhaft meldet, aber erst, nachdem sich der Load Balancer im Status befindet. `InService` Weitere Informationen finden Sie unter Überprüfen des Anhangsstatus Ihres Load Balancers.

VPC-Lattice-Zustandsprüfungen

Standardmäßig ignoriert Amazon EC2 Auto Scaling die Ergebnisse der VPC-Lattice-Zustandsprüfungen. Sie können diese Integritätsprüfungen optional für Ihre Auto Scaling Scaling-Gruppe aktivieren. Wenn Sie dies getan haben und VPC-Lattice eine registrierte Instance als `Unhealthy` meldet, markiert Amazon EC2 Auto Scaling die Instance bei der nächsten regelmäßigen Zustandsprüfung als `Unhealthy` und ersetzt sie. Der Prozess des Registrierens von Instances und der anschließenden Überprüfung ihres Zustands entspricht der Funktionsweise von Elastic-Load-Balancing-Zustandsprüfungen.

> **Note**
>
> Anweisungen zum Anhängen der VPC Lattice-Zielgruppe und zum Aktivieren der VPC Lattice-Zustandsprüfungen für Ihre Auto Scaling Scaling-Gruppe finden Sie unter. Hinzufügen einer VPC-Lattice-Zielgruppe zu Ihrer Auto-Scaling-Gruppe
> Wenn Sie VPC Lattice-Integritätsprüfungen für eine Gruppe aktivieren, kann Amazon EC2 Auto Scaling Instances ersetzen, die VPC Lattice als fehlerhaft meldet, aber erst, nachdem sich die Zielgruppe im Status befindet. `InService` Weitere Informationen finden Sie unter Überprüfen Sie den Anhangsstatus Ihrer VPC Lattice-Zielgruppe.

Wie Amazon EC2 Auto Scaling Ausfallzeiten minimiert

Standardmäßig werden neue Instances gleichzeitig bereitgestellt, wenn Ihre vorhandenen Instances beendet werden. Dadurch können neue Anfragen möglicherweise nicht akzeptiert werden, bis die neuen Instances voll funktionsfähig sind.

Wenn Amazon EC2 Auto Scaling feststellt, dass Instances nicht mehr laufen (oder sie `Unhealthy` mit dem Befehl set-instance-health markiert wurden), werden sie sofort ersetzt. Wenn jedoch andere Instances als fehlerhaft eingestuft werden, verwendet Amazon EC2 Auto Scaling den folgenden Ansatz, um sich von Ausfällen zu erholen. Dieser Ansatz minimiert Ausfallzeiten, die aufgrund vorübergehender Probleme oder falsch konfigurierter Integritätsprüfungen auftreten können.

- Wenn gerade eine Skalierungsaktivität läuft und Ihre Auto Scaling-Gruppe die gewünschte Kapazität um 10 Prozent oder mehr unterschreitet, wartet Amazon EC2 Auto Scaling auf die laufende Skalierungsaktivität, bevor die fehlerhaften Instances ersetzt werden.
- Bei der Skalierung wartet Amazon EC2 Auto Scaling darauf, dass die Instances eine erste Zustandsprüfung bestehen. Es wartet auch darauf, dass die Standard-Instance-Aufwärmphase beendet ist, um sicherzustellen, dass die neuen Instances bereit sind.
- Nachdem die Instances das Warmlaufen abgeschlossen haben und die Gruppe auf mehr als 90 Prozent der gewünschten Kapazität angewachsen ist, ersetzt Amazon EC2 Auto Scaling die fehlerhaften Instances wie folgt:
  - Amazon EC2 Auto Scaling ersetzt jeweils nur bis zu 10 Prozent der gewünschten Kapazität der Gruppe. Dies geschieht, bis alle fehlerhaften Instances ersetzt wurden.
  - Beim Ersetzen von Instances wird gewartet, bis die neuen Instances eine erste Zustandsprüfung bestanden haben. Es wartet auch darauf, dass das Aufwärmen der Standard-Instance abgeschlossen ist, bevor es fortfährt.

> **Note**
>
> Wenn die Größe einer Auto Scaling-Gruppe so klein ist, dass der resultierende Wert von 10 Prozent kleiner als eins ist, ersetzt Amazon EC2 Auto Scaling stattdessen die fehlerhaften Instances nacheinander. Dies kann zu Ausfallzeiten für die Gruppe führen.
> Wenn alle Instances in einer Auto-Scaling-Gruppe von Elastic Load Balancing-Zustandsprüfungen als fehlerhaft gemeldet werden und sich der Load Balancer im `InService`-Status befindet, markiert Amazon EC2 Auto Scaling möglicherweise weniger Instances gleichzeitig als fehlerhaft. Dies kann dazu führen, dass viel weniger Instances gleichzeitig ersetzt werden als die 10 Prozent, die in anderen Szenarien angewendet wurden. Dadurch haben Sie Zeit, das Problem zu beheben, ohne dass Amazon EC2 Auto Scaling automatisch die gesamte Gruppe beendet.

Gesundheitschecks für Instanzen in einem warmen Pool

Amazon EC2 Auto Scaling führt auch Integritätsprüfungen für Instances in einem warmen Pool durch. Weitere Informationen finden Sie unter Anzeigen des Status der Zustandsprüfung und dem Grund für Zustandsprüfungsfehler.

Überlegungen zur Zustandsprüfung

Im Folgenden finden Sie Überlegungen zur Verwendung von Amazon EC2 Auto Scaling Scaling-Zustandsprüfungen.

- Sie Lebenszyklus-Hooks verwenden, wenn auf der Instance, die beendet wird, oder auf der Instance, die gestartet wird, etwas passieren muss. Mithilfe dieser Hooks können Sie eine benutzerdefinierte Aktion ausführen, wenn Amazon EC2 Auto Scaling Instances startet oder beendet. Weitere Informationen finden Sie unter Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling.
- Amazon EC2 Auto Scaling bietet keine Möglichkeit, die Amazon EC2-Statusprüfungen und geplante Ereignisse aus den Zustandsprüfungen auszuschließen. Wenn Sie nicht möchten, dass Instances ersetzt werden, empfehlen wir Ihnen, den `ReplaceUnhealthy`- und `HealthCheck`-Prozess für einzelne Auto-Scaling-Gruppen auszusetzen. Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling Scaling-Prozesse aussetzen und fortsetzen.
- Um den Zustand einer fehlerhaften Instance manuell auf `Healthy` zu setzen, können Sie versuchen, den Befehl set-instance-health zu verwenden. Wenn Sie eine Fehlermeldung erhalten, liegt das wahrscheinlich daran, dass die Instance bereits beendet ist. Im Allgemeinen ist es nur dann sinnvoll, die Zustandsprüfung einer Instance mit dem Befehl set-instance-health wieder auf `Healthy` zu setzen, wenn entweder der `ReplaceUnhealthy`-Prozess oder der `Terminate`-Prozess selbst ausgesetzt ist.
- Wenn Sie Fehler bei einer Instance beheben müssen, ohne dass es zu Störungen durch Integritätsprüfungen kommt, können Sie die Instance in den `Standby` Status versetzen. Amazon EC2 Auto Scaling führt keine Integritätsprüfungen für Instances durch, die sich im `Standby` Status befinden, bis Sie die Instances wieder in Betrieb nehmen. Weitere Informationen finden Sie unter Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe.
- Wenn die Instance beendet wird, werden alle zugehörigen Elastic IP-Adressen von ihr getrennt und der neuen Instance nicht automatisch zugeordnet. Sie müssen die elastische IP-Adressen manuell mit der neuen Instance verknüpfen oder dies automatisch mit einer auf Lebenszyklus-Hook-basierten Lösung tun. Weitere Informationen finden Sie unter Elastische IP-Adressen im Amazon-EC2-Benutzerhandbuch.

- In ähnlicher Weise werden beim Beenden der Instance ihre zugehörigen EBS-Volumes getrennt (oder je nach `DeleteOnTermination`-Attribut des Volumes gelöscht). Sie müssen diese EBS-Volumes manuell an die neue Instance anhängen oder dies automatisch mit einer Lebenszyklus-Hook-basierten Lösung tun. Weitere Informationen finden Sie unter Anhängen eines Amazon EBS-Volumes an eine Instance im Amazon EBS-Benutzerhandbuch.

Benutzerdefinierte Zustandsprüfungen

Optional können Sie auch benutzerdefinierte Zustandserkennungs-Aufgaben für die Instances in Ihrer Auto-Scaling-Gruppe ausführen und den Zustandsstatus einer Instance als `Unhealthy` festlegen, wenn die Aufgabe fehlschlägt. Dies erweitert Ihre Zustandsprüfungen durch eine Kombination aus benutzerdefinierten Zustandsprüfungen, Amazon-EC2-Statusprüfungen und Elastic-Load-Balancing-Zustandsprüfungen, falls aktiviert.

Sie können die Zustandsinformationen der Instance über die AWS CLI oder ein SDK direkt von Ihrem System an Amazon EC2 Auto Scaling senden. Die folgenden Beispiele zeigen, wie Sie den AWS CLI Integritätsstatus einer Instance konfigurieren und anschließend den Integritätsstatus der Instance überprüfen können.

Verwenden Sie den folgenden set-instance-health-Befehl, um den Zustand der angegebenen Instance in **`Unhealthy`** zu ändern:

```
aws autoscaling set-instance-health --instance-id i-1234567890abcdef0 --health-
status Unhealthy
```

Standardmäßig wird bei diesem Befehl die Wartefrist für die Zustandsprüfung eingehalten. Sie können jedoch dieses Verhalten außer Kraft setzen und die Übergangsfrist nicht einhalten, indem Sie die Option `--no-should-respect-grace-period` einbeziehen.

Verwenden Sie den folgenden describe-auto-scaling-groups-Befehl, um zu prüfen, ob der Instance-Zustand `Unhealthy` ist.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg
```

Im Folgenden sehen Sie eine Beispielantwort, die angibt, dass der Zustand der Instance `Unhealthy` lautet und die Instance beendet wird:

```
{
    "AutoScalingGroups": [
```

```
        {
            ....
            "Instances": [
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-1234567890abcdef0"
                    },
                    "InstanceId": "i-1234567890abcdef0",
                    "InstanceType": "t2.micro",
                    "HealthStatus": "Unhealthy",
                    "LifecycleState": "Terminating"
                },
                ...
            ]
        }
    ]
}
```

Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest

Wenn durch eine Amazon EC2 Auto Scaling Zustandsprüfung festgestellt wird, dass eine `InService`-Instance fehlerhaft ist, wird die Instance durch eine neue Instance ersetzt. Die Frist für die Zustandsprüfung gibt die Mindestdauer (in Sekunden) an, die eine neue Instance in Betrieb bleiben muss, bevor sie beendet wird, wenn sie als fehlerhaft erkannt wird.

In einem bestimmten Anwendungsfall kann es erforderlich sein, auf Amazon EC2 Auto Scaling zu warten, wenn die Zustandsprüfungen für Elastic Load Balancing fehlschlagen und die Instance noch initialisiert wird. Die Zustandsprüfungen von Elastic Load Balancing werden parallel ausgeführt, beginnend mit der Registrierung der Instance beim Load Balancer. Die Übergangsfrist verhindert, dass Amazon EC2 Auto Scaling Ihre neu gestarteten Instances markiert `Unhealthy` und unnötig beendet, wenn sie diese Zustandsprüfungen nicht sofort bestehen, nachdem sie den Status erreicht haben. `InService`

In der Konsole beträgt der Kulanzzeitraum für die Zustandsprüfung standardmäßig 300 Sekunden, wenn Sie eine Auto-Scaling-Gruppe erstellen. Der Standardwert ist 0 Sekunden, wenn Sie eine Auto

Scaling Scaling-Gruppe mit dem AWS CLI oder einem SDK erstellen. Ein Wert von 0 deaktiviert die Nachfrist für Zustandsprüfungen.

Wenn Sie diesen Wert zu hoch festlegen, wird die Effektivität der Amazon EC2 Auto Scaling-Zustandsprüfungen verringert. Wenn Sie einen Lebenszyklus-Hook für den Instance-Start verwenden, können Sie den Wert der Übergangsfrist für die Zustandsprüfung auf 0 festlegen. Mit Lifecycle-Hooks bietet Amazon EC2 Auto Scaling eine Möglichkeit, sicherzustellen, dass Instances immer initialisiert werden, bevor sie in den Status `InService` gelangen. Weitere Informationen finden Sie unter Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling.

Die Nachfrist gilt für die folgenden Fälle:

- Neu gestartete Instances
- Instances, die nach dem Bereitschaftsmodus wieder in Betrieb genommen werden
- Instances, die Sie manuell an die Gruppe anhängen

> **Important**
>
> Wenn Amazon EC2 Auto Scaling erkennt, dass sich eine Instance während der Nachfrist für Zustandsprüfungen nicht mehr im Zustand `running` befindet, markiert Amazon EC2 Auto Scaling sie als `Unhealthy` und ersetzt sie. Wenn Sie beispielsweise eine Instance in einer Auto-Scaling-Gruppe beenden, wird sie als `Unhealthy` markiert und ersetzt.

Legen Sie die Wartefrist für die Zustandsprüfung einer Gruppe fest

Sie können die Wartefrist für die Zustandsprüfung für neue und vorhandene Auto-Scaling-Gruppen festlegen.

Console

Um den Kulanzzeitraum für die Integritätsprüfung für eine neue Gruppe zu ändern

Wenn Sie die Auto Scaling Scaling-Gruppe erstellen, geben Sie den Zeitraum (in Sekunden) auf der Seite Erweiterte Optionen konfigurieren unter Integritätsprüfungen und Kulanzzeitraum Health Integritätsprüfungen ein. So lange muss Amazon EC2 Auto Scaling warten, bevor es den Integritätsstatus einer Instance überprüft, nachdem sie den `InService` Status erreicht hat.

AWS CLI

Um den Kulanzzeitraum für die Integritätsprüfung für eine neue Gruppe zu ändern

Fügen Sie dem Befehl create-auto-scaling-group die Option `--health-check-grace-period` hinzu. Im folgenden Beispiel wird der Karenzzeit für die Zustandsprüfung mit einem Wert von *60* Sekunden für eine neue Auto-Scaling-Gruppe mit dem Namen *my-asg* konfiguriert.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --health-check-grace-period 60 ...
```

Console

Um den Kulanzzeitraum für die Integritätsprüfung für eine bestehende Gruppe zu ändern

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie in der Navigationsleiste oben die AWS-Region aus, in der Sie Ihre Auto-Scaling-Gruppe erstellt haben.
3. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
4. Wählen Sie auf der Registerkarte Details die Option Zustandsprüfungen, Bearbeiten aus.
5. Geben Sie unter Karenzzeit für die Zustandsprüfung die Zeit in Sekunden ein. So lange muss Amazon EC2 Auto Scaling warten, bevor es den Integritätsstatus einer Instance überprüft, nachdem sie den `InService` Status erreicht hat.
6. Wählen Sie Aktualisieren.

AWS CLI

Um den Kulanzzeitraum für die Integritätsprüfung für eine bestehende Gruppe zu ändern

Fügen Sie die Option `--health-check-grace-period` dem Befehl update-auto-scaling-group hinzu. Im folgenden Beispiel wird die Übergangsfrist für die Integritätsprüfung mit einem Wert von *120* Sekunden für eine vorhandene Auto-Scaling-Gruppe mit dem Namen *my-asg* konfiguriert.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
```

```
--health-check-grace-period 120
```

> **Note**
>
> Wir empfehlen dringend, auch die standardmäßige Instance-Vorbereitungs- und Wartefrist für Ihre Auto-Scaling-Gruppe festzulegen. Weitere Informationen finden Sie unter Legen Sie die standardmäßige Instance-Vorbereitung für eine Auto-Scaling-Gruppe fest.

Anzeigen des Grundes für Fehler bei Zustandsprüfung

Mithilfe des folgenden Verfahrens können Sie Informationen zu allen Instances einsehen, die aufgrund einer Zustandsprüfung ersetzt wurden.

Standardmäßig erstellt Amazon EC2 Auto Scaling eine neue Skalierung zum Beenden der fehlerhaften Instance und beendet diese anschließend. Während die Instance beendet wird, startet eine andere Skalierungsaktivität eine neue Instance. Sie können dieses Verhalten ändern, um so schnell wie möglich mit dem Start einer neuen Instance zu beginnen, indem Sie eine Instance-Wartungsrichtlinie verwenden. Weitere Informationen finden Sie unter Wartungsrichtlinien für Instances.

Console

Den Grund für fehlgeschlagene Zustandsprüfungen anzeigen

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe.

   Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet.
3. Auf der Registerkarte Activity (Aktivität) wird unter Activity history (Aktivitätsverlauf) in der Spalte Status angezeigt, ob Ihre Auto-Scaling-Gruppe Instances erfolgreich gestartet oder beendet hat.

   Wenn es Instances fehlerhaft beendet hat, zeigt die Spalte Ursache das Datum und die Uhrzeit der Beendigung und den Grund für den Fehler der Zustandsprüfung an. z. B. `At 2022-05-14T20:11:53Z an instance was taken out of service in response`

`to a user health-check`. Diese Meldung weist darauf hin, dass die Instanz bei einer benutzerdefinierten Integritätsprüfung als fehlerhaft eingestuft wurde.

Hilfe bei fehlgeschlagenen Zustandsprüfungen finden Sie unterFehlerbehebung bei fehlerhaften Instances in Amazon EC2 Auto Scaling.

AWS CLI

Den Grund für fehlgeschlagene Integritätsprüfungen anzeigen

Verwenden Sie den folgenden describe-scaling-activities-Befehl.

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

Im Folgenden finden Sie ein Antwortbeispiel, das den Grund für das Fehlschlagen der Integritätsprüfung `Cause` enthält.

```
{
  "Activities": [
    {
      "ActivityId": "4c65e23d-a35a-4e7d-b6e4-2eaa8753dc12",
      "AutoScalingGroupName": "my-asg",
      "Description": "Terminating EC2 instance: i-04925c838b6438f14",
      "Cause": "At 2021-04-01T21:48:35Z an instance was taken out of service in
 response to a user health-check.",
      "StartTime": "2021-04-01T21:48:35.859Z",
      "EndTime": "2021-04-01T21:49:18Z",
      "StatusCode": "Successful",
      "Progress": 100,
      "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-
west-2a\"...}",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
    },
...
  ]
}
```

Eine Beschreibung der Felder in der Ausgabe finden Sie unterAktivität in der Amazon EC2 Auto Scaling API-Referenz.

Um die Skalierungsaktivitäten zu beschreiben, nachdem die Auto Scaling Scaling-Gruppe gelöscht wurde, fügen Sie die `--include-deleted-groups` Option zum Befehl describe-scaling-activities hinzu.

Fehlerbehebung bei fehlerhaften Instances in Amazon EC2 Auto Scaling

Im Folgenden finden Sie die von Amazon EC2 Auto Scaling zurückgegebenen Fehlermeldungen, die möglichen Ursachen und die Schritte, die Sie zur Behebung der Probleme ergreifen können.

Wie Sie eine Fehlermeldung abrufen, erfahren Sie unter Anzeigen des Grundes für Fehler bei Zustandsprüfung.

Fehlermeldungen

- Eine Instance wurde aufgrund eines Fehlers der EC2-Instance-Zustandsprüfung außer Betrieb genommen
- Eine Instance wurde als Reaktion auf eine EC2-Zustandssprüfung, die anzeigte, dass sie beendet oder angehalten wurde, außer Betrieb genommen.
- Eine Instance wurde aufgrund eines Fehlers der Zustandsprüfung des ELB-Systems außer Betrieb genommen
- Weitere Ressourcen

Eine Instance wurde aufgrund eines Fehlers der EC2-Instance-Zustandsprüfung außer Betrieb genommen

Problem: Auto-Scaling-Instances lassen die Amazon EC2-Zustandsprüfungen fehlschlagen.

Ursache 1: Wenn es Probleme gibt, die dazu führen, dass Amazon EC2 die Instances in Ihrer Auto Scaling-Gruppe als beeinträchtigt betrachtet, ersetzt Amazon EC2 Auto Scaling die Instances im Rahmen seiner Zustandsprüfungen automatisch.

Lösung 1: Wenn eine Überprüfung des Instance-Status fehlschlägt, müssen Sie das Problem in der Regel selbst beheben, indem Sie Änderungen an der Instance-Konfiguration vornehmen, bis Ihre Anwendung keine Probleme mehr aufweist. Um dieses Problem zu beheben, führen Sie die folgenden Schritte aus:

1. Erstellen Sie manuell eine Amazon EC2-Instance, die nicht Teil der Auto-Scaling-Gruppe ist, und untersuchen Sie das Problem. Allgemeine Hilfe bei der Untersuchung beeinträchtigter

Instances finden Sie unter Troubleshooting Instances with failed status checks im Amazon EC2 EC2-Benutzerhandbuch und Troubleshooting Windows Instances im Amazon EC2 EC2-Benutzerhandbuch.

2. Nachdem Sie sich vergewissert haben, dass Ihre Instance erfolgreich gestartet wurde und in Ordnung ist, verteilen Sie eine neue, fehlerfreie Instanzkonfiguration an die Auto-Scaling-Gruppe.
3. Löschen Sie die von Ihnen erstellte Instance, um zu verhindern, dass Ihr AWS -Konto weiter belastet wird.

Eine Instance wurde als Reaktion auf eine EC2-Zustandssprüfung, die anzeigte, dass sie beendet oder angehalten wurde, außer Betrieb genommen.

Problem: Auto-Scaling-Instances, die angehalten, neu gestartet oder beendet wurden, werden ersetzt.

Ursache 1: Ein Benutzer hat die Instance manuell angehalten, neu gestartet oder beendet.

Lösung 1: Wenn Sie die Instances in Ihrer Auto Scaling Scaling-Gruppe beenden oder neu starten müssen, empfehlen wir, die Instances zuerst in den Standby-Modus zu versetzen. Weitere Informationen finden Sie unter Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe.

Ursache 2: Amazon EC2 Auto Scaling versucht, Spot-Instances zu ersetzen, nachdem der Amazon EC2 Spot-Services die Instances unterbricht, da der Spot-Preis über Ihren Höchstpreis steigt oder die Kapazität nicht mehr verfügbar ist.

Lösung 2: Es gibt keine Garantie, dass eine Spot-Instance existiert, um die Anforderung zu einem bestimmten Zeitpunkt zu erfüllen. Sie können die folgenden Schritte versuchen:

- Verwenden Sie einen höheren Spot-Höchstpreis (möglicherweise den On-Demand-Preis). Wenn Sie Ihren Höchstpreis höher setzen, ist die Chance höher, dass der Amazon EC2 Spot-Service Ihre erforderliche Kapazität startet und erhält.
- Erhöhen Sie die Anzahl der verschiedenen Kapazitätspools, über die Sie Instances starten können, indem Sie mehrere Instance-Typen in mehreren Availability Zones ausführen. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.
- Wenn Sie mehrere Instance-Typen verwenden, sollten Sie die Funktion „Kapazitätsausgleich“ aktivieren. Dies ist nützlich, wenn der Amazon EC2 Spot-Service versucht, eine neue Spot-

Instance zu starten, bevor eine laufende Instance beendet wird. Weitere Informationen finden Sie unter Verwenden des Kapazitätsausgleichs, um Amazon-EC2-Spot-Unterbrechungen zu behandeln.

Ursache 3: Mit Capacity Blocks beendet Amazon EC2 alle Instances, die noch laufen, 30 Minuten vor der Endzeit des Capacity Blocks. Diese abrupte Beendigung veranlasst Ihre Auto Scaling Scaling-Gruppe, neue Instances zu starten, um die gewünschte Kapazität aufrechtzuerhalten, auch wenn der Kapazitätsblock endet.

Lösung 3: Versuchen Sie Folgendes, um dieses Problem zu beheben:

- Verringern Sie die gewünschte Kapazität der Auto Scaling Scaling-Gruppe, um zu verhindern, dass sie versucht, neue Instances zu starten. Weitere Informationen finden Sie unter Manuelle Skalierung für Amazon EC2 Auto Scaling.
- Stellen Sie sicher, dass Sie in Ihrer Auto Scaling Scaling-Gruppe 30 Minuten vor dem Ende des Kapazitätsblocks skalieren, damit dieser Fehler nicht häufig auftritt. Stellen Sie sicher, dass alle Lifecycle-Hooks 30 Minuten vor dem Ende des Kapazitätsblocks abgeschlossen sind. Weitere Informationen finden Sie unter Capacity BlocksFür Machine-Learning-Workloads verwenden.

Eine Instance wurde aufgrund eines Fehlers der Zustandsprüfung des ELB-Systems außer Betrieb genommen

Problem: Auto-Scaling-Instances bestehen womöglich die EC2-Zustandsprüfungen. Jedoch können die Elastic Load Balancing-Zustandsprüfungen für die Zielgruppen oder Classic Load Balancers fehlschlagen, bei denen die Auto-Scaling-Gruppe registriert ist.

Ursache 1: Wenn sich Ihre Auto Scaling-Gruppe auf Zustandsprüfungen von Elastic Load Balancing stützt, bestimmt Amazon EC2 Auto Scaling den Integritätsstatus Ihrer Instances, indem es die Ergebnisse sowohl der EC2-Statusprüfungen als auch der Elastic Load Balancing Balancing-Zustandsprüfungen überprüft. Der Load Balancer führt Zustandsprüfungen durch, indem er eine Anforderung an jede Instance sendet und auf die richtige Antwort wartet, oder indem er eine Verbindung mit der Instance herstellt. Eine Instance kann bei einer Elastic Load Balancing-Zustandsprüfung fehlschlagen, weil eine Anwendung, die in der Instance ausgeführt wird, Probleme verursacht, die dazu führen, dass der Load Balancer die Instance außer Betrieb nimmt.

Lösung 1: So werden die Elastic Load Balancing-Zustandsprüfungen erfolgreich durchlaufen

- Stellen Sie sicher, dass die Einstellungen für die Zustandsprüfung Ihrer Zielgruppen korrekt konfiguriert sind. Sie definieren Zustandsprüfungseinstellungen für Ihren Load Balancer pro Zielgruppe. Weitere Informationen finden Sie unter Konfigurieren Sie Integritätsprüfungen für Ziele.
- Überprüfen Sie die Erfolgscodes, die der Load Balancer erwartet, und stellen Sie sicher, dass Ihre Anwendung so konfiguriert ist, dass sie diese Codes bei Erfolg zurückgibt.
- Stellen Sie sicher, dass die Sicherheitsgruppen für Ihren Load Balancer und die Auto-Scaling-Gruppe korrekt konfiguriert sind.
- Stellen Sie sicher, dass der Load Balancer in denselben Availability Zones wie Ihre Auto-Scaling-Gruppe konfiguriert ist.

Lösung 2: Aktualisieren Sie die Auto-Scaling-Gruppe, um Elastic Load Balancing-Zustandsprüfungen zu deaktivieren. Anweisungen zum Deaktivieren dieser Zustandsprüfungen finden Sie unter. Fügen Sie Ihrer Auto Scaling Scaling-Gruppe einen Elastic Load Balancing Load Balancer hinzu

Ursache 2: Es besteht eine Diskrepanz zwischen dem Kulanzzeitraum der Zustandsprüfung und der Startzeit der Instance.

Lösung 3: Bearbeiten Sie den Kulanzzeitraum für die Integritätsprüfung für Ihre Auto Scaling Scaling-Gruppe. Legen Sie den Kulanzzeitraum auf einen ausreichend langen Zeitraum fest, um die Anzahl der aufeinanderfolgenden erfolgreichen Zustandsprüfungen zu unterstützen, die erforderlich sind, bevor Elastic Load Balancing eine neu gestartete Instance als fehlerfrei einstuft. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.

Weitere Ressourcen

Wenn Sie ein anderes Problem haben, finden Sie in den folgenden AWS re:Post Artikeln zusätzliche Hilfe zur Fehlerbehebung:

- Warum hat Amazon EC2 Auto Scaling eine Instance beendet?
- Warum hat Amazon EC2 Auto Scaling eine fehlerhafte Instance nicht beendet?

AWS Health Dashboard Benachrichtigungen für Amazon EC2 Auto Scaling

Ihr AWS Health Dashboard bietet Unterstützung für Benachrichtigungen, die von Amazon EC2 Auto Scaling stammen. Diese Benachrichtigungen bieten umfassende Informationen sowie

Anweisungen zur Behebung von Ressourcenleistungs- oder Verfügbarkeitsproblemen, die sich auf Ihre Anwendungen auswirken können. Derzeit sind nur Ereignisse verfügbar, die für fehlende Sicherheitsgruppen und Startvorlagen spezifisch sind.

Das AWS Health Dashboard ist Teil 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 Ihrem AWS Health Dashboard.

Wenn Sie eine Nachricht ähnlich den folgenden erhalten, sollte sie als Alarm behandelt werden, um Maßnahmen zu ergreifen.

Beispiel: Auto Scaling-Gruppe wird aufgrund einer fehlenden Sicherheitsgruppe nicht skaliert

```
 Hello,

 At 2020-01-11 04:00 UTC, we detected an issue with your Auto Scaling group [ARN] in
AWS-Konto 123456789012.

 A security group associated with this Auto Scaling group cannot be found. Each time
a
 scale out operation is performed, it will be prevented until you make a change that
 fixes the issue.

 We recommend that you review and update your Auto Scaling group configuration to
change
 the launch template or launch configuration that depends on the unavailable security
 group.

 Sincerely,
 Amazon Web Services
```

Beispiel: Auto Scaling-Gruppe wird aufgrund einer fehlenden Startvorlage nicht skaliert

```
 Hello,

 At 2021-05-11 04:00 UTC, we detected an issue with your Auto Scaling group [ARN] in
AWS-Konto 123456789012.

 The launch template associated with this Auto Scaling group cannot be found. Each
time
 a scale out operation is performed, it will be prevented until you make a change
that
```

```
fixes the issue.

We recommend that you review and update your Auto Scaling group configuration and
specify an existing launch template to use.

Sincerely,
Amazon Web Services
```

Überwachen Sie CloudWatch Metriken für Ihre Auto Scaling Scaling-Gruppen und -Instances

Metriken sind das grundlegende Konzept bei Amazon CloudWatch. Eine Metrik steht für einen nach der Zeit geordneten Satz von Datenpunkten, die veröffentlicht werden. CloudWatch Sie können sich eine Metrik als eine zu überwachende Variable und die Datenpunkte als die Werte dieser Variablen im Laufe der Zeit vorstellen. Mit diesen Metriken können Sie überprüfen, ob Ihr System die erwartete Leistung zeigt.

Amazon EC2 Auto Scaling-Metriken, die Informationen zu Auto-Scaling-Gruppen sammeln, befinden sich im Namespace `AWS/AutoScaling`. Amazon EC2-Instance-Metriken, die CPU-Auslastungsdaten und andere Nutzungsdaten von Auto Scaling-Instances erfassen, befinden sich im Namespace `AWS/EC2`.

In der Konsole von Amazon EC2 Auto Scaling wird eine Reihe von Diagrammen für die Gruppenmetriken und die aggregierten Instance-Metriken für die Gruppe angezeigt. Je nach Ihren Anforderungen ziehen Sie es möglicherweise vor, auf Daten für Ihre Auto Scaling Scaling-Gruppen und -Instances von Amazon CloudWatch statt über die Amazon EC2 Auto Scaling Scaling-Konsole zuzugreifen.

Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Inhalt

Überwachungsgrafiken in der Amazon EC2 Auto Scaling-Konsole anzeigen

Im Bereich Amazon EC2 Auto Scaling der Amazon EC2 EC2-Konsole können Sie den minute-by-minute Fortschritt einzelner Auto Scaling Scaling-Gruppen anhand CloudWatch von Metriken überwachen.

Sie können die folgenden Arten von Metriken überwachen:

- Auto Scaling-Metriken – Die Metriken für die automatische Skalierung werden nur aktiviert, wenn Sie sie einschalten. Weitere Informationen finden Sie unter Aktivieren der Auto-Scaling-Metriken (Konsole). Wenn die Auto-Scaling-Metriken aktiviert sind, zeigen die Überwachungsgrafiken Daten an, die mit einer Granularität von einer Minute für Auto-Scaling-Metriken veröffentlicht werden.
- EC2-Metriken – Die Amazon-EC2-Instance-Metriken sind immer aktiviert. Wenn die detaillierte Überwachung aktiviert ist, zeigen die Überwachungsdiagramme Daten, die mit einer Granularität von einer Minute für Instance-Metriken veröffentlicht werden. Weitere Informationen finden Sie unter Überwachung für Auto-Scaling-Instances konfigurieren.

So zeigen Sie Überwachungsgrafiken über die Amazon EC2 Auto Scaling-Konsole an

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben der Auto-Scaling-Gruppe, für die Sie Metriken anzeigen möchten.

   Im unteren Teil der Seite Auto-Scaling-Gruppen wird ein geteilter Bereich geöffnet.
3. Wählen Sie die Registerkarte Überwachung.

   Amazon EC2 Auto Scaling zeigt Überwachungsgrafiken für Auto-Scaling-Metriken an.
4. Wenn Überwachungsdiagramme der aggregierten Instance-Metriken für die Gruppe angezeigt werden sollen, wählen Sie EC2 aus.

Graph-Aktionen

- Bewegen Sie den Mauszeiger über einen Datenpunkt, um ein Daten-Popup für eine bestimmte Zeit in UTC anzuzeigen.

- Um ein Diagramm zu vergrößern, wählen Sie Vergrößern aus dem Menüwerkzeug (die drei vertikalen Punkte) oben rechts im Diagramm. Alternativ können Sie auch auf das Symbol zum Maximieren am oberen Rand des Diagramms klicken.
- Passen Sie den Zeitraum für die im Diagramm angezeigten Daten an, indem Sie einen der vordefinierten Werte für den Zeitraum auswählen. Wenn das Diagramm vergrößert ist, können Sie Benutzerdefiniert wählen, um Ihren eigenen Zeitraum zu definieren.
- Wählen Sie Aktualisieren aus dem Menü Werkzeug, um die Daten in einem Diagramm zu aktualisieren.
- Ziehen Sie den Mauszeiger über die Diagrammdaten, um einen bestimmten Bereich auszuwählen. Sie können dann im Menü Werkzeug die Option Zeitbereich anwenden wählen.
- Wählen Sie im Menü-Tool die Option Protokolle anzeigen, um die zugehörigen Protokollstreams (falls vorhanden) in der CloudWatch Konsole anzuzeigen.
- Um ein Diagramm anzuzeigen CloudWatch, wählen Sie im Menü-Tool die Option In Metriken anzeigen aus. Dadurch gelangen Sie auf die CloudWatch Seite für das Diagramm. Dort können Sie weitere Informationen einsehen oder auf historische Daten zugreifen, um besser zu verstehen, wie sich Ihre Auto Scaling Gruppe über einen längeren Zeitraum verändert hat.

Graphische Metriken für Ihre Auto-Scaling-Gruppen

Nachdem Sie eine Auto-Scaling-Gruppe erstellt haben, können Sie die Amazon EC2 Auto Scaling-Konsole öffnen und die Überwachungsgraphen für die Gruppe auf der Registerkarte Überwachung anzeigen.

Im Abschnitt Auto Scaling enthalten die Diagramm-Metriken die folgenden Metriken. Diese Metriken liefern Messwerte, die Indikatoren für ein potenzielles Problem sein können, wie z.B. die Anzahl der abgebrochenen Instances oder die Anzahl der ausstehenden Instances. Sie finden Definitionen für diese Metriken unter CloudWatch Amazon-Metriken für Amazon EC2 Auto Scaling.

Anzeigename	CloudWatch Name der Metrik
Minimale Gruppengröße	`GroupMinSize`
Maximale Gruppengröße	`GroupMaxSize`
Gewünschte Kapazität	`GroupDesiredCapacity`
In-Service-Instances	`GroupInServiceInstances`

Anzeigename	CloudWatch Name der Metrik
Ausstehende Instances	`GroupPendingInstances`
Standby-Instances	`GroupStandbyInstances`
Beendende Instances	`GroupTerminatingInstances`
Gesamtzahl der Instances	`GroupTotalInstances`

Im Abschnitt EC2 finden Sie die folgenden grafischen Metriken, die auf den wichtigsten Leistungskennzahlen für Ihre Amazon EC2-Instanxea basieren. Diese EC2-Metriken sind ein Aggregat von Metriken für alle Instances in der Gruppe. Definitionen für diese Metriken finden Sie unter Liste der verfügbaren CloudWatch Metriken für Ihre Instances im Amazon EC2 EC2-Benutzerhandbuch.

Anzeigename	CloudWatch Name der Metrik
CPU-Auslastung	`CPUUtilization`
Datenträgerlesevorgänge	`DiskReadBytes`
Festplattenlesevorgänge	`DiskReadOps`
Datenträgerschreibvorgänge	`DiskWriteBytes`
Festplattenschreibvorgänge	`DiskWriteOps`
Netzwerkeingang	`NetworkIn`
Netzwerkausgang	`NetworkOut`
Statusprüfung fehlgeschlagen (Beliebig)	`StatusCheckFailed`
Statusprüfung fehlgeschlagen (Instance)	`StatusCheckFailed_Instance`

Anzeigename	CloudWatch Name der Metrik
Statusprüfung fehlgeschlagen (System)	`StatusCheckFailed_System`

Darüber hinaus sind einige Metriken für bestimmte Anwendungsfälle in den grafisch dargestellten Metriken für Auto Scaling verfügbar.

Die folgenden Metriken sind nützlich, wenn Ihre Gruppe Gewichtungen verwendet, die definieren, wie viele Einheiten jede Instance zur gewünschten Kapazität der Gruppe beiträgt. Sie finden Definitionen für diese Metriken unter CloudWatch Amazon-Metriken für Amazon EC2 Auto Scaling.

Anzeigename	CloudWatch Name der Metrik
In-Service-Kapazitätseinheiten	`GroupInServiceCapacity`
Ausstehende Kapazität seinheiten	`GroupPendingCapacity`
Standby-Kapazitätseinheiten	`GroupStandbyCapacity`
Beendende Kapazitätseinheite n	`GroupTerminatingCapacity`
Gesamte Kapazitätseinheiten	`GroupTotalCapacity`

Die folgenden Metriken sind nützlich, wenn Ihre Gruppe das Feature Warmer Pool verwendet. Sie finden Definitionen für diese Metriken unter CloudWatch Amazon-Metriken für Amazon EC2 Auto Scaling.

Anzeigename	CloudWatch Name der Metrik
Mindestgröße des warmen Pools	`WarmPoolMinSize`
Gewünschte Kapazität des warmen Pools	`WarmPoolDesiredCapacity`

Anzeigename	CloudWatch Name der Metrik
Ausstehende Kapazität seinheiten des warmen Pools	`WarmPoolPendingCapacity`
Beenden der Kapazität seinheiten des warmen Pools	`WarmPoolTerminatingCapacity`
Aufgewärmte Kapazität seinheiten des warmen Pools	`WarmPoolWarmedCapacity`
Insgesamt gestartete Kapazitätseinheiten des warmen Pools	`WarmPoolTotalCapacity`
Gewünschte Kapazität für Gruppe und warmen Pool	`GroupAndWarmPoolDesiredCapacity`
Gruppe und insgesamt gestartete Kapazitätseinheiten des warmen Pools	`GroupAndWarmPoolTotalCapacity`

Zugehörige Ressourcen

- Informationen zur Überwachung von Metriken pro Instanz finden Sie unter Graph-Metriken für Ihre Instances im Amazon EC2 EC2-Benutzerhandbuch.
- CloudWatch Dashboards sind anpassbare Homepages in der Konsole. CloudWatch Sie können diese Seiten verwenden, um die Ressourcen in einer Ansicht zu überwachen, auch Ressourcen, die über mehrere Regionen verteilt sind. Sie können CloudWatch Dashboards verwenden, um benutzerdefinierte Ansichten der Messwerte und Alarme für Ihre AWS Ressourcen zu erstellen. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

CloudWatch Amazon-Metriken für Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling veröffentlicht folgende Metriken im Namespace `AWS/AutoScaling`. Die tatsächlich verfügbaren Metriken für Auto-Scaling-Gruppen hängen davon ab, ob Sie Gruppenmetriken aktiviert haben und welche Gruppenmetriken Sie aktiviert haben. Gruppenmetriken

stehen ohne zusätzliche Kosten mit einer Granularität von einer Minute zur Verfügung. Sie müssen sie jedoch aktivieren.

Wenn Sie Auto Scaling-Gruppenmetriken aktivieren, sendet Amazon EC2 Auto Scaling Stichprobendaten nach bestem Wissen und Gewissen an CloudWatch jede Minute. In seltenen Fällen, wenn es zu einer CloudWatch Serviceunterbrechung kommt, werden Daten nicht aufgefüllt, um Lücken im Verlauf der Gruppenmetriken zu schließen.

Inhalt

Metriken zu Auto-Scaling-Gruppen

Mit diesen Metriken erhalten Sie nahezu kontinuierliche Einblicke in den Verlauf Ihrer Auto-Scaling-Gruppe. Hierzu zählen beispielsweise Größenänderungen der Gruppe im Laufe der Zeit.

Metrik	Beschreibung
`GroupMinSize`	Die Mindestgröße der Auto-Scaling-Gruppe. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupMaxSize`	Die maximale Größe der Auto-Scaling-Gruppe. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupDesiredCapacity`	Die Anzahl von Instances, welche die Auto-Scaling-Gruppe beizubehalten versucht. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.

Metrik	Beschreibung
`GroupInServiceInstances`	Die Anzahl von Instances, die im Rahmen der Auto-Scaling-Gruppe ausgeführt werden. Diese Metrik umfasst keine Instances, die schwebend oder beendet sind. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupPendingInstances`	Die Anzahl von schwebenden Instances. Eine schwebende Instance ist noch nicht in Betrieb. Diese Metrik umfasst keine Instances, die in Betrieb oder beendet sind. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupStandbyInstances`	Die Anzahl von Instances mit dem Status `Standby`. Instances in diesem Status werden zwar ausgeführt, sind aber nicht aktiv in Betrieb. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupTerminatingInstances`	Die Anzahl von Instances, die beendet werden. Diese Metrik umfasst keine Instances, die in Betrieb oder schwebend sind. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupTotalInstances`	Die Gesamtanzahl von Instances in der Auto-Scaling-Gruppe. Diese Metrik gibt die Anzahl von Instances an, die in Betrieb, schwebend oder beendet sind. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.

Wenn Sie eine gemischte Instance-Gruppe so konfigurieren, dass die gewünschte Kapazität in verschiedenen Einheiten gemessen wird, z. B. indem Sie Gewichtungen auf der Grundlage der vCPU-Anzahl jedes Instance-Typs zuweisen, zählen die folgenden Metriken die Anzahl der von

Ihrer Auto-Scaling-Gruppe verwendeten Einheiten. Wenn Sie eine gemischte Instance-Gruppe nicht so konfiguriert haben, dass die gewünschte Kapazität in verschiedenen Einheiten gemessen wird, werden die folgenden Metriken zwar ausgefüllt, entsprechen aber den in der vorherigen Tabelle definierten Metriken. Weitere Informationen finden Sie unter Übersicht über die Einrichtung.

Metrik	Beschreibung
`GroupInServiceCapacity`	Die Anzahl der Kapazitätseinheiten, die als Teil der Auto-Scaling-Gruppe ausgeführt werden. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupPendingCapacity`	Die Anzahl der schwebenden Kapazitätseinheiten. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupStandbyCapacity`	Die Anzahl der Kapazitätseinheiten, die sich in einem `Standby`-Status befinden. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupTerminatingCapacity`	Die Anzahl der Kapazitätseinheiten, die im Begriff sind, beendet zu werden. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupTotalCapacity`	Die Gesamtanzahl der Kapazitätseinheiten in der Auto-Scaling-Gruppe. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.

Für Auto-Scaling-Gruppen mit warmem Pool meldet Amazon EC2 Auto Scaling zudem folgende Metriken. Weitere Informationen finden Sie unter Warm-Pools für Amazon EC2 Auto Scaling.

Metrik	Beschreibung
`WarmPoolMinSize`	Die Mindestgröße des warmen Pools. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`WarmPoolDesiredCapacity`	Die Menge an Kapazität, die Amazon EC2 Auto Scaling im warmen Pool zu erhalten versucht. Dies entspricht der maximalen Größe der Auto-Scaling-Gruppe abzüglich ihrer gewünschten Kapazität oder, falls eingestellt, der maximalen vorbereiteten Kapazität der Auto-Scaling-Gruppe abzüglich ihrer gewünschten Kapazität. Wenn jedoch die Mindestgröße des warmen Pools gleich oder größer ist als die Differenz zwischen der maximalen Größe (oder, falls eingestellt, der maximal vorbereiteten Kapazität) und der gewünschten Kapazität der Auto-Scaling-Gruppe, dann entspricht die gewünschte Kapazität des warmen Pools der `WarmPoolMinSize` . Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`WarmPoolPendingCapacity`	Die Menge an Kapazität im warmen Pool, die noch ausstehend ist. Diese Metrik umfasst keine laufenden, gestoppten oder beendeten Instances. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`WarmPoolTerminatingCapacity`	Die Menge der Kapazität im warmen Pool, die gerade abgebaut wird. Diese Metrik umfasst keine laufenden, gestoppten oder ausstehenden Instances. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.

Metrik	Beschreibung
`WarmPoolWarmedCapacity`	Die verfügbare Kapazität, um die Auto-Scaling-Gruppe während der Aufskalierung zu betreten. Diese Metrik umfasst keine Instances, die schwebend oder beendet sind. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`WarmPoolTotalCapacity`	Die Gesamtkapazität des warmen Pools, einschließlich der laufenden, gestoppten, anstehenden oder beendeten Instances. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupAndWarmPoolDesiredCapacity`	Die gewünschte Kapazität der Auto-Scaling-Gruppe und des warmen Pools zusammen. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.
`GroupAndWarmPoolTotalCapacity`	Die Gesamtkapazität der Auto-Scaling-Gruppe und des warmen Pools zusammen. Dazu gehören laufende, gestoppte, ausstehende, beendete oder in Betrieb befindliche Instances. Reporting criteria (Berichtskriterien): Wird gemeldet, wenn die Metrikensammlung aktiviert ist.

Dimensionen für Metriken zu Auto-Scaling-Gruppen

Sie können die folgenden Dimensionen verwenden, um die in den vorherigen Tabellen aufgeführten Metriken zu verfeinern.

Dimension	Beschreibung
`AutoScalingGroupName`	Filtert nach dem Namen einer Auto-Scaling-Gruppe.

Metriken und Dimensionen für die prädiktive Skalierung

Der Namespace `AWS/AutoScaling` enthält folgende Metriken für die prädiktive Skalierung.

Metriken sind mit einer Auflösung von einer Stunde verfügbar.

Zur Bewertung der Prognosegenauigkeit können Sie die prognostizierten Werte mit tatsächlichen Werten vergleichen. Weitere Informationen zur Bewertung der Prognosegenauigkeit mit diesen Metriken finden Sie unter Überwachen Sie Metriken zur vorausschauenden Skalierung mit CloudWatch.

Metrik	Beschreibung	Dimensionen
`PredictiveScalingLoadForecast`	Die Last, die voraussichtlich von Ihrer Anwendung generiert wird. Die Statistiken `Average`, `Minimum` und `Maximum` sind hilfreich, die Statistik `Sum` allerdings nicht. Berichtskriterien: Werden nach Erstellung der ursprüngliche Prognose gemeldet.	`AutoScalingGroupName`, `PolicyName`, `PairIndex`
`PredictiveScalingCapacityForecast`	Die voraussichtliche Kapazität, die zur Deckung des Anwendungsbedarfs erforderlich ist. Dieser Wert basiert auf der Lastprognose und der gewünschten Zielauslastung für Ihre Auto Scaling-Instances. Die Statistiken `Average`, `Minimum` und `Maximum` sind hilfreich, die Statistik `Sum` allerdings nicht. Berichtskriterien: Werden nach Erstellung der ursprüngliche Prognose gemeldet.	`AutoScalingGroupName`, `PolicyName`
`PredictiveScalingM`	Die Korrelation zwischen der Skalierungsmetrik und dem Durchschnitt der Lastmetrik pro Instance. Prädiktive Skalierung geht immer von einer hohen Korrelation aus. Wenn Sie	`AutoScalingGroupName`,

Metrik	Beschreibung	Dimensionen
`etricPair Correlation`	also einen niedrigen Wert für diese Metrik beobachten, ist es besser, kein Metrikpaar zu verwenden. Die Statistiken `Average`, `Minimum` und `Maximum` sind hilfreich, die Statistik `Sum` allerdings nicht. Berichtskriterien: Werden nach Erstellung der ursprüngliche Prognose gemeldet.	`PolicyName` , `PairIndex`

> **Note**
>
> Die Dimension `PairIndex` gibt Informationen im Zusammenhang mit dem Index des Metrikpaars für Last und Skalierung gemäß Zuweisung durch Amazon EC2 Auto Scaling zurück. Der einzige gültige Wert ist derzeit `0`.

Aktivieren der Auto-Scaling-Metriken (Konsole)

So aktivieren Sie Metriken zu Gruppen

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Überwachung Auto-Scaling-Metriken aus und markieren Sie das Feld Aktivieren oben auf der Seite unter Auto Scaling.

So deaktivieren Sie Metriken zu Gruppen

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie Ihre Auto-Scaling-Gruppe aus.

3. Entfernen Sie auf der Registerkarte Überwachung unter Erfassung von Metriken zu Auto-Scaling-Gruppen die Option Aktivieren.

Aktivieren der Metriken zu Auto-Scaling-Gruppen (AWS CLI)

Um Metriken für Auto-Scaling-Gruppen zu aktivieren

Mit dem Befehl enable-metrics-collection können einzelne oder mehrere Gruppenmetriken aktiviert werden. Mit dem folgenden Befehl wird beispielsweise eine einzelne Metrik für die angegebene Auto-Scaling-Gruppe aktiviert.

```
aws autoscaling enable-metrics-collection --auto-scaling-group-name my-asg \
  --metrics GroupDesiredCapacity --granularity "1Minute"
```

Wenn Sie die Option `--metrics` weglassen, werden alle Metriken aktiviert.

```
aws autoscaling enable-metrics-collection --auto-scaling-group-name my-asg \
  --granularity "1Minute"
```

Um Metriken für Auto-Scaling-Gruppen zu deaktivieren

Verwenden Sie den Befehl disable-metrics-collection, um alle Gruppenmetriken zu deaktivieren.

```
aws autoscaling disable-metrics-collection --auto-scaling-group-name my-asg
```

Überwachung für Auto-Scaling-Instances konfigurieren

Amazon EC2 sammelt und verarbeitet Rohdaten von Instances, um lesbare Quasi-Echtzeit-Metriken zu erhalten, die Aufschluss über die CPU-Auslastung und andere Nutzungsdaten für Ihre Auto-Scaling-Gruppe geben. Sie können das Intervall für die Überwachung dieser Metriken konfigurieren, indem Sie eine Granularität von einer Minute oder fünf Minuten auswählen.

Die Instance-Überwachung wird beim Start einer Instance aktiviert und es wird entweder die Basisüberwachung (mit einer Granularität von fünf Minuten) oder die detaillierte Überwachung (mit einer Granularität von einer Minute) verwendet. Für die detaillierte Überwachung werden zusätzliche Gebühren fällig. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise und Überwachung Ihrer Instances mithilfe von Amazon CloudWatch im Amazon EC2-Benutzerhandbuch.

Es empfiehlt sich, vor der Erstellung einer Auto-Scaling-Gruppe eine Startvorlage oder eine Startkonfiguration zu erstellen, die den für Ihre Anwendung geeigneten Überwachungstyp zulässt. Wenn Sie Ihrer Gruppe eine Skalierungsrichtlinie hinzufügen, empfehlen wir dringend, die detaillierte Überwachung zu verwenden, um Metrikdaten für EC2-Instances im Minutentakt zu erhalten, da so schneller auf Laständerungen reagiert werden kann.

Inhalt

Aktivieren der detaillierten Überwachung (Konsole)

Standardmäßig ist die grundlegende Überwachung aktiviert, wenn Sie die verwenden AWS Management Console , um eine Startvorlage oder eine Startkonfiguration zu erstellen.

Aktivieren der detaillierten Überwachung in einer Startvorlage

Wenn Sie die Startvorlage mithilfe von erstellen AWS Management Console, wählen Sie im Abschnitt Erweiterte Details für detaillierte CloudWatch Überwachung die Option Aktivieren aus. Andernfalls ist die grundlegende Überwachung aktiviert. Weitere Informationen finden Sie unter Erstellen einer Startvorlage mithilfe erweiterter Einstellungen.

Aktivieren der detaillierten Überwachung in einer Startvorlage

Wenn Sie die Startkonfiguration mithilfe von erstellen AWS Management Console, wählen Sie im Abschnitt Zusätzliche Konfiguration die Option Detaillierte Überwachung der EC2-Instance aktivieren aus. CloudWatch Andernfalls ist die grundlegende Überwachung aktiviert. Weitere Informationen finden Sie unter Erstellen einer Startkonfiguration.

Aktivieren der detaillierten Überwachung (AWS CLI)

Standardmäßig ist die grundlegende Überwachung aktiviert, wenn Sie die AWS CLI verwenden, um eine Startvorlage zu erstellen. Die detaillierte Überwachung ist standardmäßig aktiviert, wenn Sie eine Startkonfiguration mithilfe der AWS CLI erstellen.

Aktivieren der detaillierten Überwachung in einer Startvorlage

Verwenden Sie für Startvorlagen den Befehl create-launch-template und übergeben Sie eine JSON-Datei, welche die Informationen für die Erstellung der Startvorlage enthält. Legen Sie die Überwachungsparameter zum Aktivieren der detaillierten Überwachung auf `"Monitoring": {"Enabled":true}` und zum Aktivieren der grundlegenden Überwachung auf `"Monitoring": {"Enabled":false}` fest.

Aktivieren der detaillierten Überwachung in einer Startvorlage

Verwenden Sie für Startkonfigurationen den Befehl create-launch-configuration mit der Option `--instance-monitoring`. Legen Sie für diese Option zum Aktivieren der detaillierten Überwachung `true` oder zum Aktivieren der grundlegenden Überwachung `false` fest.

```
--instance-monitoring Enabled=true
```

Zwischen grundlegender und detaillierter Überwachung wechseln

Um die Art der Überwachung zu ändern, die auf neuen EC2-Instances aktiviert ist, aktualisieren Sie die Startvorlage oder aktualisieren Sie die Auto-Scaling-Gruppe so, dass sie eine neue Startvorlage oder Startkonfiguration verwendet. Für bestehende Instances wird weiterhin die zuvor aktivierte Überwachungsart verwendet. Um alle Instances zu aktualisieren, beenden Sie sie, damit sie durch Ihre Auto-Scaling Gruppe ersetzt werden, oder aktualisieren Sie Instances einzeln mithilfe des Befehls monitor-instances und des Befehls unmonitor-instances.

> **Note**
>
> Mit den Funktionen Instance Refresh und maximale Instance-Lebensdauer können Sie auch alle Instances in der Auto-Scaling-Gruppe ersetzen, um neue Instances zu starten, welche die neuen Einstellungen verwenden. Weitere Informationen finden Sie unter Recyceln der Instances in Ihrer Auto-Scaling-Gruppe.

Wenn Sie zwischen grundlegender und detaillierter Überwachung wechseln:

Wenn Sie CloudWatch Alarme mit den schrittweisen Skalierungsrichtlinien oder einfachen Skalierungsrichtlinien für Ihre Auto Scaling Scaling-Gruppe verknüpft haben, verwenden Sie den Befehl put-metric-alarm, um jeden Alarm zu aktualisieren. Sorgen Sie dafür, dass jeder Zeitraum mit der Überwachungsart übereinstimmt (300 Sekunden bei der grundlegenden und 60 Sekunden bei der detaillierten Überwachung). Wenn Sie von der detaillierten zur grundlegenden Überwachung wechseln, den Zeitraum der Alarme jedoch nicht zu fünf Minuten ändern, werden weiterhin minütlich

Statistiken abgerufen. In diesem Fall kann das Ergebnis in vier von fünf Zeitabschnitten lauten, dass keine Daten verfügbar sind.

Erfassen Sie mit dem Agenten zusätzliche Metriken CloudWatch

Um Metriken auf Betriebssystemebene wie verfügbaren und belegten Arbeitsspeicher zu erfassen, müssen Sie den CloudWatch Agenten installieren. Es können zusätzliche Gebühren anfallen. Sie können den CloudWatch Agenten verwenden, um sowohl Systemmetriken als auch Protokolldateien von Amazon EC2 EC2-Instances zu sammeln. Weitere Informationen finden Sie unter Vom CloudWatch Agenten erhobene Metriken im CloudWatch Amazon-Benutzerhandbuch.

Amazon EC2 Auto Scaling API-Aufrufe protokollieren mit AWS CloudTrail

Amazon EC2 Auto Scaling ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem Service mithilfe von Amazon EC2 Auto Scaling ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für Amazon EC2 Auto Scaling als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amazon EC2 Auto Scaling-Konsole und Code-Aufrufe an die Amazon EC2 Auto Scaling-API.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Amazon EC2 Auto Scaling. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon EC2 Auto Scaling gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

Informationen zu Amazon EC2 Auto Scaling in CloudTrail

CloudTrail ist in Ihrem Amazon Web Services Services-Konto aktiviert, wenn Sie das Konto erstellen. Wenn in Amazon EC2 Auto Scaling eine CloudTrail Aktivität auftritt, wird diese Aktivität zusammen mit anderen Amazon Web Services Services-Ereignissen im Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können die neusten Ereignisse in Ihrem Amazon Web Services-Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Erstellen Sie einen Trail zur laufenden Aufzeichnung der Ereignisse in Ihrem Amazon Web Services-Konto, einschließlich der Ereignisse für Amazon-EC2-Auto-Scaling. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie ein Trail in der Konsole anlegen, gilt dieser für alle Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der Amazon Web Services-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie andere Amazon Web Services so konfigurieren, dass sie die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter analysieren und darauf reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle Amazon EC2 Auto Scaling-Aktionen werden von der Amazon EC2 Auto Scaling API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Aufrufe der CreateLaunchConfigurationUpdateAutoScalingGroupAktionen, und generieren beispielsweise Einträge in den CloudTrail Protokolldateien. DescribeAutoScalingGroup

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder AWS Identity and Access Management (IAM-) Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anforderung aus einem anderen -Service gesendet wurde

Weitere Informationen finden Sie unter dem CloudTrail`userIdentity`Element.

Grundlegendes zu Amazon EC2 Auto Scaling-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen

Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die CreateLaunchConfiguration Aktion 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": "2018-08-21T17:07:49Z",
    "eventSource": "autoscaling.amazonaws.com",
    "eventName": "CreateLaunchConfiguration",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Coral/Jakarta",
    "requestParameters": {
        "ebsOptimized": false,
        "instanceMonitoring": {
            "enabled": false
        },
        "instanceType": "t2.micro",
        "keyName": "EC2-key-pair-oregon",
        "blockDeviceMappings": [
            {
                "deviceName": "/dev/xvda",
                "ebs": {
                    "deleteOnTermination": true,
                    "volumeSize": 8,
                    "snapshotId": "snap-01676e0a2c3c7de9e",
                    "volumeType": "gp2"
```

```
                }
            }
        ],
        "launchConfigurationName": "launch_configuration_1",
        "imageId": "ami-6cd6f714d79675a5",
        "securityGroups": [
            "sg-00c429965fd921483"
        ]
    },
    "responseElements": null,
    "requestID": "0737e2ea-fb2d-11e3-bfd8-99133058e7bb",
    "eventID": "3fcfb182-98f8-4744-bd45-b38835ab61cb",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Zugehörige Ressourcen

Mit CloudWatch Logs können Sie bestimmte Ereignisse überwachen und Warnmeldungen erhalten, die von erfasst wurden CloudTrail. Die an CloudWatch Logs gesendeten Ereignisse sind so konfiguriert, dass sie von Ihrem Trail protokolliert werden. Stellen Sie also sicher, dass Sie Ihren Trail oder Ihre Trails so konfiguriert haben, dass die Ereignistypen protokolliert werden, die Sie überwachen möchten. CloudWatch Mithilfe von Protokollen können Informationen in den Protokolldateien überwacht und Sie benachrichtigt werden, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch und im Thema Überwachung von CloudTrail Protokolldateien mit Amazon CloudWatch Logs im AWS CloudTrail Benutzerhandbuch.

Amazon SNS-Benachrichtigungsoptionen für Amazon EC2 Auto Scaling

Sie können Ihre Auto Scaling-Gruppe so konfigurieren, dass Sie über wichtige Ereignisse informiert werden, die sich auf Ihre Anwendung auswirken. Mit Benachrichtigungen können Sie auch Abfragen eliminieren, und Sie werden nicht auf den `RequestLimitExceeded` Fehler stoßen, der sich manchmal aus der Abfrage ergibt.

Es gibt zwei Möglichkeiten, Benachrichtigungen über Amazon EC2 Auto Scaling zu erhalten:

- Amazon Simple Notification Service – Amazon SNS kann Sie benachrichtigen, wenn Ihre Auto Scaling-Gruppe Instances startet oder beendet. Sie können Amazon SNS-Benachrichtigungen lediglich aktivieren oder deaktivieren. Weitere Informationen finden Sie unter Amazon SNS und Amazon EC2 Auto Scaling.
- Amazon EventBridge – EventBridge stellt erweiterte, ereignisgesteuerte Benachrichtigungen bereit, die bestimmten Kriterien entsprechen und an eine Vielzahl von Zielen gesendet werden, einschließlich Amazon SNS . EventBridge kann auch eine breitere Palette von Auto Scaling-Ereignissen überwachen, um eine genauere Überwachung zu ermöglichen. Weitere Informationen finden Sie unter Wird EventBridge zur Behandlung von Auto Scaling Scaling-Ereignissen verwendet.

Sie können auch eine benutzerdefinierte Aktion ausführen, wenn eine Instance beim Start oder Beenden in den Status „Ausstehend“ wechselt, indem Sie Lebenszyklus-Hooks und Services wie EventBridge, Amazon SNS und Amazon SQS verwenden. Lebenszyklus-Hooks können einer neuen Instance auch zusätzliche Zeit geben, um ein in den Benutzerdaten angegebenes Skript abzuschließen, bevor Amazon EC2 Auto Scaling die Instance zur Gruppe hinzufügt. Weitere Informationen finden Sie unter Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling.

Amazon SNS und Amazon EC2 Auto Scaling

In diesem Abschnitt wird gezeigt, wie Sie Amazon SNS verwenden, um zu überwachen, wann Ihre Auto Scaling-Gruppe Instances startet oder beendet.

Wenn Sie zum Beispiel Ihre Auto Scaling-Gruppe konfigurieren, die `autoscaling:EC2_INSTANCE_TERMINATE`-Benachrichtigungsweise zu benutzen, und Ihre Auto Scaling-Gruppe beendet eine Instance, sendet es eine Benachrichtigung per Mail. Diese E-Mail enthält die Details der beendeten Instance, z. B. die Instance-ID und den Grund für die Beendigung.

Beachten Sie, dass während Amazon EC2 Auto Scaling Instances zur Gruppe hinzufügt oder daraus entfernt, Benachrichtigungen über diese Änderungen an Sie gesendet werden, wobei eine Benachrichtigung pro Instance gesendet wird. Die Zustellung dieser Benachrichtigungen erfolgt jedoch nach bestem Wissen und Gewissen, und Ihre Instances könnten auch nach der ersten Benachrichtigung fehlschlagen, z. B. wenn eine spätere Zustandsprüfung fehlschlägt. Auch wenn Amazon EC2 Auto Scaling Sie zunächst benachrichtigt, könnte eine Instance später trotzdem ausfallen. Beachten Sie, dass Sie konfigurieren können, wie lange nach dem Start einer Instance Amazon EC2 Auto Scaling wartet, bevor die erste Zustandsprüfung durchgeführt wird. Weitere

Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.

Weitere Informationen zu Amazon SNS im Allgemeinen finden Sie im Amazon Simple Notification Service-Entwicklerhandbuch.

Inhalt

SNS-Benachrichtigungen

Amazon EC2 Auto Scaling unterstützt das Senden von Amazon SNS-Benachrichtigungen bei folgenden Ereignissen.

Ereignis	Beschreibung
`autoscaling:EC2_INSTANCE_LAUNCH`	Erfolgreiches Starten einer Instance
`autoscaling:EC2_INSTANCE_LAUNCH_ERROR`	Fehlgeschlagenes Starten einer Instance
`autoscaling:EC2_INSTANCE_TERMINATE`	Erfolgreiches Beenden einer Instance
`autoscaling:EC2_INSTANCE_TERMINATE_ERROR`	Fehlgeschlagenes Beenden einer Instance

Die Nachricht enthält die folgenden Informationen:

- `Event` – Das Ereignis
- `AccountId` – Die Konto-ID von Amazon Web Services.
- `AutoScalingGroupName` – Der Name der Auto Scaling-Gruppe.
- `AutoScalingGroupARN` – Der ARN der Auto Scaling-Gruppe.
- `EC2InstanceId` – Die ID der EC2-Instance.

Beispiel:

```
Service: AWS Auto Scaling
Time: 2016-09-30T19:00:36.414Z
RequestId: 4e6156f4-a9e2-4bda-a7fd-33f2ae528958
Event: autoscaling:EC2_INSTANCE_LAUNCH
AccountId: 123456789012
AutoScalingGroupName: my-asg
AutoScalingGroupARN: arn:aws:autoscaling:region:123456789012:autoScalingGroup...
ActivityId: 4e6156f4-a9e2-4bda-a7fd-33f2ae528958
Description: Launching a new EC2 instance: i-0598c7d356eba48d7
Cause: At 2016-09-30T18:59:38Z a user request update of AutoScalingGroup constraints
 to ...
StartTime: 2016-09-30T19:00:04.445Z
EndTime: 2016-09-30T19:00:36.414Z
StatusCode: InProgress
StatusMessage:
Progress: 50
EC2InstanceId: i-0598c7d356eba48d7
Details: {"Subnet ID":"subnet-id","Availability Zone":"zone"}
Origin: AutoScalingGroup
Destination: EC2
```

Konfigurieren von Amazon SNS-Benachrichtigungen für Amazon EC2 Auto Scaling

Damit Sie Amazon SNS zum Versenden von E-Mail-Benachrichtigungen verwenden können, müssen Sie zunächst ein Thema erstellen und es mit Ihren E-Mail-Adressen abonnieren.

Erstellen Sie ein Amazon SNS-Thema.

Ein SNS-Thema ist ein logischer Zugriffspunkt, ein Kommunikationskanal der Auto Scaling-Gruppe zum Versenden von Benachrichtigungen. Sie erstellen ein Thema, indem Sie einen Namen dafür angeben.

Wenn Sie einen Themanamen vergeben, muss der Name folgende Anforderungen erfüllen:

- Er muss zwischen 1 und 256 Zeichen lang sein.
- Er muss ASCII-Buchstaben mit Groß- und Kleinschreibung, Zahlen, Unterstriche oder Bindestriche enthalten.

Weitere Informationen finden Sie unter Amazon SNS-Thema anlegen im Amazon Simple Notification Service-Entwicklerhandbuch.

Amazon SNS-Thema abonnieren

Zum Empfangen der Benachrichtigungen, die die Auto-Scaling-Gruppe an das Thema sendet, müssen Sie einen Endpunkt für das Thema abonnieren. In diesem Verfahren, für Endpoint, geben Sie die E-Mailadresse an, unter der Sie die Benachrichtigungen von Amazon EC2 Auto Scaling erhalten möchten.

Weitere Informationen finden Sie unter Amazon SNS-Thema abonnieren im Amazon Simple Notification Service-Entwicklerhandbuch.

Bestätigen Ihres Amazon SNS-Abonnements

Amazon SNS sendet eine Bestätigungs-E-Mail an die E-Mail-Adresse, die Sie im vorherigen Schritt angegeben haben.

Öffnen Sie die E-Mail von AWS Notifications und klicken Sie auf den Link zur Bestätigung des Abonnements, bevor Sie mit dem nächsten Schritt fortfahren.

Sie erhalten eine Bestätigungsnachricht von AWS. Amazon SNS ist jetzt so konfiguriert, dass Benachrichtigungen empfangen und als E-Mail an die angegebene E-Mail-Adresse gesendet werden.

Konfigurieren Ihrer Auto Scaling-Gruppe zum Senden von Benachrichtigungen

Sie können die Auto Scaling-Gruppe so konfigurieren, dass im Falle von Skalierungsereignissen (z. B. Starten oder Beenden von Instances) Benachrichtigungen an Amazon SNS gesendet werden. Amazon SNS sendet eine Benachrichtigung mit Informationen zu den Instances an die E-Mail-Adresse, die Sie angegeben haben.

So konfigurieren Sie Amazon SNS-Benachrichtigungen für Ihre Auto Scaling-Gruppe (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.

2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet, in dem Informationen über die ausgewählte Gruppe angezeigt werden.
3. Klicken Sie auf der Registerkarte Aktivität Benachrichtigungen über Aktivitäten, Benachrichtigung erstellen.
4. Führen Sie im Bereich Create notifications die folgenden Schritte aus:

   a. Wählen Sie unter SNS-Thema das SNS-Thema aus.

   b. Wählen Sie unter Event types die Ereignisse aus, zu denen Benachrichtigungen gesendet werden sollen.

   c. Wählen Sie Erstellen.

So konfigurieren Sie Amazon SNS-Benachrichtigungen für Ihre Auto Scaling-Gruppe (AWS CLI)

Verwenden Sie den folgenden put-notification-configuration-Befehl.

```
aws autoscaling put-notification-configuration --auto-scaling-group-name my-
asg --topic-arn arn --notification-types "autoscaling:EC2_INSTANCE_LAUNCH"
 "autoscaling:EC2_INSTANCE_TERMINATE"
```

Testen der Benachrichtigung

Aktualisieren Sie die Auto Scaling-Gruppe, indem Sie die gewünschte Kapazität der Auto Scaling-Gruppe um 1 erhöhen, um eine Benachrichtigung für ein Startereignis zu generieren. Sie erhalten innerhalb weniger Minuten nach dem Start der Instance eine Benachrichtigung.

So ändern Sie die gewünschte Kapazität (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite Auto-Scaling-Gruppen wird ein geteilter Bereich geöffnet, in dem Informationen über die ausgewählte Gruppe angezeigt werden.
3. Wählen Sie auf der Registerkarte Details die Option Gruppendetails, Bearbeiten.

4. Erhöhen Sie für Desired capacity (Gewünschte Kapazität den aktuellen Wert um 1. Wenn dieser Wert die Maximum capacity (Maximalkapazität) überschreitet, müssen Sie auch den Wert der Maximum capacity (Maximalkapazität) um 1 erhöhen.
5. Wählen Sie Aktualisieren.
6. Nach einigen Minuten erhalten Sie eine Benachrichtigung über das Ereignis. Wenn Sie die zusätzliche Instance, die Sie für diesen Test gestartet haben, nicht benötigen, können Sie Desired capacity (Gewünschte Kapazität) um 1 reduzieren. Nach einigen Minuten erhalten Sie eine Benachrichtigung über das Ereignis.

Löschen der Benachrichtigungskonfiguration

Sie können Ihre Amazon EC2 Auto Scaling-Benachrichtigungskonfiguration löschen, wenn sie nicht mehr verwendet wird.

So löschen Sie die Amazon EC2 Auto Scaling-Benachrichtigungskonfiguration (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie Ihre Auto Scaling-Gruppe aus.
3. Klicken Sie auf der Registerkarte Aktivität auf das Kontrollkästchen neben der Benachrichtigung, die Sie löschen möchten, und wählen Sie dann Aktionen, Löschenaus.

So löschen Sie die Amazon EC2 Auto Scaling-Benachrichtigungskonfiguration (AWS CLI)

Verwenden Sie den folgenden delete-notification-configuration-Befehl.

```
aws autoscaling delete-notification-configuration --auto-scaling-group-name my-asg --
topic-arn arn
```

Informationen über die Löschung des Amazon SNS -Themas und aller mit Ihrer Auto Scaling-Gruppe verbundenen Abonnements finden Sie unter Löschen eines Amazon SNS-Abonnements und -Themas im Amazon Simple Notification Service-Entwicklerhandbuch.

Schlüsselrichtlinie für ein verschlüsseltes Amazon-SNS-Thema

Das von Ihnen angegebene Amazon-SNS-Thema ist möglicherweise mit einem vom Kunden verwalteten Schlüssel verschlüsselt, der mit dem AWS Key Management Service erstellt wurde.

Um Amazon EC2 Auto Scaling die Berechtigung zu erteilen, in verschlüsselten Themen zu veröffentlichen, müssen Sie zuerst Ihren KMS-Schlüssel erstellen und dann die folgende Anweisung zur Richtlinie für den KMS-Schlüssel hinzufügen. Ersetzen Sie den Beispiel-ARN durch den ARN der entsprechenden serviceverknüpften Rolle, der Zugriff auf den Schlüssel gewährt wird. Weitere Informationen erhalten Sie unter AWS KMS -Berechtigungen konfigurieren im Entwicklerhandbuch für Amazon Simple Notification Service.

In diesem Beispiel erteilt die Richtlinienanweisung der serviceverknüpften Rolle mit dem Namen AWSServiceRoleForAutoScaling Berechtigungen zur Verwendung des vom Kunden verwalteten Schlüssels. Weitere Informationen zur serviceverknüpften Rolle für Amazon EC2 Auto Scaling finden Sie unter Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling.

```
{
  "Sid": "Allow service-linked role use of the customer managed key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789012:role/aws-service-role/autoscaling.amazonaws.com/
AWSServiceRoleForAutoScaling"
  },
  "Action": [
    "kms:GenerateDataKey*",
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

Die Bedingungsschlüssel `aws:SourceArn` und `aws:SourceAccount` werden in Schlüsselrichtlinien nicht unterstützt, die es Amazon EC2 Auto Scaling ermöglichen, in verschlüsselten Themen zu veröffentlichen.

AWS in Amazon EC2 Auto Scaling integrierte Services

Amazon EC2 Auto Scaling kann in andere AWS Services integriert werden. Lesen Sie die folgenden Integrationsoptionen, um mehr darüber zu erfahren, wie jeder Dienst mit Amazon EC2 Auto Scaling funktioniert.

Themen

Verwenden des Kapazitätsausgleichs, um Amazon-EC2-Spot-Unterbrechungen zu behandeln

Sie können Amazon-EC2-Auto-Scaling zur Überwachung von Veränderungen und zur automatisierten Reaktion auf Änderungen, die sich auf die Verfügbarkeit Ihrer Spot-Instances auswirken, konfigurieren. Der Kapazitätsausgleich hilft Ihnen, die Verfügbarkeit von Workloads aufrechtzuerhalten, indem Sie Ihre Flotte proaktiv um eine neue Spot-Instance erweitern, bevor eine ausgeführte Instance durch Amazon EC2 unterbrochen wird.

Das Ziel des Kapazitätsausgleichs ist es, Ihre Workload ohne Unterbrechung weiter zu verarbeiten. Wenn Spot-Instances einem erhöhten Unterbrechungsrisiko ausgesetzt sind, benachrichtigt der Amazon-EC2-Spot-Service Amazon EC2 Auto Scaling und empfiehlt, eine EC2-Instance erneut auszugleichen.

Wenn Sie die Kapazitätsausgleichsempfehlung für Ihre Auto-Scaling-Gruppe aktivieren, versucht Amazon EC2 Auto Scaling, die Spot-Instances in Ihrer Gruppe, die eine Empfehlung für einen erneuten Ausgleich erhalten haben, proaktiv zu ersetzen. Dies gibt Ihnen die Möglichkeit Ihr Workload auf neue oder bestehende Spot-Instances auszugleichen, die nicht einem erhöhten Risiko einer Unterbrechung ausgesetzt sind. Ihr Workload kann die Arbeit weiter verarbeiten, während Amazon EC2 Auto Scaling eine neue Spot-Instance startet, bevor Ihre vorhandene Instance unterbrochen wird.

Wenn Sie keine Kapazitätswiederherstellungen verwenden, ersetzt Amazon EC2 Auto Scaling Spot-Instances nicht, bis der Amazon EC2 Spot-Services die Instances unterbricht und deren Zustandsprüfung fehlschlägt. Bevor Sie eine Instance unterbrechen, gibt Amazon EC2 immer sowohl eine EC2-Instance-Neuausgleichsempfehlung als auch zwei Minuten im Voraus einen Hinweis auf die Spot-Instance-Unterbrechung aus.

Inhalt

Übersicht

Um Kapazitätswiederherstellungen mit Ihrer Auto-Scaling-Gruppe zu verwenden, gehen Sie wie folgt vor:

1. Konfigurieren Sie Ihre Auto-Scaling-Gruppe für die Verwendung mehrerer Instance-Typen und Availability Zones. Auf diese Weise kann Amazon EC2 Auto Scaling die verfügbare Kapazität für Spot-Instances in jeder Availability Zone betrachten. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.
2. Fügen Sie bei Bedarf Lebenszyklus-Hooks hinzu, um Ihre Anwendung beim Skalieren innerhalb der Instances, die die Benachrichtigung zur erneuten Verteilung empfangen, ordnungsgemäß herunterzufahren. Weitere Informationen finden Sie unter Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling.

Im Folgenden finden Sie einige Gründe, warum Sie einen Lebenszyklus-Hook verwenden könnten:

- Für das ordnungsgemäße Herunterfahren von Amazon SQS-Workern
- Zum Abschließen der Abmeldung vom Domain Name System (DNS)
- Zum Abrufen und Hochladen von System- oder Anwendungsprotokollen in Amazon Simple Storage Service (Amazon S3)

3. Entwickeln Sie eine benutzerdefinierte Aktion für den Lebenszyklus-Hook. Um Ihre benutzerdefinierte Aktion so schnell wie möglich aufzurufen, müssen Sie wissen, wann eine Instance bereit ist, beendet zu werden. Sie finden dies heraus, indem Sie den Lebenszyklusstatus der Instance ermitteln.
   - Um eine Aktion außerhalb der Instanz aufzurufen, schreiben Sie eine EventBridge Regel und automatisieren Sie, welche Aktion ausgeführt werden soll, wenn ein Ereignismuster mit der Regel übereinstimmt.
   - Um eine Aktion innerhalb der Instance aufzurufen, konfigurieren Sie die Instance so, dass sie ein Beendigungsskript ausführt und den Lebenszyklusstatus über die Instance-Metadaten abruft.

   Es ist wichtig, die benutzerdefinierte Aktion so zu gestalten, dass sie in weniger als zwei Minuten abgeschlossen ist. Dadurch wird sichergestellt, dass genügend Zeit zur Verfügung steht, um Aufgaben zu erledigen, bevor die Instance beendet wird.

Nachdem Sie diese Schritte abgeschlossen haben, können Sie mit den Kapazitätswiederherstellungen beginnen.

Verhalten bei Kapazitätswiederherstellungen

Bei Kapazitätswiederherstellungen verhält sich Amazon EC2 Auto Scaling folgendermaßen, wenn eine Instance eine Empfehlung zur erneuten Verteilung erhält:

- Beim Starten einer neuen Spot Instance wartet Amazon EC2 Auto Scaling, bis die neue Instance ihre Zustandsprüfung besteht, bevor die vorherige Instance beendet wird. Wenn mehr als eine Instance ersetzt wird, beginnt die Beendigung jeder vorherigen Instance, nachdem die neue Instance gestartet wurde und ihre Zustandsprüfung bestanden hat.
- Da Amazon EC2 Auto Scaling vor dem Beenden der vorherigen Instances versucht, neue zu starten, kann das Wiederherstellen des Gleichgewichts beeinträchtigt und sogar gänzlich unterbrochen werden, falls die angegebene maximale Kapazität nahezu oder gänzlich erreicht

ist. Um dieses Problem zu vermeiden, kann Amazon EC2 Auto Scaling die maximale Größe der Gruppe vorübergehend bis zu 10 Prozent der gewünschten Kapazität überschreiten.

- Wenn Sie keinen Lebenszyklus-Hook zu Ihrer Auto-Scaling-Gruppe hinzufügen, beginnt Amazon EC2 Auto Scaling mit dem Beenden der vorherigen Instances, sobald die neuen Instances ihre Zustandsprüfung bestehen.
- Wenn Sie einen Lebenszyklus-Hook hinzugefügt haben, verlängert sich die Zeit, die benötigt wird, bis wir mit der Beendigung der vorherigen Instances beginnen, um den Timeout-Wert, den Sie für den Lebenszyklus-Hook angegeben haben.
- Wenn Sie Skalierungsrichtlinien oder eine geplante Skalierung verwenden, werden die Skalierungsaktivitäten parallel ausgeführt. Wenn eine Skalierungsaktivität ausgeführt wird und Ihre Auto-Scaling-Gruppe unter der neuen gewünschten Kapazität liegt, wird Amazon EC2 Auto Scaling zuerst skaliert, bevor die vorherigen Instances beendet werden.

Wenn in einer Availability Zone keine Kapazität für Ihre Instance-Typen vorhanden ist, versucht Amazon EC2 Auto Scaling weiterhin, Spot-Instances in anderen aktivierten Availability Zones zu starten, bis es erfolgreich ist.

Im Worst-Case-Szenario, wenn die neuen Instances nicht gestartet werden oder die Zustandsprüfung fehlschlägt, versucht Amazon EC2 Auto Scaling weiterhin, sie neu zu starten. Während es versucht, neue Instances zu starten, werden Ihre vorherigen schließlich unterbrochen und mit einer zweiminütigen Unterbrechungsmeldung zwangsweise beendet.

Überlegungen

Berücksichtigen Sie bei der Verwendung von Kapazitätswiederherstellungen die folgenden Punkte:

Gestalten Sie Ihre Anwendung so, dass sie Spot-Unterbrechungen toleriert

Ihre Anwendung sollte dynamische Änderungen in der Anzahl der Instances und die Möglichkeit, dass eine Spot-Instance frühzeitig unterbrochen wird, bewältigen können. Wenn sich z. B. die Auto-Scaling-Gruppe hinter einem Elastic Load Balancing Load Balancer befindet, wartet Amazon EC2 Auto Scaling darauf, dass die Instance vom Load Balancer abgemeldet wird, bevor Ihr Lebenszyklus-Hook aufgerufen wird. Wenn die Zeit zum Abmelden der Instance und zum Abschließen der Lebenszyklus-Aktion zu lange dauert, wird die Instance möglicherweise unterbrochen, während Amazon EC2 Auto Scaling auf den Abschluss Ihrer Lebenszyklus-Aktion wartet, bevor es die Instance beendet.

Es ist Amazon EC2 nicht immer möglich, das Signal für die Neuausgleichsempfehlung vor der zweiminütigen Spot-Instance-Unterbrechungsbenachrichtigung zu senden. Daher kann das Empfehlungssignal für eine erneute Verteilung manchmal zusammen mit der zweiminütigen Unterbrechungsbenachrichtigung eingehen. In diesem Fall ruft Amazon EC2 Auto Scaling den Lebenszyklus-Hook auf und versucht, sofort eine neue Spot-Instance zu starten.

Vermeiden Sie ein erhöhtes Risiko einer Unterbrechung von Ersatz-Spot-Instances

Ihre Ersatz-Spot-Instances haben möglicherweise ein erhöhtes Risiko einer Unterbrechung, wenn Sie die `lowest-price`-Zuweisungsstrategie verwenden. Das liegt daran, dass wir Instances im preisgünstigsten Pool starten, der zu diesem Zeitpunkt über verfügbare Kapazität verfügt, auch wenn Ihre Ersatz-Spot-Instances wahrscheinlich kurz nach dem Start unterbrochen werden. Um ein erhöhtes Unterbrechungsrisiko zu vermeiden, wird dringend empfohlen, die `lowest-price`-Zuweisungsstrategie nicht zu verwenden. Stattdessen empfehlen wir die `price-capacity-optimized`-Zuweisungsstrategie. Diese Strategie startet Ersatz-Spot-Instances in Spot-Pools, bei denen die Wahrscheinlichkeit einer Unterbrechung am geringsten ist und die den niedrigsten Preis haben. Daher ist es weniger wahrscheinlich, dass sie in naher Zukunft unterbrochen werden.

Amazon EC2 Auto Scaling startet eine neue Instance nur dann, wenn die Verfügbarkeit gleich oder besser ist

Eines der Ziele des Kapazitätsausgleichs ist die Verbesserung der Verfügbarkeit einer Spot Instance. Wenn eine vorhandene Spot Instance eine Neuausgleichsempfehlung erhält, startet Amazon EC2 Auto Scaling nur dann eine neue Instance, wenn die neue Instance dieselbe oder eine bessere Verfügbarkeit als die vorhandene Instance bietet. Wenn das Risiko einer Unterbrechung einer neuen Instance größer ist als das der vorhandenen Instance, startet Amazon EC2 Auto Scaling keine neue Instance. Amazon EC2 Auto Scaling wird die Spot-Kapazitätspools jedoch weiterhin auf der Grundlage der vom Amazon-EC2-Spot-Service bereitgestellten Informationen bewerten und eine neue Instance starten, falls sich die Verfügbarkeit verbessert.

Es besteht die Möglichkeit, dass Ihre vorhandene Instance unterbrochen wird, ohne dass Amazon EC2 Auto Scaling proaktiv eine neue Instance startet. In diesem Fall versucht Amazon EC2 Auto Scaling, eine neue Instance zu starten, sobald die Spot-Instance-Unterbrechungsmeldung eingeht. Das geschieht unabhängig davon, ob bei der neuen Instance ein hohes Unterbrechungsrisiko besteht.

Capacity Rebalancing erhöht nicht die Unterbrechungsrate Ihrer Spot-Instance

Wenn Sie Capacity Rebalancing aktivieren, wird Ihre Spot-Instance-Unterbrechungsrate (die Anzahl der Spot-Instances, die zurückgefordert werden, wenn Amazon EC2 die Kapazität zurück

benötigt) nicht erhöht. Wenn der Kapazitätsausgleich jedoch feststellt, dass bei einer Instance das Risiko einer Unterbrechung besteht, versucht Amazon EC2 Auto Scaling sofort, eine neue Instance zu starten. Daher werden möglicherweise mehr Instances ersetzt, als wenn Sie darauf gewartet hätten, dass Amazon EC2 Auto Scaling eine neue Instance startet, nachdem die gefährdete Instance unterbrochen wurde.

Sie können zwar mehr Instances mit aktivierten Kapazitätswiederherstellungen ersetzen, profitieren jedoch davon, dass Sie eher proaktiv als reaktiv sind. Dadurch haben Sie mehr Zeit, Maßnahmen zu ergreifen, bevor Ihre Instances unterbrochen werden. Mit einer Spot-Instance-Unterbrechungsbenachrichtigung haben Sie normalerweise nur bis zu zwei Minuten Zeit, um Ihre Instance ordnungsgemäß herunterzufahren. Wenn die Kapazitätswiederherstellungen eine neue Instance im Voraus starten, geben Sie vorhandenen Prozessen eine bessere Chance, auf Ihrer gefährdeten Instance abgeschlossen zu werden. Sie können auch mit dem Herunterfahren Ihrer Instance beginnen, verhindern, dass neue Arbeiten für Ihre gefährdete Instance geplant werden, und die neu gestartete Instance auf die Übernahme der Anwendung vorbereiten. Mit dem proaktiven Ersetzen durch Kapazitätswiederherstellungen profitieren Sie von einer reibungslosen Kontinuität.

Betrachten Sie als theoretisches Beispiel zur Demonstration der Risiken und Vorteile des Einsatzes von Kapazitätswiederherstellungen das folgende Szenario:

- 14:00 Uhr – Für Instance A wird eine Empfehlung zum erneuten Ausgleich empfangen, und Amazon EC2 Auto Scaling versucht sofort, eine Ersatz-Instance B zu starten, sodass Sie Zeit haben, Ihre Shutdown-Verfahren zu starten.
- 14:30 Uhr – Für Instance B wird eine Empfehlung zum erneuten Ausgleich empfangen, die durch Instance C ersetzt wird, sodass Sie Zeit haben, Ihre Shutdown-Verfahren zu starten.
- 14:32 Uhr — Wenn keine Kapazitätswiederherstellungen aktiviert wären und um 14:32 Uhr eine Spot-Instance-Unterbrechungsmeldung für Instance A eingegangen wäre, hätten Sie nur bis zu zwei Minuten Zeit gehabt, um Maßnahmen zu ergreifen. Instance A wäre jedoch bis zu diesem Zeitpunkt ausgeführt worden.

Aktivieren des Kapazitätsausgleichs (Konsole)

Sie können den Kapazitätsausgleich aktivieren oder deaktivieren, wenn Sie eine Auto-Scaling-Gruppe erstellen oder aktualisieren.

So aktivieren Sie den Kapazitätsausgleich für eine neue Auto-Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
3. Für Schritt 1: Startvorlage/-konfiguration auswählen, einen Namen für die Auto-Scaling-Gruppe eingeben, eine Startvorlage auswählen und dann die Option Weiter auswählen, um mit dem nächsten Schritt fortzufahren.
4. Für Schritt 2: Instance-Startoptionen auswählen, für die Anforderungen an den Instance-Typ Einstellungen auswählen, um eine gemischte Instance-Gruppe zu erstellen. Dazu gehören die Instance-Typen, die gestartet werden können, Instance-Kaufoptionen und Zuweisungsstrategien für Spot- und On-Demand-Instances. Standardmäßig sind diese Einstellungen nicht konfiguriert. Um sie zu konfigurieren, müssen Sie Override launch template (Startvorlage überschreiben) auswählen. Weitere Informationen zum Erstellen von Gruppen mit gemischten Instances finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.
5. Wählen Sie die gewünschten Optionen unter Netzwerk aus. Stellen Sie sicher, dass sich die Subnetze, die Sie verwenden möchten, in verschiedenen Availability Zones befinden.
6. Wählen Sie im Abschnitt Zuweisungsstrategien eine Spot-Zuweisungsstrategie aus. Um die Kapazitätswiederherstellungen zu aktivieren oder zu deaktivieren, müssen Sie das Kontrollkästchen unter Kapazitätswiederherstellungen aktivieren oder deaktivieren. Diese Option wird nur angezeigt, wenn Sie einen Prozentsatz der Auto-Scaling-Gruppe anfordern, der als Spot-Instances gestartet werden soll, im Abschnitt Instance-Kaufoptionen angeben.
7. Erstellen Sie die Auto-Scaling-Gruppe
8. (Optional) Fügen Sie nach Bedarf Lebenszyklus-Hooks hinzu. Weitere Informationen finden Sie unter Lebenszyklus-Hooks hinzufügen.

So aktivieren oder deaktivieren Sie den Kapazitätsausgleich für eine vorhandene Auto-Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe. Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Optionen Allocation strategies (Zuweisungsstrategien), Edit (Bearbeiten) aus.

4. Aktivieren oder deaktivieren Sie im Abschnitt Zuweisungsstrategien die Kapazitätswiederherstellungen, indem Sie das Kontrollkästchen unter Kapazitätswiederherstellungen aktivieren oder deaktivieren.
5. Wählen Sie Aktualisieren.

Aktivieren Sie den Kapazitätsneuausgleich (AWS CLI)

Die folgenden Beispiele zeigen, wie Sie Capacity AWS CLI Rebalancing mithilfe von aktivieren und deaktivieren können.

Geben Sie mit dem Befehl create-auto-scaling-group oder update-auto-scaling-group den folgenden Parameter an:

- `--capacity-rebalance/--no-capacity-rebalance`— Boolescher Wert, der angibt, ob Capacity Rebalancing aktiviert ist.

Bevor Sie den create-auto-scaling-group-Befehl aufrufen, benötigen Sie den Namen einer Startvorlage, die für die Verwendung mit einer Auto-Scaling-Gruppe konfiguriert ist. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.

> **Note**
>
> Das folgende Verfahren zeigt, wie Sie eine in JSON oder YAML formatierte Konfigurationsdatei verwenden. Wenn Sie AWS CLI Version 1 verwenden, müssen Sie eine Konfigurationsdatei im JSON-Format angeben. Wenn Sie AWS CLI Version 2 verwenden, können Sie eine Konfigurationsdatei angeben, die entweder in YAML oder JSON formatiert ist.

JSON

So erstellen und konfigurieren Sie eine neue Auto-Scaling-Gruppe

- Verwenden Sie den folgenden create-auto-scaling-group-Befehl, um eine neue Auto-Scaling-Gruppe zu erstellen und Kapazitätswiederherstellungen zu aktivieren. Durch diesen Befehl wird auf eine JSON-Datei als einziger Parameter für Ihre Auto-Scaling-Gruppe verwiesen.

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Wenn Sie noch nicht über eine CLI-Konfigurationsdatei verfügen, die eine Richtlinie für gemischte Instances angibt, erstellen Sie eine.

Fügen Sie die folgende Zeile zum übergeordneten JSON-Objekt in der Konfigurationsdatei hinzu.

```
{
    "CapacityRebalance": true
}
```

Im Folgenden sehen Sie ein Beispiel für eine `config.json`-Datei.

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredCapacity": 12,
    "MinSize": 12,
    "MaxSize": 15,
    "CapacityRebalance": true,
    "MixedInstancesPolicy": {
        "InstancesDistribution": {
            "OnDemandBaseCapacity": 0,
            "OnDemandPercentageAboveBaseCapacity": 25,
            "SpotAllocationStrategy": "price-capacity-optimized"
        },
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Default"
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
                {
                    "InstanceType": "m5.large"
                },
                {
```

```
                    "InstanceType": "m5a.large"
                },
                {
                    "InstanceType": "c4.large"
                },
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
                }
            ]
        }
    },
    "TargetGroupARNs": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-alb-target-group/943f017f100becff",
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

So erstellen und konfigurieren Sie eine neue Auto-Scaling-Gruppe

- Verwenden Sie den folgenden create-auto-scaling-group-Befehl, um eine neue Auto-Scaling-Gruppe zu erstellen und Kapazitätswiederherstellungen zu aktivieren. Dieser Befehl verweist auf eine YAML-Datei als einziger Parameter für Ihre Auto-Scaling-Gruppe.

  ```
  aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
  ```

  Fügen Sie der in YAML formatierten Konfigurationsdatei die folgende Zeile hinzu.

  ```
  CapacityRebalance: true
  ```

  Im Folgenden sehen Sie ein Beispiel für eine `config.yaml`-Datei.

  ```
  ---
  AutoScalingGroupName: my-asg
  ```

```
DesiredCapacity: 12
MinSize: 12
MaxSize: 15
CapacityRebalance: true
MixedInstancesPolicy:
  InstancesDistribution:
    OnDemandBaseCapacity: 0
    OnDemandPercentageAboveBaseCapacity: 25
    SpotAllocationStrategy: price-capacity-optimized
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
    - InstanceType: c3.large
    - InstanceType: m3.large
TargetGroupARNs:
- arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-alb-target-group/943f017f100becff
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

So aktivieren Sie den Kapazitätsausgleich für eine vorhandene Auto-Scaling-Gruppe

- Verwenden Sie den folgenden update-auto-scaling-group-Befehl, um den Kapazitätsausgleich zu aktivieren.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --capacity-rebalance
```

So überprüfen Sie, ob der Kapazitätsausgleich für eine Auto-Scaling-Gruppe aktiviert ist

- Verwenden Sie den folgenden auto-scaling-groups Befehl, um zu überprüfen, ob der Kapazitätsausgleich aktiviert ist, und um die Details anzuzeigen.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn",
            ...
            "CapacityRebalance": true
        }
    ]
}
```

So deaktivieren Sie den Neuausgleich der Kapazität

Verwenden Sie den update-auto-scaling-group-Befehl mit der Option `--no-capacity-rebalance`, um den Kapazitätsausgleich zu deaktivieren.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --no-capacity-rebalance
```

Zugehörige Ressourcen

Weitere Informationen zum Kapazitätsausgleich finden Sie im Compute-Blog unter Proaktive Verwaltung des Spot-Instance-Lebenszyklus mithilfe der neuen Capacity Rebalancing-Funktion für EC2 Auto Scaling. AWS

Weitere Informationen zu den Empfehlungen zur Neuverteilung von EC2-Instances finden Sie unter Empfehlungen zur Neuverteilung von EC2-Instances im Amazon EC2-Benutzerhandbuch.

Weitere Informationen zu Lebenszyklus-Hooks finden Sie in den folgenden Ressourcen.

- Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft(verwenden) EventBridge
- Tutorial: Konfigurieren Sie Benutzerdaten zum Abrufen des Ziellebenszyklusstatus über Instance-Metadaten

Einschränkungen

- Amazon EC2 Auto Scaling kann die Instance, welche die Neuausgleichsbenachrichtigung erhält, nur beenden, wenn die Instance nicht vor dem Abskalieren geschützt ist. Der Abskalieren-Schutz verhindert jedoch nicht, dass eine Beendigung aufgrund einer Spot-Unterbrechung erfolgt. Weitere Informationen finden Sie unter Instance-Abskalierungsschutz verwenden.
- Support für Kapazitätswiederherstellungen ist in allen kommerziellen AWS-Regionen verfügbar, in denen Amazon EC2 Auto Scaling verfügbar ist, mit Ausnahme der Region Naher Osten (VAE).

Verwenden Sie On-Demand-Kapazitätsreservierungen, um Kapazitäten in bestimmten Availability Zones zu reservieren.

Mit Amazon EC2 On-Demand-Kapazitätsreservierungen können Sie Rechenkapazität in bestimmten Availability Zones reservieren. Um mit der Nutzung von Kapazitätsreservierungen zu beginnen, erstellen Sie eine Kapazitätsreservierung in einer bestimmten Availability Zone. Nun können Sie Instances in der reservierten Kapazität starten, die Kapazitätsauslastung in Echtzeit anzeigen und die Kapazität nach Bedarf erhöhen oder verringern.

Kapazitätsreservierungen werden entweder als `open` oder `targeted` konfiguriert. Wenn die Kapazitätsreservierung `open` ist, werden neue Instances und vorhandene Instances mit übereinstimmenden Attributen automatisch in der Kapazität der Kapazitätsreservierung ausgeführt. Wenn die Kapazitätsreservierung `targeted` ist, müssen die Instances speziell für die Ausführung in der reservierten Kapazität ausgerichtet sein.

In diesem Thema wird gezeigt, wie eine Auto-Scaling-Gruppe erstellt wird, die On-Demand-Instanzen in `targeted` Kapazitätsreservierungen einführt. So haben Sie mehr Kontrolle darüber, wann Sie bestimmte Kapazitätsreservierungen verwenden.

Die grundlegenden Schritte sind:

1. Erstellen Sie Kapazitätsreservierungen in mehreren Availability Zones, die denselben Instance-Typ, dieselbe Plattform und dieselbe Instance-Anzahl haben.
2. Gruppenkapazitätsreservierungen mithilfe von AWS Resource Groups.
3. Erstellen Sie eine Auto-Scaling-Gruppe mit einer Startvorlage, die auf die Ressourcengruppe abzielt, und verwenden Sie dabei dieselben Availability Zones wie die Kapazitätsreservierungen.

Inhalt

Schritt 1: Erstellen von Kapazitätsreservierungen

Der erste Schritt besteht darin, in jeder Availability Zone, in der Ihre Auto-Scaling-Gruppe bereitgestellt wird, eine Kapazitätsreservierung zu erstellen.

> **Note**
>
> Sie können `targeted`-Reservierungen nur erstellen, wenn Sie die Kapazitätsreservierungen zum ersten Mal erstellen.

Console

So erstellen Sie Kapazitätsreservierungen

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie Kapazitätsreservierungen (Kapazitätsreservierungen) und dann Create Kapazitätsreservierung (Kapazitätsreservierung erstellen) aus.
3. Konfigurieren Sie auf der Seite zum Erstellen einer Kapazitätsreservierung die folgenden Einstellungen im Abschnitt Instance-details: Der Instance-Typ, die Plattform und die Availability Zone der Instances, die Sie starten, müssen dem Instance-Typ, der Plattform und der Availability Zone entsprechen, die Sie hier angegeben haben. Andernfalls wird die Kapazitätsreservierung nicht angewendet.

    a. Wählen Sie unter Instance-Typ den Instance-Typ, mit dem in der reservierten Kapazität gestartet werden soll.

    b. Wählen Sie unter Plattform das Betriebssystem für Ihre Instances.

    c. Wählen Sie unter Availability Zone die erste Availability Zone aus, in der Sie Kapazität reservieren möchten.

d. Wählen Sie unter Gesamtkapazität die Anzahl der Instances aus, die Sie benötigen. Berechnen Sie die Gesamtzahl der Instances, die Sie für Ihre Auto-Scaling-Gruppe benötigen, geteilt durch die Anzahl der Availability Zones, die Sie verwenden möchten.

4. Wählen Sie unter Details der Kapazitätsreservierung eine der folgenden Optionen aus:

   - Zu einem bestimmten Zeitpunkt — Stornieren Sie die Kapazitätsreservierung automatisch zum angegebenen Datum und zur angegebenen Uhrzeit.
   - Manuell — Reservieren Sie die Kapazität, bis Sie sie ausdrücklich stornieren.

5. Wählen Sie für Instance-Eignung die Option Gezielt: Nur Instances aus, die auf die Kapazitätsreservierung abzielen.
6. (Optional) Geben Sie für Tags alle Tags an, die mit der Kapazitätsreservierung verknüpft werden sollen.
7. Wählen Sie Create (Erstellen) aus.
8. Notieren Sie unter Kapazitätsreservierung eine der folgenden Optionen aus: Sie benötigen sie, um die Gruppe „Kapazitätsreservierung“ einzurichten.

Wiederholen Sie dieses Verfahren für jede Availability Zone, die Sie für Ihre Auto- Scaling-Gruppe aktivieren möchten, und ändern Sie dabei nur den Wert der Availability Zone-Option.

AWS CLI

So erstellen Sie Kapazitätsreservierungen

Verwenden Sie den Befehl create-capacity-reservation (Kapazitätsreservierung erstellen), um die Kapazitätsreservierungen zu erstellen. Ersetzen Sie die Beispielwerte für `--availability-zone`, `--instance-type`, `--instance-platform` und `--instance-count`.

```
aws ec2 create-capacity-reservation \
    --availability-zone us-east-1a \
    --instance-type c5.xlarge \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

Beispiel für die resultierende ID der Kapazitätsreservierung

```
{
```

```
    "CapacityReservation": {
        "CapacityReservationId": "cr-1234567890abcdef1",
        "OwnerId": "123456789012",
        "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789012:capacity-
reservation/cr-1234567890abcdef1",
        "InstanceType": "c5.xlarge",
        "InstancePlatform": "Linux/UNIX",
        "AvailabilityZone": "us-east-1a",
        "Tenancy": "default",
        "TotalInstanceCount": 3,
        "AvailableInstanceCount": 3,
        "EbsOptimized": false,
        "EphemeralStorage": false,
        "State": "active",
        "StartDate": "2023-07-26T21:36:14+00:00",
        "EndDateType": "unlimited",
        "InstanceMatchCriteria": "targeted",
        "CreateDate": "2023-07-26T21:36:14+00:00"
    }
}
```

Notieren Sie unter Kapazitätsreservierung eine der folgenden Optionen aus: Sie benötigen sie, um die Gruppe „Kapazitätsreservierung“ einzurichten.

Wiederholen Sie diesen Befehl für jede Availability Zone, die Sie für Ihre Auto- Scaling-Gruppe aktivieren möchten, und ändern Sie dabei nur den Wert der Availability Zone`--availability-zone`-Option.

Schritt 2: Erstellen einer Gruppe für Kapazitätsreservierung

Wenn Sie mit der Erstellung der Kapazitätsreservierungen fertig sind, können Sie sie mithilfe des AWS Resource Groups-Dienstes gruppieren. AWS Resource Groups unterstützt verschiedene Gruppentypen für unterschiedliche Zwecke. Amazon EC2 verwendet eine spezielle Gruppe, die als serviceverknüpfte Ressourcengruppe bezeichnet wird, um eine Gruppe von Kapazitätsreservierungen gezielt anzusprechen. Um mit dieser serviceverknüpften Ressourcengruppe zu interagieren, können Sie das AWS CLI oder ein SDK verwenden, aber nicht die Konsole. Weitere Informationen zu serviceverknüpften Ressourcengruppen finden Sie unter Dienstkonfigurationen für Ressourcengruppen im AWS Benutzerhandbuch Ressourcengruppen.

Um eine Kapazitätsreservierungsgruppe mit dem zu erstellen AWS CLI

Verwenden Sie den Befehl create-group (Gruppe erstellen), um eine Ressourcengruppe zu erstellen, die nur Kapazitätsreservierungen enthalten kann. In diesem Beispiel hat die Ressourcengruppe den Namen *my-cr-group*.

```
aws resource-groups create-group \
    --name my-cr-group \
    --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
 '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-
types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-east-1:123456789012:group/my-cr-group",
        "Name": "my-cr-group"
    },
    "GroupConfiguration": {
        "Configuration": [
            {
                "Type": "AWS::EC2::CapacityReservationPool"
            },
            {
                "Type": "AWS::ResourceGroups::Generic",
                "Parameters": [
                    {
                        "Name": "allowed-resource-types",
                        "Values": [
                            "AWS::EC2::CapacityReservation"
                        ]
                    }
                ]
            }
        ],
        "Status": "UPDATE_COMPLETE"
    }
}
```

Notieren Sie den ARN der neuen Ressourcengruppe. Sie benötigen ihn, um die Startvorlage für Ihre Auto-Scaling-Gruppe einzurichten.

Um Ihre Kapazitätsreservierungen mit der neu erstellten Gruppe zu verknüpfen, verwenden Sie AWS CLI

Verwenden Sie den folgenden Befehl group-resources (Gruppenressourcen), um die Kapazitätsreservierungen der neu erstellten Kapazitätsreservierungs-Gruppe zuzuordnen. Geben Sie für die `--resource-arns`-Option die Kapazitätsreservierungen anhand ihrer ARNs an. Konstruieren Sie die ARNs unter Verwendung der entsprechenden Region, Ihrer Konto-ID und der Reservierungs-IDs, die Sie zuvor notiert haben. In diesem Beispiel werden die Reservierungen mit den IDs *cr-1234567890abcdef1* und *cr-54321abcdef567890* in der Gruppe mit dem Namen *my-cr-group* zusammengefasst.

```
aws resource-groups group-resources \
    --group my-cr-group \
    --resource-arns \
      arn:aws:ec2:region:account-id:capacity-reservation/cr-1234567890abcdef1 \
      arn:aws:ec2:region:account-id:capacity-reservation/cr-54321abcdef567890
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
   "Succeeded": [
   "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1",
   "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"
   ],
   "Failed": [],
   "Pending": []
}
```

Informationen zum Ändern oder Löschen der Ressourcengruppe finden Sie in der API-Referenz für AWS Ressourcengruppen.

Schritt 3: Eine Startvorlage erstellen

Console

Eine Startvorlage erstellen

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Launch Templates aus.

3. Wählen Sie Startvorlage erstellen. Geben Sie einen Namen und eine Beschreibung für die anfängliche Version der Startvorlage ein.
4. Unter Auto-Scaling-Anleitung aktivieren Sie das Kontrollkästchen.
5. Erstellen Sie die Startvorlage. Wählen Sie ein AMI und einen Instance-Typ, die den Kapazitätsreservierungen entsprechen, die Sie verwenden möchten, und optional ein Schlüsselpaar, eine oder mehrere Sicherheitsgruppen und alle zusätzlichen EBS-Volumes oder Instance-Speicher-Volumes für Ihre Instances.
6. Erweitern Sie erweiterte Einstellungen und tun Sie Folgendes:

   a. Wählen Sie für Kapazitätsreservierung die Option Ziel nach Gruppe aus.

   b. Wählen Sie unter Kapazitätsreservierung – Zielgruppenspezifisch die Gruppe Kapazitätsreservierungen aus, die Sie im vorherigen Abschnitt erstellt haben, und klicken Sie dann auf Speichern.
7. Wählen Sie Startvorlage erstellen.
8. Wählen Sie auf der Bestätigungsseite Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

AWS CLI

Eine Startvorlage erstellen

Verwenden Sie den folgenden Befehl create-launch-template (Startvorlage erstellen), um eine Startvorlage zu erstellen, die angibt, dass die Kapazitätsreservierung auf eine bestimmte Ressourcengruppe abzielt. Ersetzen Sie den Beispielwert für `--launch-template-name`. Ersetzen Sie *c5.xlarge* durch den Instance-Typ, den Sie bei der Kapazitätsreservierung verwendet haben, und *ami-0123456789EXAMPLE* durch die ID des AMI, das Sie verwenden möchten. Ersetzen Sie *arn:aws:resource-groups:region:account-id:group/my-cr-group* durch den ARN der Ressourcengruppe, die Sie am Anfang des vorherigen Abschnitts erstellt haben.

```
aws ec2 create-launch-template \
    --launch-template-name my-launch-template \
    --launch-template-data \
        '{"InstanceType": "c5.xlarge",
          "ImageId": "ami-0123456789EXAMPLE",
          "CapacityReservationSpecification":
            {"CapacityReservationTarget":
```

```
                { "CapacityReservationResourceGroupArn": "arn:aws:resource-
groups:region:account-id:group/my-cr-group" }
            }
        }'
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-0dd77bd41dEXAMPLE",
        "LaunchTemplateName": "my-launch-template",
        "CreateTime": "2023-07-26T21:42:48+00:00",
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

Schritt 4: Erstellen einer Auto-Scaling-Gruppe

Console

Erstellen Sie Ihre Auto-Scaling-Gruppe wie gewohnt, aber wenn Sie Ihre VPC-Subnetze auswählen, wählen Sie aus jeder Availability Zone ein Subnetz aus, das den von Ihnen erstellten `targeted`-Kapazitätsreservierungen entspricht. Wenn Ihre Auto-Scaling-Gruppe dann eine On-Demand-Instance in einer dieser Availability Zones startet, wird die Instance in der reservierten Kapazität für diese Availability Zone ausgeführt. Wenn der Ressourcengruppe die Kapazitätsreservierungen ausgehen, bevor Ihre gewünschte Kapazität erreicht ist, starten wir alles, was über die reservierte Kapazität hinausgeht, als reguläre On-Demand-Kapazität.

So erstellen Sie eine simple Auto Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm dieselbe aus, AWS-Region die Sie bei der Erstellung der Startvorlage verwendet haben.
3. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
4. Geben Sie auf der Seite Startvorlage oder -konfiguration auswählen für Auto-Scaling-Gruppenname einen Namen für Ihre Auto-Scaling-Gruppe ein.

5. Wählen Sie für Launch template (Startvorlage) eine vorhandene Startvorlage aus.
6. Wählen Sie unter Launch template version (Version der Startvorlage) aus, ob die Auto-Scaling-Gruppe beim horizontalen Skalieren nach oben die standardmäßige, die neueste oder eine bestimmte Version der Startvorlage verwenden soll.
7. Überspringen Sie auf der Seite Startoptionen für die Instance auswählen den Abschnitt Anforderungen an den Instance-Typ, um den EC2-Instance-Typ zu verwenden, der in der Startvorlage angegeben ist.
8. Wählen Sie unter Netzwerk für VPC eine VPC. Die Auto-Scaling-Gruppe muss in derselben VPC erstellt werden wie die Sicherheitsgruppe, die Sie in Ihrer Startvorlage angegeben haben. Wenn Sie in Ihrer Startvorlage keine Sicherheitsgruppe angegeben haben, können Sie eine beliebige VPC auswählen, deren Teilnetze sich in denselben Availability Zones befinden wie Ihre Kapazitätsreservierungen.
9. Wählen Sie für , Availability Zones und Subnetze aus jeder Availability Zone aus, die Sie einbeziehen möchten, je nachdem, in welchen Availability Zones sich Ihre Kapazitätsreservierungen befinden.
10. Klicken Sie zweimal auf Weiter.
11. Geben Sie unter Konfigurieren von Gruppengröße und Skalierungsrichtlinien für Gewünschte Kapazität die anfängliche Anzahl von Instances ein, die gestartet werden sollen. Wenn Sie diese Zahl in einen Wert außerhalb der minimalen oder maximalen Kapazitätsgrenzen ändern, müssen Sie die Werte Mindestkapazität oder Maximalkapazitätaktualisieren. Weitere Informationen finden Sie unter Festlegen von Skalierungslimits für Ihre Auto-Scaling-Gruppe.
12. Wählen Sie Skip to review (Mit Prüfen fortfahren) aus.
13. Wählen Sie auf der Seite Review (Prüfen) Create Auto Scaling group (Auto-Scaling-Gruppe erstellen) aus.

AWS CLI

So erstellen Sie eine simple Auto Scaling-Gruppe

Verwenden Sie den folgenden Befehl create-auto-scaling-group und geben Sie den Namen und die Version Ihrer Startvorlage als Wert für die Option `--launch-template` an. Ersetzen Sie die Beispielwerte für `--auto-scaling-group-name`, `--min-size`, `--max-size` und `--vpc-zone-identifier`.

Geben Sie für die Option `--availability-zones` die Availability Zones an, für die Sie Kapazitätsreservierungen erstellt haben. Wenn Ihre Kapazitätsreservierungen beispielsweise

die Zonen `us-east-1a` und `us-east-1b` Availability Zones angeben, müssen Sie Ihre Auto-Scaling-Gruppe in denselben Zonen erstellen. Wenn Ihre Auto-Scaling-Gruppe dann eine On-Demand-Instance in einer dieser Availability Zones startet, wird die Instance in der reservierten Kapazität für diese Availability Zone ausgeführt. Wenn der Ressourcengruppe die Kapazitätsreservierungen ausgehen, bevor Ihre gewünschte Kapazität erreicht ist, starten wir alles, was über die reservierte Kapazität hinausgeht, als reguläre On-Demand-Kapazität.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateName=my-launch-template,Version='1' \
    --min-size 6 \
    --max-size 6 \
    --vpc-zone-identifier "subnet-5f46ec3b,subnet-0ecac448" \
    --availability-zones us-east-1a us-east-1b
```

Zugehörige Ressourcen

Eine Beispielimplementierung finden Sie in der AWS CloudFormation Vorlage im folgenden AWS GitHub Beispiel-Repository: https://github.com/aws-samples/aws-auto-scaling-backed-by-on-demand-capacity-reservations/.

Die folgenden verwandten Themen können hilfreich sein, wenn Sie mehr über Kapazitätsreservierungen erfahren.

- On-Demand Capacity Reservations
  - Erstellen Sie eine Kapazitätsreservierung im Amazon EC2 EC2-Benutzerhandbuch
  - Kapazitätsreservierungen auf Abruf im Amazon EC2 EC2-Benutzerhandbuch
  - Nehmen Sie im AWS Cloud Operations & Migrations Blog eine Gruppe von Amazon EC2 EC2-On-Demand-Kapazitätsreservierungen ins Visier
- Kapazitätsblöcke (Kapazitätsreservierungen mit einer definierten Dauer)
  - Kapazitätsblöcke für ML im Amazon EC2 EC2-Benutzerhandbuch
  - Capacity BlocksFür Machine-Learning-Workloads verwenden

Erstellen Sie Auto Scaling Scaling-Gruppen über die Befehlszeile mit AWS CloudShell

Unter unterstützt können Sie AWS CLI Befehle ausführen AWS-Regionen, indem AWS CloudShell Sie eine browserbasierte, vorauthentifizierte Shell verwenden, die direkt von der aus gestartet wird. AWS Management Console Sie können AWS CLI Befehle für Dienste ausführen, indem Sie Ihre bevorzugte Shell (Bash- oder Z-Shell) verwenden. PowerShell

Sie können AWS CloudShell von der aus starten, AWS Management Console indem Sie eine der folgenden beiden Methoden verwenden:

- 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.

Erstellen von Auto-Scaling-Gruppen mit AWS CloudFormation

Amazon EC2 Auto Scaling ist in einen Service integriert AWS CloudFormation, der Sie bei der Modellierung und Einrichtung Ihrer AWS Ressourcen unterstützt, sodass Sie weniger Zeit mit der Erstellung und Verwaltung Ihrer Ressourcen und Infrastruktur verbringen müssen. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen beschreibt (z. B. Auto Scaling Scaling-Gruppen) und diese Ressourcen für Sie AWS CloudFormation bereitstellt und konfiguriert.

Wenn Sie es verwenden AWS CloudFormation, können Sie Ihre Vorlage wiederverwenden, um Ihre Amazon EC2 Auto Scaling Scaling-Ressourcen konsistent und wiederholt einzurichten. Beschreiben Sie Ihre Ressourcen einmal und stellen Sie dann dieselben Ressourcen immer wieder in mehreren Regionen AWS-Konten bereit.

Amazon EC2 Auto Scaling und Vorlagen AWS CloudFormation

Um Ressourcen für Amazon EC2 Auto Scaling und damit verbundene Dienste bereitzustellen und zu konfigurieren, müssen Sie AWS CloudFormation -Vorlagen 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 die ersten Schritte mit Vorlagen zu erleichtern. AWS CloudFormation Weitere Informationen finden Sie unter Was ist AWS CloudFormation Designer? im AWS CloudFormation Benutzerhandbuch.

Bevor Sie mit der Erstellung Ihrer eigenen Stack-Vorlagen für Amazon EC2 Auto Scaling beginnen, führen Sie die folgenden Aufgaben durch:

- Erstellen Sie eine Startvorlage mit AWS::EC2::LaunchTemplate.
- Erstellen Sie mithilfe von Group Group eine Auto Scaling AWS::AutoScaling::AutoScaling.

Eine exemplarische Vorgehensweise, die Ihnen zeigt, wie Sie eine Auto-Scaling-Gruppe hinter einem Application Load Balancer bereitstellen, finden Sie unter Exemplarische Vorgehensweise: Erstellen einer skalierten Anwendung mit Lastenausgleich im AWS CloudFormation -Benutzerhandbuch.

Weitere nützliche Beispiele für Vorlagenausschnitte, mit denen Auto Scaling Scaling-Gruppen erstellt werden, und verwandte Ressourcen finden Sie in den folgenden Abschnitten des AWS CloudFormation Benutzerhandbuchs:

- Referenz zum Amazon EC2 Auto Scaling Scaling-Ressourcentyp Referenz zum
- Konfigurieren Sie Amazon EC2 Auto Scaling Scaling-Ressourcen mit AWS CloudFormation

Erfahren Sie mehr über AWS CloudFormation

Weitere Informationen AWS CloudFormation finden Sie in den folgenden Ressourcen:

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

Wird verwendet AWS Compute Optimizer , um Empfehlungen für den Instance-Typ für eine Auto Scaling Scaling-Gruppe abzurufen

AWS bietet Amazon EC2 EC2-Instance-Empfehlungen, um Ihnen zu helfen, die Leistung zu verbessern, Geld zu sparen oder beides zu tun, indem Sie Funktionen verwenden, die von bereitgestellt werden AWS Compute Optimizer. Mit diesen Empfehlungen können Sie entscheiden, ob Sie zu einem neuen Instance-Typ wechseln möchten.

Um Empfehlungen abzugeben, analysiert Compute Optimizer Ihre vorhandenen Instance-Spezifikationen und den letzten Metrikverlauf. Die kompilierten Daten werden dann verwendet, um zu empfehlen, welche Amazon-EC2-Instance-Typen am besten für die Verarbeitung der vorhandenen Leistungs-Workload optimiert sind. Empfehlungen werden zusammen mit den Preisen der Instance pro Stunde zurückgegeben.

> **Note**
>
> Um Empfehlungen von Compute Optimizer zu erhalten, müssen Sie sich zunächst bei Compute Optimizer anmelden. Weitere Informationen finden Sie unter Erste Schritte in AWS Compute Optimizer im AWS Compute Optimizer -Benutzerhandbuch.

Inhalt

Einschränkungen

Compute Optimizer generiert Empfehlungen für Instances in Auto-Scaling-Gruppen, die zum Starten und Ausführen von M-, C-, R-, T- und X-Instance-Typen konfiguriert sind. Es werden jedoch keine Empfehlungen für -g-Instance-Typen generiert, die auf AWS Graviton2-Prozessoren (z. B. C6g) basieren, und für -n-Instance-Typen, die eine höhere Netzwerkbandbreitenleistung aufweisen (z. B. M5n).

Die Auto-Scaling-Gruppen müssen auch so konfiguriert sein, dass sie einen einzelnen Instance-Typ ausführen (d. h. keine gemischten Instance-Typen), dürfen keiner Skalierungsrichtlinie zugeordnet sein und müssen dieselben Werte für die gewünschte, minimale und maximale Kapazität aufweisen (d. h. eine Auto-Scaling-Gruppe mit einer festen Anzahl von Instances). Compute Optimizer generiert Empfehlungen für Instances in Auto-Scaling-Gruppen, die alle dieser Konfigurationsanforderungen erfüllen.

Funde

Compute Optimizer klassifiziert die Ergebnisse für Auto-Scaling-Gruppen wie folgt:

- Nicht optimiert – Eine Auto-Scaling-Gruppe gilt als nicht optimiert, wenn Compute Optimizer eine Empfehlung identifiziert hat, die eine bessere Leistung für Ihr Workload bieten kann.
- Optimiert – Eine Auto-Scaling-Gruppe wird als optimiert angesehen, wenn Compute Optimizer feststellt, dass die Gruppe korrekt bereitgestellt ist, um Ihr Workload auszuführen, basierend auf dem gewählten Instance-Typ. Für optimierte Ressourcen empfiehlt Compute Optimizer manchmal einen Instance-Typ der neuen Generation.
- Keine – Für diese Auto-Scaling-Gruppe liegen keine Empfehlungen vor. Dies kann vorkommen, wenn Sie bei Computer Optimizer weniger als 12 Stunden angemeldet waren oder die Auto-Scaling-Gruppe weniger als 30 Stunden ausgeführt wurde oder wenn dieAuto-Scaling-Gruppe oder der Instance-Typ von Compute Optimizer nicht unterstützt wird. Weitere Informationen finden Sie im Abschnitt Einschränkungen.

Anzeigen von Empfehlungen

Nachdem Sie sich für Compute Optimizer entschieden haben, können Sie die Ergebnisse und Empfehlungen anzeigen, die für Ihre Auto-Scaling-Gruppen generiert werden. Wenn Sie sich kürzlich angemeldet haben, werden Empfehlungen möglicherweise bis zu 12 Stunden nicht angezeigt.

So zeigen Sie Empfehlungen an, die für eine Auto-Scaling-Gruppe generiert wurden

1. Öffnen Sie die Compute-Optimizer-Konsole unter https://console.aws.amazon.com/compute-optimizer/.

   Die Dashboard-Seite wird geöffnet.
2. Wählen Sie View recommendations for all Auto Scaling groups (Empfehlungen für alle Auto-Scaling-Gruppen anzeigen) aus.

3. Wählen Sie Ihre Auto-Scaling-Gruppe aus.
4. Wählen Sie die Option View details (Details anzeigen) aus.

   Die Ansicht ändert sich, um bis zu drei verschiedene Instance-Empfehlungen in einer vorkonfigurierten Ansicht anzuzeigen, basierend auf den Standard-Tabelleneinstellungen. Es stellt auch aktuelle CloudWatch Metrikdaten (durchschnittliche CPU-Auslastung, durchschnittlicher Netzwerkeingang und durchschnittlicher Netzwerkausgang) für die Auto Scaling Scaling-Gruppe bereit.

Legen Sie fest, ob Sie eine der Empfehlungen verwenden möchten. Entscheiden Sie, ob Sie die Leistungssteigerung, Kostensenkung oder beides optimieren möchten.

Um den Instance-Typ in Ihrer Auto-Scaling-Gruppe zu ändern, aktualisieren Sie die Startvorlage oder aktualisieren Sie die Auto-Scaling-Gruppe so, dass sie eine neue Startkonfiguration verwendet. Für bestehende Instances wird weiterhin die vorherige Konfiguration verwendet. Um die vorhandenen Instances zu aktualisieren, beenden Sie sie, damit sie durch Ihre Auto-Scaling-Gruppe ersetzt werden. Sie können auch zulassen, dass die Auto-Scaling-Gruppe ältere Instances schrittweise durch neuere Instances basierend auf Ihren Beendigungsrichtlinien ersetzt.

> **Note**
>
> Mit den Funktionen für maximale Instance-Lebensdauer und Instance-Aktualisierung können Sie auch vorhandene Instances in Ihrer Auto-Scaling-Gruppe ersetzen, um neue Instances zu starten, die die neue Startvorlage oder Startkonfiguration verwenden. Weitere Informationen finden Sie unter Auto-Scaling-Instances basierend auf der maximalen Instance-Lebensdauer ersetzen und Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren.

Überlegungen zur Bewertung der Empfehlungen

Bevor Sie zu einem neuen Instance-Typ wechseln, sollten Sie Folgendes beachten:

- Die Empfehlungen prognostizieren nicht Ihre Nutzung. Die Empfehlungen basieren auf Ihrer bisherigen Nutzung während des letzten 14-Tage-Zeitraums. Stellen Sie sicher, dass Sie einen Instance-Typ auswählen, der Ihren zukünftigen Verwendungsanforderungen entspricht.
- Konzentrieren Sie sich auf die grafisch dargestellten Metriken, um zu ermitteln, ob die tatsächliche Nutzung geringer als die Instance-Kapazität ist. Sie können auch Metrikdaten (Durchschnitt,

Spitze, Perzentil) einsehen, CloudWatch um Ihre EC2-Instance-Empfehlungen weiter auszuwerten. Beachten Sie zum Beispiel, wie sich die prozentualen CPU-Prozentsatzmetriken im Laufe des Tages verändern und ob es Datenverkehrsspitzen gibt, die berücksichtigt werden müssen. Weitere Informationen finden Sie unter Verfügbare Messwerte anzeigen im CloudWatch Amazon-Benutzerhandbuch.

- Compute Optimizer bietet möglicherweise Empfehlungen für Instances mit Spitzenlastleistung, bei denen es sich um T3-, T3a- und T2-Instances handelt. Wenn Sie regelmäßig über Ihre Basisleistung hinausgehen, stellen Sie sicher, dass Sie dies weiterhin auf der Grundlage der vCPUs des neuen Instance-Typs tun können. Weitere Informationen finden Sie unter CPU-Guthaben und Basisleistung für Burstable-Performance-Instances im Amazon EC2 EC2-Benutzerhandbuch.
- Wenn Sie eine Reserved Instance erworben haben, wird Ihnen Ihre On-Demand-Instance möglicherweise als Reserved Instance in Rechnung gestellt. Bevor Sie den aktuellen Instance-Typ ändern, sollten Sie zunächst die Auswirkungen auf die Nutzung und Abdeckung der Reserved Instance bewerten.
- Ziehen Sie nach Möglichkeit einen Umstieg auf Instances der neueren Generation in Betracht.
- Bei der Migration auf eine andere Instance-Familie ist darauf zu achten, dass der aktuelle Instance-Typ und der neue Instance-Typ miteinander kompatibel sind, z. B. in Bezug auf Virtualisierung, Architektur oder Netzwerktyp. Weitere Informationen finden Sie unter Kompatibilität bei der Größenänderung von Instances im Amazon EC2 EC2-Benutzerhandbuch.
- Berücksichtigen Sie abschließend die Bewertung des Leistungsrisikos, die für jede Empfehlung angegeben ist. Das Leistungsrisiko gibt den Aufwand an, den Sie möglicherweise aufwenden müssen, um zu überprüfen, ob der empfohlene Instance-Typ den Leistungsanforderungen Ihrem Workload entspricht. Darüber hinaus empfehlen wir, vor und nach jeder Änderung Last- und Leistungstests durchzuführen.

Weitere Ressourcen

Weitere Informationen zu den Themen auf dieser Seite finden Sie in den folgenden Quellen:

- Amazon-EC2-Instance-Typen
- AWS Compute Optimizer Benutzerhandbuch

Um den Datenverkehr über die Instances in Ihrer Auto-Scaling-Gruppe zu verteilen, verwenden Sie Elastic-Load-Balancing.

Mit Elastic Load Balancer können Sie den eingehenden Datenverkehr der Anwendung automatisch auf sämtliche EC2-Instances verteilen, die Sie ausführen. Sie können mit Elastic Load Balancing eingehende Anforderungen verwalten, indem Sie den Datenverkehr optimal weiterleiten, sodass keine Instance überfordert ist.

Um Elastic Load Balancing mit Ihrer Auto-Scaling-Gruppe zu verwenden, fügen Sie den Load Balancer an Ihre Auto-Scaling-Gruppe an. Damit wird die Auto-Scaling-Gruppe beim Load Balancer registriert, der als einziger Kontaktpunkt für den gesamten eingehenden Datenverkehr zu den Instances in Ihrer Auto-Scaling-Gruppe fungiert.

Wenn Sie Ihren Elastic Load Balancer mit einer Auto-Scaling-Gruppe verwenden, ist es nicht erforderlich, die EC2-Instances beim Load Balancer oder der Zielgruppe anzumelden. Instances, die von Ihrer Auto-Scaling-Gruppe gestartet werden, werden automatisch beim Load Balancer registriert. Ebenso werden Instances, die durch Ihre Auto-Scaling-Gruppe beendet werden, automatisch vom Load Balancer abgemeldet.

Nachdem Sie einen Load Balancer an Ihre Auto-Scaling-Gruppe angefügt haben, können Sie Ihre Auto-Scaling-Gruppe so konfigurieren, dass Elastic Load Balancing-Metriken (wie die Application Load Balancer-Anforderungsanzahl pro Ziel) verwendet werden, um die Anzahl der Instances in der Gruppe zu skalieren, wenn der Bedarf schwankt.

Optional können Sie Elastic Load Balancing-Zustandsprüfungen zu Ihrer Auto-Scaling-Gruppe hinzufügen, damit Amazon EC2 Auto Scaling auf der Grundlage dieser zusätzlichen Zustandsprüfungen fehlerhafte Instances identifizieren und ersetzen kann. Andernfalls können Sie einen CloudWatch Alarm einrichten, der Sie benachrichtigt, wenn die Anzahl gesunder Hosts der Zielgruppe niedriger als zulässig ist.

Inhalt

- Überprüfen des Anhangsstatus Ihres Load Balancers
- Hinzufügen oder Entfernen von Availability Zones
- Beispiele für die Arbeit mit Elastic Load Balancing mit dem AWS Command Line Interface

Arten von Elastic Load Balancing

Elastic Load Balancing bietet vier Arten von Load Balancern, die mit Ihrer Auto-Scaling-Gruppe verwendet werden können: Application Load Balancer, Network Load Balancer, Gateway Load Balancer und Classic Load Balancer.

Es gibt einen entscheidenden Unterschied in der Konfiguration der Load Balancer-Typen. Bei Application Load Balancern, Network Load Balancern und Gateway Load Balancern werden Instances als Ziele bei einer Zielgruppe registriert, und Sie leiten den Datenverkehr an die Zielgruppe weiter. Bei Classic Load Balancern werden Instances direkt beim Load Balancer registriert.

Application Load Balancer

Führt das Routing und den Lastenausgleich auf Anwendungsebene (HTTP/HTTPS) durch und unterstützt das pfadbasierte Routing. Ein Application Load Balancer kann Anfragen an ein oder mehrere registrierte Ziele weiterleiten, z. B. EC2-Instances in Ihre Virtual Private Cloud (VPC).

Network Load Balancer

Führt das Routing und den Lastenausgleich auf Transportebene (TCP/UDP-Layer-4) basierend auf extrahierten Adressinformationen aus dem Layer-4-Header durch. Network Load Balancers können Datenverkehrsspitzen verarbeiten, die Quell-IP-Adresse des Clients beibehalten und eine feste IP für die Nutzungsdauer des Load Balancers verwenden.

Gateway Load Balancer

Verteilt den Datenverkehr an eine Flotte von Appliance-Instances. Bietet Skalierung, Verfügbarkeit und Einfachheit für virtuelle Appliances von Drittanbietern, wie Firewalls, Eindringungserkennungs- und -präventionssysteme und andere Appliances. Gateway Load Balancer arbeiten mit virtuellen Appliances, die das GENEVE-Protokoll unterstützen. Zusätzliche technische Integration ist erforderlich. Bitte konsultieren Sie daher das Benutzerhandbuch, bevor Sie einen Gateway Load Balancer auswählen.

Classic Load Balancer

Führt das Routing und den Lastenausgleich auf Transportebene (TCP/SSL) oder Anwendungsebene (HTTP/HTTPS) durch.

Weitere Informationen zu den verschiedenen verfügbaren Load Balancer-Typen finden Sie in den folgenden Ressourcen:

- Was ist Elastic Load Balancing?
- Was ist ein Application Load Balancer?
- Was ist ein Network Load Balancer?
- Was ist ein Gateway Load Balancer?
- Was ist ein Classic Load Balancer?

Bereiten Sie sich darauf vor, Ihrer Auto Scaling Scaling-Gruppe einen Elastic Load Balancing-Load Balancer hinzuzufügen

Bevor Sie Ihrer Auto Scaling Scaling-Gruppe einen Elastic Load Balancing Load Balancer hinzufügen, müssen Sie die folgenden Voraussetzungen erfüllen:

- Sie müssen bereits den Load Balancer und die Zielgruppe erstellt haben, die für die Weiterleitung des Datenverkehrs an Ihre Auto Scaling Scaling-Gruppe verwendet werden.

  Es gibt zwei Möglichkeiten, den Load Balancer und die Zielgruppe zu erstellen:

  - Elastic Load Balancing verwenden — Folgen Sie den Verfahren in der Elastic Load Balancing Balancing-Dokumentation, um den Load Balancer und die Zielgruppe zu erstellen und zu konfigurieren, bevor Sie die Auto Scaling Scaling-Gruppe erstellen. Den Schritt zur Registrierung Ihrer Amazon EC2-Instances überspringen. Amazon EC2 Auto Scaling kümmert sich automatisch um die Registrierung (und Deregistrierung) von Instances, wenn Sie Ihrer Auto Scaling Scaling-Gruppe eine Zielgruppe zuordnen. Weitere Informationen finden Sie unter Erste Schritte mit Elastic Load Balancing im Elastic Load Balancing-Benutzerhandbuch.
  - Verwenden von Amazon EC2 Auto Scaling — Erstellen, konfigurieren und verknüpfen Sie den Load Balancer und die Zielgruppe mit einer Basiskonfiguration von der Amazon EC2 Auto Scaling Scaling-Konsole aus. Weitere Informationen finden Sie unter Konfigurieren eines Application Load Balancer oder Network Load Balancer aus der Amazon EC2 Auto Scaling-Konsole.
- Bevor Sie einen Load Balancer erstellen, sollten Sie wissen, welche Art von Load Balancer Sie benötigen. Weitere Informationen finden Sie unter Arten von Elastic Load Balancing.
- Der Load Balancer und seine Zielgruppe müssen sich in derselben AWS-Konto VPC und Region wie Ihre Auto Scaling Scaling-Gruppe befinden.

- Die Zielgruppe muss den Zieltyp `instance` aufweisen. Sie können keinen Zieltyp von `ip` angeben, wenn Sie eine Auto-Scaling-Gruppe verwenden.
- Wenn die Startvorlage für Ihre Auto Scaling Scaling-Gruppe nicht die richtige Sicherheitsgruppe enthält, um den erforderlichen eingehenden Datenverkehr vom Load Balancer zuzulassen, müssen Sie die Startvorlage aktualisieren. Die empfohlenen Regeln hängen vom Typ des Load Balancers und den Arten von Backends ab, die der Load Balancer verwendet. Um beispielsweise Datenverkehr an Webserver weiterzuleiten, lassen Sie den eingehenden HTTP-Zugriff auf Port 80 vom Load Balancer zu. Bestehende Instances werden nicht mit den neuen Einstellungen aktualisiert, wenn die Startvorlage geändert wird. Um bestehende Instances zu aktualisieren, können Sie eine Instance-Aktualisierung starten, um die Instances zu ersetzen. Weitere Informationen finden Sie unter Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren.
- Die Sicherheitsgruppen in der Startvorlage müssen außerdem den Zugriff vom Load Balancer auf den richtigen Port zulassen, damit Elastic Load Balancing seine Integritätsprüfungen durchführen kann.
- Bei der Bereitstellung virtueller Appliances hinter einem Gateway Load Balancer muss das Amazon Machine Image (AMI) in der Startvorlage die ID eines AMI angeben, das das GENEVE-Protokoll unterstützt, damit die Auto Scaling Scaling-Gruppe Datenverkehr mit einem Gateway Load Balancer austauschen kann. Außerdem müssen die Sicherheitsgruppen in der Startvorlage UDP-Verkehr auf Port 6081 zulassen.

> **Tip**
>
> Wenn Sie Bootstrapping-Skripte haben, deren Fertigstellung eine Weile dauert, können Sie Ihrer Auto-Scaling-Gruppe optional einen Start-Lebenszyklus-Hook hinzufügen, um die Registrierung von Instances hinter dem Load Balancer zu verzögern, bevor Ihre Bootstrap-Skripte erfolgreich abgeschlossen wurden und die Anwendungen auf den Instances bereit sind, Datenverkehr zu akzeptieren. Sie können keinen Lebenszyklus-Hook hinzufügen, wenn Sie zum ersten Mal eine Auto-Scaling-Gruppe in der Amazon-EC2-Auto-Scaling-Konsole erstellen. Sie können jedoch einen Lifecycle-Hook hinzufügen, nachdem die Gruppe erstellt wurde. Weitere Informationen finden Sie unter Lebenszyklus-Hooks bei Amazon EC2 Auto Scaling.

Konfigurieren Sie Integritätsprüfungen für Ziele

Sie können Integritätsprüfungen für Ihre Ziele konfigurieren, die bei einem Elastic Load Balancing Load Balancer registriert sind, um sicherzustellen, dass sie den Datenverkehr ordnungsgemäß verarbeiten können. Die spezifischen Schritte variieren je nach Art des Load Balancers, den Sie verwenden. Weitere Informationen finden Sie in den folgenden Ressourcen:

- Application Load Balancer — Informationen zu den Zustandsprüfungen für Ihre Zielgruppen finden Sie im Benutzerhandbuch für Application Load Balancer.
- Network Load Balancer — Weitere Informationen finden Sie im Benutzerhandbuch für Network Load Balancer unter Gesundheitschecks für Ihre Zielgruppen.
- Gateway Load Balancer — Weitere Informationen finden Sie im Benutzerhandbuch für Gateway Load Balancer unter Gesundheitschecks für Ihre Zielgruppen.
- Classic Load Balancer — Weitere Informationen finden Sie unter Konfigurieren von Zustandsprüfungen für Ihren Classic Load Balancer im Benutzerhandbuch für Classic Load Balancer.

Standardmäßig betrachtet Amazon EC2 Auto Scaling eine Instance nicht als fehlerhaft und ersetzt sie, wenn sie die Elastic Load Balancing Balancing-Zustandsprüfungen nicht besteht. Die Standard-Zustandsprüfungen für eine Auto-Scaling-Gruppe sind ausschließlich EC2-Zustandsprüfungen. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Damit Amazon EC2 Auto Scaling Instances ersetzen kann, die von Elastic Load Balancing als fehlerhaft gemeldet wurden, können Sie Ihre Auto Scaling Scaling-Gruppe so konfigurieren, dass sie Elastic Load Balancing Health Checks verwendet. Auf diese Weise betrachtet Amazon EC2 Auto Scaling die Instance als fehlerhaft, wenn sie entweder die EC2-Zustandsprüfungen oder die Elastic Load Balancing Balancing-Zustandsprüfungen nicht besteht. Wurden einer Gruppe mehrere Load Balancer-Zielgruppen oder Classic Load Balancer hinzugefügt, müssen alle melden, dass die Instance fehlerfrei ist, damit die Instance als fehlerfrei eingestuft wird. Wenn eine davon eine Instance als fehlerhaft meldet, ersetzt die Auto-Scaling-Gruppe die Instance, selbst dann, wenn sie von anderen als fehlerfrei gemeldet wird.

Informationen darüber, wie Sie diese Integritätsprüfungen für Ihre Auto Scaling Scaling-Gruppe aktivieren, finden Sie unterFügen Sie Ihrer Auto Scaling Scaling-Gruppe einen Elastic Load Balancing Load Balancer hinzu.

> **Note**
>
> Um sicherzustellen, dass diese Zustandsprüfungen so schnell wie möglich beginnen, stellen Sie sicher, dass die Kulanzfrist für die Integritätsprüfung Ihrer Gruppe nicht zu hoch, sondern hoch genug ist, damit Ihre Elastic Load Balancing Balancing-Zustandsprüfungen feststellen können, ob ein Ziel für die Bearbeitung von Anfragen verfügbar ist. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.

Fügen Sie Ihrer Auto Scaling Scaling-Gruppe einen Elastic Load Balancing Load Balancer hinzu

In diesem Thema wird beschrieben, wie Sie einen Elastic Load Balancing Load Balancer an eine Auto Scaling Scaling-Gruppe anhängen. Außerdem wird beschrieben, wie Elastic Load Balancing Health Checks aktiviert werden, damit Amazon EC2 Auto Scaling Instances ersetzen kann, die Elastic Load Balancing als fehlerhaft meldet.

Standardmäßig ersetzt Amazon EC2 Auto Scaling nur Instances, die aufgrund von Amazon EC2-Zustandsprüfungen nicht zustandsbehaftet oder nicht erreichbar sind. Wenn Sie Elastic Load Balancing Health Checks aktivieren, kann Amazon EC2 Auto Scaling eine laufende Instance ersetzen, falls einer der Elastic Load Balancing Load Balancer, die Sie der Auto Scaling Scaling-Gruppe zuordnen, sie als fehlerhaft meldet.

Ein Tutorial zum Hinzufügen eines Application Load Balancer zu Ihrer Auto Scaling Scaling-Gruppe finden Sie unter. Tutorial: Einrichten einer skalierten Anwendung mit Load Balancing

> **Important**
>
> Bevor Sie fortfahren, müssen Sie alle im vorherigen Abschnitt genannten Voraussetzungen erfüllen.

Inhalt

Fügen Sie eine Zielgruppe oder einen Classic Load Balancer hinzu

Wenn Sie eine Auto Scaling Scaling-Gruppe erstellen oder aktualisieren, können Sie eine oder mehrere Zielgruppen oder Classic Load Balancer anhängen. Wenn Sie einen Application Load Balancer, Network Load Balancer oder Gateway Load Balancer anhängen, fügen Sie eine Zielgruppe und nicht den Load Balancer selbst hinzu.

Befolgen Sie die Schritte in diesem Abschnitt, um die Konsole für Folgendes zu verwenden:

- Einer Auto Scaling Scaling-Gruppe eine Zielgruppe oder einen Classic Load Balancer zuordnen
- Schalten Sie die Integritätsprüfungen für Elastic Load Balancing ein

So fügen Sie beim Erstellen einer neuen Auto-Scaling-Gruppe einen vorhandenen Load Balancer hinzu

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, in der AWS-Region Sie Ihren Load Balancer erstellt haben.
3. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
4. Wählen Sie in den Schritten 1 und 2 die gewünschten Optionen aus und fahren Sie mit Schritt 3: Konfigurieren von erweiterten Optionen fort.
5. Für Load Balancing, wählen Sie An einen vorhandenen Load Balancer anhängen aus.
6. Bei An einen vorhandenen Load Balancer anhängenführen Sie im einen der folgenden Schritte aus:

   a. Gehen Sie für Application Load Balancers, Network Load Balancers und Gateway Load Balancers folgendermaßen vor:

      Klicken Sie auf Aus Ihren Zielgruppen für Load Balancer auswählen und wählen Sie dann im Feld Vorhandene Zielgruppen für Load Balancer eine Zielgruppe aus.

   b. Für Classic Load Balancer:

      Klicken Sie auf Auswählen aus Classic Load Balancersund wählen Sie dann Ihren Load Balancer im Feld Classic Load Balancer aus.
7. (Optional) Wählen Sie für Zustandsprüfungen und Zusätzliche Zustandsprüfungstypen die Option Elastic Load Balancing-Zustandsprüfungen aktivieren aus.

8. (Optional) Geben Sie unter Karenzzeit für die Zustandsprüfung die Zeit in Sekunden ein. So lange muss Amazon EC2 Auto Scaling warten, bevor der Zustand einer Instance überprüft wird nachdem Sie den Zustand `InService` erreicht hat. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.
9. Fahren Sie mit dem Erstellen der Auto-Scaling-Gruppe fort. Ihre Instances werden automatisch beim Load Balancer registriert, nachdem die Auto-Scaling-Gruppe erstellt wurde.

So fügen Sie einen vorhandenen Load Balancer an Ihre Auto-Scaling-Gruppe nach ihrer Erstellung an

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option Load Balancing, Bearbeiten.
4. Führen Sie unter Load balancing (Lastenausgleich) eine der folgenden Aktionen aus:

   a. Für Zielgruppen für Application, Network oder Gateway Load Balancer wählen Sie das entsprechende Kontrollkästchen und wählen eine Zielgruppe aus.

   b. Für Classic Load Balancer wählen Sie das entsprechende Kontrollkästchen Ihren Load Balancer aus.
5. Wählen Sie Aktualisieren.

Wenn Sie mit dem Anhängen des Load Balancers fertig sind, können Sie optional die Integritätsprüfungen aktivieren, die ihn verwenden.

So aktivieren Sie die Elastic Load Balancing Health Checks

1. Wählen Sie auf der Registerkarte Details die Option Zustandsprüfungen, Bearbeiten aus.
2. Wählen Sie für Zustandsprüfungen und Zusätzliche Zustandsprüfungstypen die Option Elastic Load Balancing-Zustandsprüfungen aktivieren aus.
3. Geben Sie unter Frist für Zustandsprüfungen die Zeit in Sekunden ein. So lange muss Amazon EC2 Auto Scaling warten, bevor der Zustand einer Instance überprüft wird nachdem Sie

den Zustand `InService` erreicht hat. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.

4. Wählen Sie Aktualisieren.

> **Note**
>
> Sie können den Status des Load Balancers überwachen, während er angefügt wird, indem Sie die AWS CLI verwenden. Wenn Amazon EC2 Auto Scaling die Instances erfolgreich registriert hat und mindestens eine registrierte Instance die Zustandsprüfungen besteht, erhalten Sie den Status von `InService`. Weitere Informationen finden Sie unter Überprüfen des Anhangsstatus Ihres Load Balancers.

Eine Zielgruppe oder einen Classic Load Balancer abtrennen

Wird der Load Balancer nicht mehr benötigt, führen Sie die folgenden Schritte aus, um ihn von der Auto-Scaling-Gruppe zu trennen.

Trennen Sie einen Load Balancer wie folgt von einer Gruppe:

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben einer vorhandenen Gruppe.

   Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option Load Balancing, Bearbeiten.
4. Führen Sie unter Load balancing (Lastenausgleich) eine der folgenden Aktionen aus:

   a. Wählen Sie für Application, Network oder Gateway Load Balancer-Zielgruppe das Löschsymbol (X) neben der Zielgruppe aus.

   b. Wählen Sie unter Classic Load Balancer das Löschsymbol (X) neben dem Load Balancer aus.
5. Wählen Sie Aktualisieren.

Wenn Sie mit dem Trennen der Zielgruppe fertig sind, können Sie die Elastic Load Balancing Health Checks deaktivieren.

So deaktivieren Sie die Integritätsprüfungen von Elastic Load Balancing

1. Wählen Sie auf der Registerkarte Details die Option Zustandsprüfungen, Bearbeiten aus.
2. Deaktivieren Sie für Integritätsprüfungen und Zusätzliche Zustandsprüfungstypen die Option Elastic Load Balancing Balancing-Zustandsprüfungen aktivieren.
3. Wählen Sie Aktualisieren.

Konfigurieren eines Application Load Balancer oder Network Load Balancer aus der Amazon EC2 Auto Scaling-Konsole

Gehen Sie wie folgt vor, um beim Erstellen der Auto-Scaling-Gruppe einen Application Load Balancer oder einen Network Load Balancer zu erstellen und anzufügen.

Beim Erstellen einer neuen Auto-Scaling-Gruppe einen vorhandenen Load Balancer erstellen oder hinzufügen

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
3. Wählen Sie in den Schritten 1 und 2 die gewünschten Optionen aus und fahren Sie mit Schritt 3: Konfigurieren von erweiterten Optionen fort.
4. Für Load Balancing wählen Sie An einen neuen Load Balancer anfügen aus.
    a. Bei An einen neuen Load Balancer anfügen wählen Sie für Load Balancer-Typ aus, ob ein Application Load Balancer oder Network Load Balancer erstellt werden soll.
    b. Für Load Balancer-Name geben Sie einen Namen für den Load Balancer ein, oder behalten Sie den Standardnamen bei.
    c. Für Load Balancer-Plan wählen Sie aus, ob ein öffentlicher mit dem Internet verbundener Load Balancer erstellt ewerden oder die Standardeinstellung für einen internen Load Balancer beibehalten werden soll.
    d. Für Availability Zones und Subnetze wählen Sie für jede Availability Zone, in der Sie Ihre EC2-Instances starten möchten, das öffentliche Subnetz aus. (Diese werden ab Schritt 2 ausgefüllt.)

e. Für Listener und Routing aktualisieren Sie die Portnummer für Ihren Listener (falls erforderlich) und unter Standard-Routing wählen SieErstellen einer Zielgruppeaus. Alternativ können Sie eine vorhandene Zielgruppe aus der Dropdown-Liste auswählen.

f. Wenn Sie im letzten Schritt Erstellen einer Zielgruppe ausgewählt haben, geben Sie für Name der neuen Zielgruppe einen Namen für die Zielgruppe ein, oder behalten Sie den Standardnamen bei.

g. Um Tags zu Ihrem Load Balancer hinzuzufügen, wählen Sie Add tag (Tag hinzufügen)und geben Sie einen Tag-Schlüssel und den Wert für jedes Tag an.

5. (Optional) Wählen Sie für Zustandsprüfungen und Zusätzliche Zustandsprüfungstypen die Option Elastic Load Balancing-Zustandsprüfungen aktivieren aus.
6. (Optional) Geben Sie unter Karenzzeit für die Zustandsprüfung die Zeit in Sekunden ein. So lange muss Amazon EC2 Auto Scaling warten, bevor der Zustand einer Instance überprüft wird nachdem Sie den Zustand `InService` erreicht hat. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.
7. Fahren Sie mit dem Erstellen der Auto-Scaling-Gruppe fort. Ihre Instances werden automatisch beim Load Balancer registriert, nachdem die Auto-Scaling-Gruppe erstellt wurde.

> **Note**
>
> Nachdem Sie die Auto-Scaling-Gruppe erstellt haben, können Sie mit der Elastic Load Balancing-Konsole zusätzliche Listener erstellen. Dies ist nützlich, wenn Sie einen Listener mit einem sicheren Protokoll wie HTTPS oder einem UDP-Listener erstellen müssen. Sie können vorhandenen Load Balancern weitere Listener hinzufügen, sofern Sie unterschiedliche Ports verwenden.

Überprüfen des Anhangsstatus Ihres Load Balancers

Nachdem Sie einen Load Balancer hinzufügen, wird er in den Status `Adding` versetzt, solange er die Instances der Gruppe registriert. Wenn alle Instances der Gruppe angemeldet sind, wird sie in den Status `Added` versetzt. Besteht zumindest eine angemeldete Instance die Zustandsprüfungen, wird er in den Status `InService` versetzt. Wurde der Load Balancer in den Status `InService` versetzt, kann Amazon EC2 Auto Scaling alle Instances beenden und ersetzen, die als fehlerhaft gemeldet werden. Besteht keine angemeldete Instance die Zustandsprüfungen (beispielsweise aufgrund einer falsch konfigurierten Zustandsprüfung), wird der Load Balancer nicht in den Status `InService` versetzt. Amazon EC2 Auto Scaling beendet und ersetzt die Instances nicht.

Wenn Sie einen Load Balancer trennen, wird er in den Status `Removing` versetzt, solange er die Instances der Gruppe abmeldet. Nach der Abmeldung werden die Instances weiterhin ausgeführt. Standardmäßig ist Connection Draining für Application Load Balancer, Network Load Balancer und Gateway Load Balancer aktiviert. Ist Connection Draining aktiviert, wartet Elastic Load Balancing darauf, dass aktive Anforderungen abgeschlossen werden oder das maximale Zeitlimit abgelaufen ist (je nachdem, was zuerst eintritt), bevor die Instances abgemeldet werden.

Sie können den Status des Anhangs mit den SDKs AWS Command Line Interface (AWS CLI) oder den AWS SDKs überprüfen. Sie können den Status des Anhangs nicht von der Konsole aus überprüfen.

Um den Status des Anhangs AWS CLI zu überprüfen

Der folgende Befehl describe-traffic-sources gibt den Anhangsstatus aller Traffic-Quellen für die angegebene Auto-Scaling-Gruppe zurück.

```
aws autoscaling describe-traffic-sources --auto-scaling-group-name my-asg
```

Das Beispiel gibt den ARN von Elastic-Load-Balancing zurück, das an die Auto-Scaling-Gruppe angehängt ist, zusammen mit dem Anhangsstatus der Zielgruppe im Element `State`.

```
{
    "TrafficSources": [
        {
            "Identifier": "arn:aws:elasticloadbalancing:region:account-
id:targetgroup/my-targets/1234567890123456",
            "State": "InService",
            "Type": "elbv2"
        }
    ]
}
```

Hinzufügen oder Entfernen von Availability Zones

Sie können die Vorteile der Sicherheit und Zuverlässigkeit geografischer Redundanz nutzen, indem Sie Ihre Auto-Scaling-Gruppe auf mehrere Availability Zones der Region, in der Sie arbeiten, verteilen und dann einen Load Balancer zum Verteilen des eingehenden Datenverkehrs auf diese Availability Zones hinzufügen.

Wenn eine Availability Zone fehlerhaft oder nicht verfügbar ist, startet Amazon EC2 Auto Scaling neue Instances in einer nicht betroffenen Availability Zone. Wenn die fehlerhafte Availability Zone meldet, dass sie wieder fehlerfrei ist, verteilt Amazon EC2 Auto Scaling die Anwendungs-Instances automatisch gleichmäßig auf alle Availability Zones für Ihre Auto-Scaling-Gruppe. Amazon EC2 Auto Scaling versucht dazu, in der Availability Zone mit den wenigsten Instances neue Instances zu starten. Scheitert dies jedoch, setzt Amazon EC2 Auto Scaling den Versuch in anderen Availability Zones fort, bis er erfolgreich ist.

Elastic Load Balancing erstellt einen Load Balancer-Knoten für jede Availability Zone, die Sie für den Load Balancer aktivieren. Wenn zonenübergreifendes Load Balancing für Ihren Load Balancer aktiviert ist, verteilt jeder Load Balancer-Knoten den Datenverkehr gleichmäßig auf die registrierten Ziele in allen aktivierten Availability Zones. Wenn zonenübergreifendes Load Balancing deaktiviert ist, verteilt jeder Load Balancer-Knoten Anfragen gleichmäßig nur auf die registrierten Instances in seiner aktivierten Availability Zone.

Sie müssen mindestens eine Availability Zone angeben, wenn Sie Ihre Auto-Scaling-Gruppe erstellen. Sie können die Verfügbarkeit der skalierten Anwendung mit Lastenausgleich erweitern, indem Sie der Auto-Scaling-Gruppe ein Availability Zone hinzufügen und dann dem Load Balancer Zugriff auf sie gewähren (sofern der Load Balancer dies unterstützt).

Inhalt

Fügen Sie eine Availability Zone hinzu

Gehen Sie wie folgt vor, um Ihre Auto-Scaling-Gruppe und Ihren Load Balancer auf ein Subnetz in einer zusätzlichen Availability Zone zu erweitern.

Hinzufügen einer Availability Zone

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben einer vorhandenen Gruppe.

Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet.

3. Wählen Sie auf der Registerkarte Details die Option Netzwerk, Bearbeiten.
4. In Subnetze wählen Sie das Subnetz aus, das der Availability Zone entspricht, das Sie der Auto-Scaling-Gruppe hinzufügen möchten.
5. Wählen Sie Aktualisieren.
6. Führen Sie die folgenden Schritte aus, um die Availability Zones für Ihren Load Balancer so zu aktualisieren, dass er dieselben Zonen wie Ihre Auto-Scaling-Gruppe verwendet:

   a. Wählen Sie im Navigationsbereich unter LOAD BALANCING die Option Load Balancers aus.

   b. Wählen Sie Ihren -Load Balancer.

   c. Führen Sie eine der folgenden Aktionen aus:

      - Für Application und Network Load Balancer:

        1. Wählen Sie auf der Registerkarte Description (Beschreibung) für Availability Zones Edit subnets (Subnetze bearbeiten) aus.
        2. Klicken Sie auf der Seite Subnetze bearbeiten für Availability Zones auf das Kontrollkästchen für die hinzuzufügende Availability Zone. Wenn es nur ein Subnetz für diese Zone gibt, ist es ausgewählt. Wenn es mehr als ein Subnetz für diese Zone gibt, wählen Sie eines der Subnetze aus.

      - Für Classic Load Balancer in einer VPC:

        1. Wählen Sie auf der Registerkarte Instances die Option Edit Availability Zones.
        2. Wählen Sie auf der Seite Add and Remove Subnets (Subnetze hinzufügen und entfernen) unter Available subnets (Verfügbare Subnetze) für das hinzuzufügende Subnetz das Symbol „Hinzufügen” (+) aus. Das Subnetz wird nach Selected subnets verschoben.

   d. Wählen Sie Speichern.

Entfernen einer Availability Zone

Gehen Sie wie folgt vor, um eine Availability Zone aus Ihrer Auto-Scaling-Gruppe und Ihrem Load Balancer zu entfernen.

Entfernen einer Availability Zone

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben einer vorhandenen Gruppe.

   Im unteren Teil der Seite Auto Scaling groups (Auto-Scaling-Gruppen) wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option Netzwerk, Bearbeiten.
4. In Subnetze wählen Sie das Lösch-Symbol (X) für das Subnetz aus, das der Availability Zone entspricht, das Sie aus der Auto-Scaling-Gruppe entfernen möchten. Wenn es mehr als ein Subnetz für diese Zone gibt, wählen Sie für jede Zone das Löschsymbol (X) aus.
5. Wählen Sie Aktualisieren.
6. Führen Sie die folgenden Schritte aus, um die Availability Zones für Ihren Load Balancer so zu aktualisieren, dass er dieselben Zonen wie Ihre Auto-Scaling-Gruppe verwendet:

   a. Wählen Sie im Navigationsbereich unter LOAD BALANCING die Option Load Balancers aus.

   b. Wählen Sie Ihren -Load Balancer.

   c. Führen Sie eine der folgenden Aktionen aus:

      - Für Application und Network Load Balancer:

        1. Wählen Sie auf der Registerkarte Description (Beschreibung) für Availability Zones Edit subnets (Subnetze bearbeiten) aus.
        2. Deaktivieren Sie auf der Seite Subnetze bearbeiten für Availability Zones das Kontrollkästchen, um das Subnetz für die Availability Zone zu entfernen.

      - Für Classic Load Balancer in einer VPC:

        1. Wählen Sie auf der Registerkarte Instances die Option Edit Availability Zones.
        2. Entfernen Sie auf der Seite Add and Remove Subnets (Subnetze hinzufügen und entfernen) unter Available subnets (Verfügbare Subnetze) das Subnetz über dessen Löschsymbol (-). Das Subnetz wird zu Available Subnets verschoben.

   d. Wählen Sie Speichern.

Zugehörige Ressourcen

Amazon EC2 Auto Scaling gleicht Ihre Gruppe aus, wenn Sie Availability Zones ändern. Das heißt, dass einige Instances ersetzt und neu verteilt werden müssen. Weitere Informationen finden Sie unter Beispiel: Aufteilen von Instances in mehrere Availability Zones.

Wenn Sie Ziele in Availability Zones registriert haben, die nicht für den Load Balancer aktiviert sind, leitet der Load Balancer keinen Traffic an diese Ziele weiter. Weitere Informationen finden Sie unter Funktionsweise von Elastic Load Balancing im Benutzerhandbuch für Elastic Load Balancing.

Einschränkungen

Zur Aktualisierung der Availability Zones, die für Ihren Load Balancer aktiviert sind, müssen Sie die folgenden Einschränkungen kennen:

- Wenn Sie eine Availability Zone für Ihren Load Balancer aktivieren, geben Sie ein Subnetz aus dieser Availability Zone an. Beachten Sie, dass Sie höchstens ein Subnetz pro Availability Zone für Ihren Load Balancer aktivieren können.
- Für Load Balancer mit Internetverbindung müssen die von Ihnen für den Load Balancer angegebenen Subnetze über mindestens acht verfügbare IP-Adressen verfügen.
- Für Application Load Balancers müssen Sie Subnetze aus mindestens zwei Availability Zones angeben.
- Bei Network Load Balancern können Sie die aktivierten Availability Zones nicht deaktivieren, aber Sie können zusätzliche Zones aktivieren.
- Für Gateway Load Balancer können Sie die aktivierten Availability Zones nicht deaktivieren, aber Sie können zusätzliche aktivieren.

Beispiele für die Arbeit mit Elastic Load Balancing mit dem AWS Command Line Interface

Verwenden Sie AWS Command Line Interface (AWS CLI), um Load Balancer und Zielgruppen anzuhängen, zu trennen und zu beschreiben, Elastic Load Balancing Health Checks hinzuzufügen und zu entfernen und zu ändern, welche Availability Zones aktiviert sind.

Dieses Thema zeigt Beispiele für AWS CLI Befehle, die allgemeine Aufgaben für Amazon EC2 Auto Scaling ausführen.

> **Important**
>
> Weitere Befehlsbeispiele finden Sie unter `aws elbv2` und `aws elb` in der AWS CLI - Befehlsreferenz.

Inhalt

Hängen Sie Ihre Zielgruppe oder Ihren Classic Load Balancer an.

Verwenden Sie den folgenden Befehl create-auto-scaling-group, um eine Auto Scaling -Gruppe zu erstellen und gleichzeitig eine Zielgruppe anzuhängen, indem Sie ihren Amazon-Ressourcennamen (ARN) angeben. Die Zielgruppe kann einem Application Load Balancer, einem Network Load Balancer oder einem Gateway Load Balancer zugeordnet werden.

Ersetzen Sie die Beispielwerte für `--auto-scaling-group-name`, `--vpc-zone-identifier`, `--min-size` und `--max-size`. Ersetzen Sie für die Option `--launch-template` *`my-launch-template`* und *`1`* durch den Namen und die Version einer Startvorlage für Ihre Auto-Scaling-Gruppe. Für die Option `--traffic-sources` ersetzen Sie den Beispiel-ARN durch den ARN einer Zielgruppe für einen Application Load Balancer, Network Load Balancer oder Gateway Load Balancer.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
  --min-size 1 --max-size 5 \
  --traffic-sources "Identifier=arn:aws:elasticloadbalancing:region:account-
id:targetgroup/my-targets/12345678EXAMPLE1"
```

Verwenden Sie den Befehl attach-traffic-sources, um zusätzliche Zielgruppen an eine Auto-Scaling-Gruppe anzuhängen, nachdem sie erstellt wurde.

Mit dem folgenden Befehl fügen Sie derselben Gruppe eine weitere Zielgruppe hinzu.

```
aws autoscaling attach-traffic-sources --auto-scaling-group-name my-asg \
  --traffic-sources "Identifier=arn:aws:elasticloadbalancing:region:account-
id:targetgroup/my-targets/12345678EXAMPLE2"
```

Um Ihrer Gruppe einen Classic Load Balancer hinzuzufügen, geben Sie alternativ die Optionen `--traffic-sources` und `--type` an, wenn Sie create-auto-scaling-group oder attach-traffic-sources verwenden, wie im folgenden Beispiel. Ersetzen Sie *my-classic-load-balancer* durch den Namen eines Classic Load Balancer. Geben Sie für die Option `--type` einen Wert von **elb** an.

```
--traffic-sources "Identifier=my-classic-load-balancer" --type elb
```

Beschreiben Sie Ihre Zielgruppen oder Classic Load Balancers.

Verwenden Sie den folgenden Befehl describe-traffic-sources, um die Load Balancer oder Zielgruppen zu beschreiben, die Ihrer Auto-Scaling-Gruppe zugeordnet sind. Ersetzen Sie *my-asg* durch den Namen Ihrer Gruppe.

```
aws autoscaling describe-traffic-sources --auto-scaling-group-name my-asg
```

Das Beispiel gibt den ARN der Elastic Load Balancing-Zielgruppen zurück, die Sie der Auto-Scaling-Gruppe hinzugefügt haben.

```
{
    "TrafficSources": [
        {
            "Identifier": "arn:aws:elasticloadbalancing:region:account-
id:targetgroup/my-targets/12345678EXAMPLE1",
            "State": "InService",
            "Type": "elbv2"
        },
        {
            "Identifier": "arn:aws:elasticloadbalancing:region:account-
id:targetgroup/my-targets/12345678EXAMPLE2",
            "State": "InService",
```

```
            "Type": "elbv2"
        }
    ]
}
```

Eine Erklärung des `State`-Felds in der Ausgabe finden Sie unterÜberprüfen des Anhangsstatus Ihres Load Balancers.

Hinzufügen von Elastic Load Balancing-Zustandsprüfungen

Um Elastic Load Balancing-Zustandsprüfungen zu den Zustandsprüfungen hinzuzufügen, die Ihre Auto-Scaling-Gruppe auf Instances durchführt, führen Sie den folgenden update-auto-scaling-group-Befehl aus und geben Sie **`ELB`** als Wert für die Option `--health-check-type` an. Ersetzen Sie *`my-asg`* durch den Namen Ihrer Gruppe.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --health-check-type "ELB"
```

Neue Instances benötigen oft Zeit für eine kurze Aufwärmphase, bevor sie eine Zustandsprüfung bestehen können. Wenn die Übergangszeit nicht ausreichend Aufwärmzeit bietet, scheinen die Instances möglicherweise nicht bereit zu sein, Traffic zu verarbeiten. Amazon EC2 Auto Scaling kann diese Instances als fehlerhaft betrachten und ersetzen.

Verwenden Sie zum Aktualisieren der Frist für die Zustandsprüfung die Option `--health-check-grace-period`, wenn Sie update-auto-scaling-group verwenden, wie im folgenden Beispiel. Ersetzen Sie *`300`* durch die Anzahl der Sekunden, um neue Instances in Betrieb zu halten, bevor sie beendet werden, falls sie sich als fehlerhaft erweisen.

```
--health-check-grace-period 300
```

Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Ändern Ihrer Availability Zones

Für das Ändern Ihrer Availability Zones gelten einige Einschränkungen, die Sie kennen sollten. Weitere Informationen finden Sie unter Einschränkungen.

So ändern Sie die Availability Zones für einen Application Load Balancer oder Network Load Balancer

1. Bevor Sie die Availability Zones des Load Balancers ändern, empfiehlt es sich, zunächst die Availability Zones der Auto-Scaling-Gruppe zu aktualisieren, um sicherzustellen, dass Ihre Instance-Typen in den angegebenen Zonen verfügbar sind.

   Um die Availability Zones für Ihre Auto-Scaling-Gruppe zu aktualisieren, verwenden Sie den folgenden update-auto-scaling-group-Befehl. Ersetzen Sie die Beispiel-Subnetz-IDs durch die IDs der Subnetze in den Availability Zones, um diese zu aktivieren. Die angegebenen Subnetze ersetzen die zuvor aktivierten Subnetze. Ersetzen Sie *my-asg* durch den Namen Ihrer Gruppe.

   ```
   aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
     --vpc-zone-identifier "subnet-41767929,subnet-cb663da2,subnet-8360a9e7"
   ```

2. Verwenden Sie den folgenden describe-auto-scaling-groups-Befehl, um zu prüfen, ob die Instances in den neuen Subnetzen gestartet wurden. Wenn die Instances gestartet wurden, wird eine Liste der Instances und deren Status angezeigt. Ersetzen Sie *my-asg* durch den Namen Ihrer Gruppe.

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
   ```

3. Verwenden Sie den folgenden Befehl set-subnets, um die Subnetze für Ihren Load Balancer anzugeben. Ersetzen Sie die Beispiel-Subnetz-IDs durch die IDs der Subnetze in den Availability Zones, um diese zu aktivieren. Sie können nur ein Subnetz pro Availability Zone angeben. Die angegebenen Subnetze ersetzen die zuvor aktivierten Subnetze. Ersetzen Sie *my-lb-arn* durch den ARN Ihres Load Balancers.

   ```
   aws elbv2 set-subnets --load-balancer-arn my-lb-arn \
     --subnets subnet-41767929 subnet-cb663da2 subnet-8360a9e7
   ```

Ändern der Availability Zones für einen Classic Load Balancer

1. Bevor Sie die Availability Zones des Load Balancers ändern, empfiehlt es sich, zunächst die Availability Zones der Auto-Scaling-Gruppe zu aktualisieren, um sicherzustellen, dass Ihre Instance-Typen in den angegebenen Zonen verfügbar sind.

   Um die Availability Zones für Ihre Auto-Scaling-Gruppe zu aktualisieren, verwenden Sie den folgenden update-auto-scaling-group-Befehl. Ersetzen Sie die Beispiel-Subnetz-IDs durch die

IDs der Subnetze in den Availability Zones, um diese zu aktivieren. Die angegebenen Subnetze ersetzen die zuvor aktivierten Subnetze. Ersetzen Sie *my-asg* durch den Namen Ihrer Gruppe.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --vpc-zone-identifier "subnet-41767929,subnet-cb663da2"
```

2. Verwenden Sie den folgenden describe-auto-scaling-groups-Befehl, um zu prüfen, ob die Instances in den neuen Subnetzen gestartet wurden. Wenn die Instances gestartet wurden, wird eine Liste der Instances und deren Status angezeigt. Ersetzen Sie *my-asg* durch den Namen Ihrer Gruppe.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

3. Verwenden Sie den folgenden attach-load-balancer-to-subnets-Befehl, um die neue Availability Zone für Ihren Classic Load Balancer zu aktivieren. Ersetzen Sie zum Aktivieren die Beispiel-Subnetz-ID durch die ID des Subnetzes für die Availability Zone. Ersetzen Sie *my-lb* durch den Namen Ihres Load Balancers.

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-lb \
  --subnets subnet-cb663da2
```

Um eine Availability Zone zu deaktivieren, führen Sie den folgenden detach-load-balancer-from-subnets-Befehl aus. Ersetzen Sie zum Deaktivieren die Beispiel-Subnetz-ID durch die ID des Subnetzes für die Availability Zone. Ersetzen Sie *my-lb* durch den Namen Ihres Load Balancers.

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-lb \
  --subnets subnet-8360a9e7
```

Trennen Sie Ihre Zielgruppe oder Ihren Classic Load Balancer.

Der folgende detach-traffic-sources-Befehl trennt eine Zielgruppe von Ihrer Auto-Scaling-Gruppe, wenn Sie sie nicht mehr benötigen.

Ersetzen Sie für die Option `--auto-scaling-group-name` *my-asg* durch den Namen Ihrer Gruppe. Für die Option `--traffic-sources` ersetzen Sie den Beispiel-ARN durch den ARN einer Zielgruppe für einen Application Load Balancer, Network Load Balancer oder Gateway Load Balancer.

```
aws autoscaling detach-traffic-sources --auto-scaling-group-name my-asg \
  --traffic-sources "Identifier=arn:aws:elasticloadbalancing:region:account-
id:targetgroup/my-targets/1234567890123456"
```

Um einen Classic Load Balancer von Ihrer Gruppe zu trennen, geben Sie die Optionen `--traffic-sources` und `--type` an, wie im folgenden Beispiel. Ersetzen Sie *my-classic-load-balancer* durch den Namen eines Classic Load Balancer. Geben Sie für die Option `--type` einen Wert von **elb** an.

```
--traffic-sources "Identifier=my-classic-load-balancer" --type elb
```

Entfernen von Elastic Load Balancing-Zustandsprüfungen

Um Elastic Load Balancing-Zustandsprüfungen zu einer Auto-Scaling-Gruppe zu entfernen, führen Sie den folgenden update-auto-scaling-group-Befehl aus und geben Sie **EC2** als Wert für die Option `--health-check-type` an. Ersetzen Sie *my-asg* durch den Namen Ihrer Gruppe.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --health-check-type "EC2"
```

Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

Legacybefehle

Die folgenden Beispiele beschreiben, wie Sie Legacy-CLI-Befehle verwenden können, um Load Balancer und Zielgruppen anzufügen, zu trennen und zu beschreiben. Sie bleiben in diesem Dokument als Referenz für alle Kunden, die sie weiterhin verwenden möchten. Wir unterstützen weiterhin die alten CLI-Befehle, empfehlen jedoch, die neuen CLI-Befehle für „Trafficquellen“ zu verwenden, mit denen mehrere Datenverkehrsquellen-Typen angehängt und getrennt werden können. Sie können sowohl die Legacy-CLI-Befehle als auch die CLI-Befehle für „Trafficquellen“ in derselben Auto-Scaling-Gruppe verwenden.

Hängen Sie Ihre Zielgruppe oder Ihren Classic Load Balancer an (Legacy)

Fügen Sie Ihre Zielgruppe wie folgt hinzu:

Der folgende create-auto-scaling-group-Befehl erstellt eine Auto-Scaling-Gruppe mit angefügter Zielgruppe. Geben Sie den Amazon-Ressourcenname (ARN) einer Zielgruppe für einen Application Load Balancer, Network Load Balancer oder Gateway Load Balancer an.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
  --target-group-arns "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-targets/1234567890123456" \
  --min-size 1 --max-size 5
```

Der folgende attach-load-balancer-target-groups-Befehl fügt eine Zielgruppe an eine vorhandene Auto-Scaling-Gruppe an.

```
aws autoscaling attach-load-balancer-target-groups --auto-scaling-group-name my-asg \
  --target-group-arns "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-targets/1234567890123456"
```

Anhängen Ihres Classic Load Balancers

Der folgende create-auto-scaling-group-Befehl erstellt eine Auto-Scaling-Gruppe mit einem angefügten Classic Load Balancer.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-configuration-name my-launch-config \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
  --load-balancer-names "my-load-balancer" \
  --min-size 1 --max-size 5
```

Der folgende attach-load-balancers-Befehl fügt den angegebenen Classic Load Balancer an eine vorhandene Auto-Scaling-Gruppe an.

```
aws autoscaling attach-load-balancers --auto-scaling-group-name my-asg \
  --load-balancer-names my-lb
```

Beschreiben Sie Ihre Zielgruppe oder Ihren Classic Load Balancer (Legacy)

Beschreiben von Zielgruppen

Um die Zielgruppen zu beschreiben, die einer Auto-Scaling-Gruppe zugeordnet sind, verwenden Sie den describe-load-balancer-target-groups-Befehl. Im folgenden Beispiel werden die Zielgruppen für *my-asg* aufgelistet.

```
aws autoscaling describe-load-balancer-target-groups --auto-scaling-group-name my-asg
```

Beschreiben eines Classic Load Balancers

Um die Classic Load Balancers zu beschreiben, die mit einer Auto-Scaling-Gruppe verknüpft sind, verwenden Sie den describe-load-balancers-Befehl. Im folgenden Beispiel werden die Classic Load Balancers für*my-asg* aufgelistet.

```
aws autoscaling describe-load-balancers --auto-scaling-group-name my-asg
```

Trennen Sie Ihre Zielgruppe oder Ihren Classic Load Balancer (Legacy)

Trennen Sie eine Zielgruppe wie folgt:

Der folgende detach-load-balancer-target-groups-Befehl, trennt eine Zielgruppe von Ihrer Auto-Scaling-Gruppe, wenn Sie sie nicht mehr benötigen.

```
aws autoscaling detach-load-balancer-target-groups --auto-scaling-group-name my-asg \
  --target-group-arns "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-targets/1234567890123456"
```

Trennen eines Classic Load Balancers

Verwenden Sie den folgenden detach-load-balancers-Befehl, um einen Classic Load Balancer von der Auto-Scaling-Gruppe zu trennen, wenn Sie ihn nicht mehr benötigen.

```
aws autoscaling detach-load-balancers --auto-scaling-group-name my-asg \
  --load-balancer-names my-lb
```

Weiterleitung des Datenverkehrs zu Ihrer Auto-Scaling-Gruppe mit einer VPC Lattice-Zielgruppe

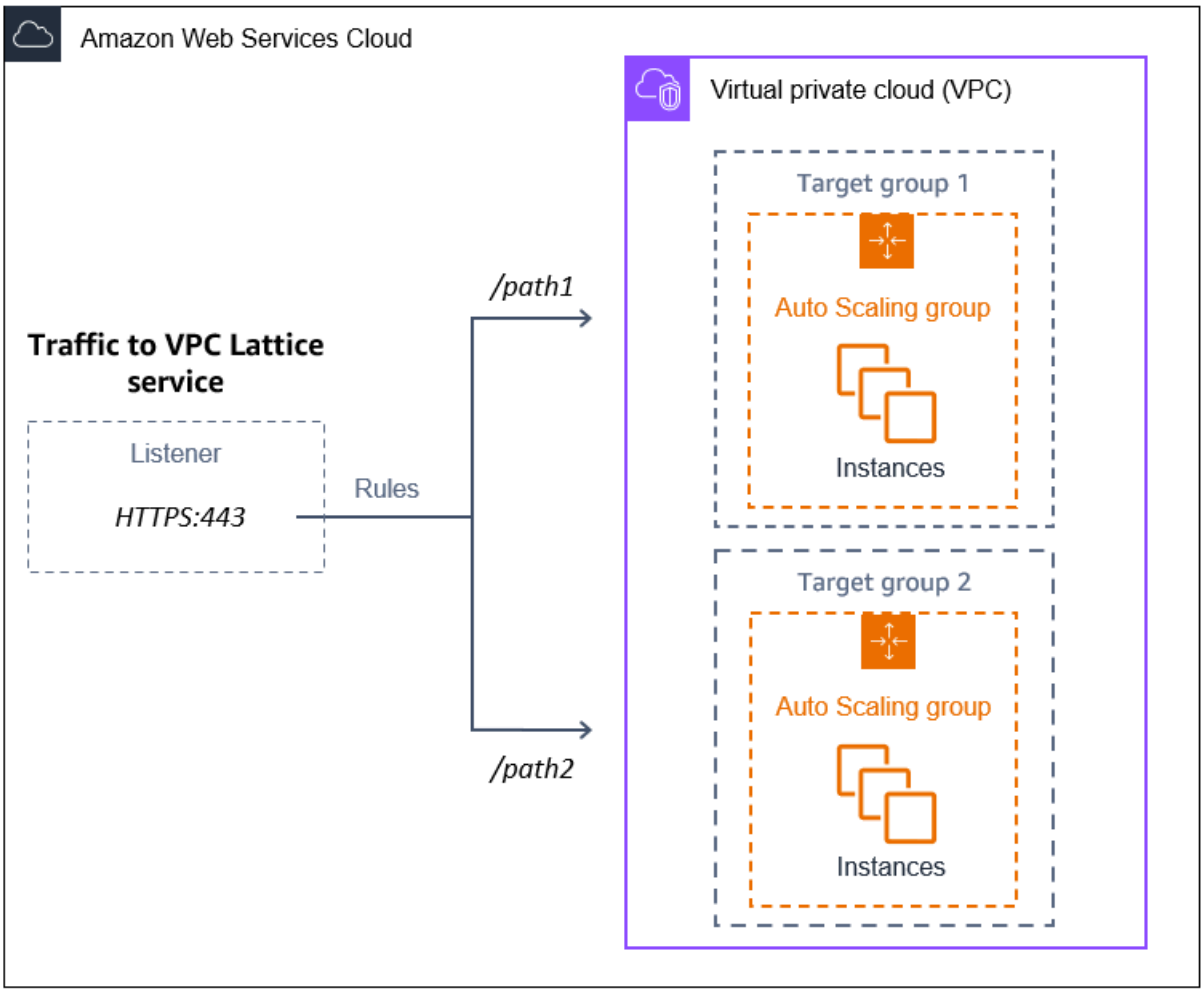
Sie können Amazon VPC Lattice verwenden, um den Datenverkehr und die API-Aufrufe zwischen Ihren Anwendungen und Services zu verwalten, die auf separaten Ressourcen laufen, wie z. B. Auto Scaling-Gruppen oder Lambda-Funktionen. VPC Lattice ist ein Anwendungsnetzwerk-Service, mit dem Sie alle Ihre Services über mehrere Konten und Virtual Private Clouds (VPC) hinweg verbinden, sichern und überwachen können. Weitere Informationen zu VPC Lattice finden Sie unter Was ist VPC Lattice?

Um mit VPC Lattice zu beginnen, erstellen Sie zunächst die erforderlichen VPC Lattice-Ressourcen, die es den Ressourcen in einer mit einem Service-Netzwerk verbundenen VPC erlaubt, sich

miteinander zu verbinden. Zu diesen Ressourcen gehören die Services, Listener, Listener-Regeln und Zielgruppen.

Um eine Auto Scaling-Gruppe mit einem VPC Lattice-Dienst zu verknüpfen, erstellen Sie eine Zielgruppe für den Dienst, die Anfragen an die nach Instance-ID registrierten Instances weiterleitet, und fügen Sie dem Dienst einen Listener hinzu, der Anfragen an die Zielgruppe sendet. Verbinden Sie dann die Zielgruppe mit Ihrer Auto-Scaling-Gruppe. Amazon EC2 Auto Scaling registriert die EC2-Instances automatisch als Ziele bei der Zielgruppe. Wenn Amazon EC2 Auto Scaling später eine Instance beenden muss, wird die Instance vor der Beendigung automatisch von der Zielgruppe abgemeldet.

Nachdem Sie die Zielgruppe hinzugefügt haben, ist sie der Einstiegspunkt für alle eingehenden Anfragen an Ihre Auto-Scaling-Gruppe. Wie das Beispiel im folgenden Diagramm zeigt, können eingehende Anfragen dann mithilfe der für einen VPC Lattice-Dienst angegebenen Listener-Regeln an die entsprechende Zielgruppe weitergeleitet werden.



Wenn der Datenverkehr durch VPC Lattice zu Ihrer Auto-Scaling-Gruppe geleitet wird, gleicht VPC Lattice die Anfragen unter den Instances in der Gruppe durch Round-Robin-Load Balancing aus. VPC Lattice kann auch den Zustand seiner registrierten Instances überwachen und den Datenverkehr nur an gesunde Instances weiterleiten.

Um Ihre Instances für eingehende Anfragen verfügbar zu halten, können Sie Ihrer Auto-Scaling-Gruppe optional VPC Lattice-Zustandsprüfungen hinzufügen. Wenn eine der EC2-Instances ausfällt, startet Ihre Auto Scaling-Gruppe automatisch eine neue Instance, um sie zu ersetzen. Das Verhalten der Zustandsprüfungen von VPC Lattice ähnelt dem Verhalten der Zustandsprüfungen des Elastic Load Balancing. Die Standard-Zustandsprüfungen für eine Auto-Scaling-Gruppe sind ausschließlich EC2-Zustandsprüfungen.

Weitere Informationen zu VPC Lattice finden Sie im Blog unter Vereinfachen von Service-to-Service-Konnektivität, Sicherheit und Überwachung mit Amazon VPC Lattice — Jetzt allgemein verfügbar. AWS

Inhalt

Vorbereiten des Hinzufügens einer VPC-Lattice-Zielgruppe an Ihre Auto-Scaling-Gruppe

Bevor Sie eine VPC Lattice-Zielgruppe mit Ihrer Auto-Scaling-Gruppe verbinden, müssen Sie die folgenden Voraussetzungen erfüllen:

- Sie müssen bereits ein VPC-Lattice-Dienstnetzwerk, einen Dienst, einen Listener und eine Zielgruppe erstellt haben. Weitere Informationen finden Sie unter den folgenden Themen im VPC Lattice Benutzerhandbuch:
  - Servicenetzwerke
  - Services
  - Listener
  - Zielgruppen
- Die Zielgruppe muss sich in derselben AWS-Konto VPC und Region wie Ihre Auto Scaling Scaling-Gruppe befinden.

- Die Zielgruppe muss den Zieltyp `instance` aufweisen. Sie können keinen Zieltyp von `ip` angeben, wenn Sie eine Auto-Scaling-Gruppe verwenden.
- Sie benötigen ausreichende IAM-Berechtigungen, um die Zielgruppe an die Auto-Scaling-Gruppe anhängen zu können. Die folgende Beispielrichtlinie zeigt die mindestens erforderlichen Berechtigungen, die zum Anhängen und Trennen von Zielgruppen erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:AttachTrafficSources",
                "autoscaling:DetachTrafficSources",
                "autoscaling:DescribeTrafficSources",
                "vpc-lattice:RegisterTargets",
                "vpc-lattice:DeregisterTargets"
            ],
            "Resource": "*"
        }
    ]
}
```

- Wenn die Startvorlage für Ihre Auto-Scaling-Gruppe nicht die richtigen Einstellungen für VPC Lattice enthält, z. B. eine kompatible Sicherheitsgruppe, müssen Sie die Startvorlage aktualisieren. Bestehende Instances werden nicht mit den neuen Einstellungen aktualisiert, wenn die Startvorlage geändert wird. Um bestehende Instances zu aktualisieren, können Sie eine Instance-Aktualisierung starten, um die Instances zu ersetzen. Weitere Informationen finden Sie unter Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren.
- Bevor Sie die VPC Lattice-Integritätsprüfungen für Ihre Auto-Scaling-Gruppe aktivieren, können Sie eine anwendungsbasierte Zustandsprüfung konfigurieren, um sicherzustellen, dass Ihre Anwendung wie erwartet reagiert. Weitere Informationen finden Sie unter Zustandsprüfungen für Ihre Zielgruppen im VPC Lattice Benutzerhandbuch.

Sicherheitsgruppen: Eingehende und ausgehende Regeln

Sicherheitsgruppen fungieren als Firewall für zugehörige EC2-Instances und kontrollieren sowohl den eingehenden als auch den ausgehenden Datenverkehr auf Instance-Ebene.

> **Note**
>
> Die Netzwerkkonfiguration ist so komplex, dass wir Ihnen dringend empfehlen, eine neue Sicherheitsgruppe für die Verwendung mit VPC Lattice zu erstellen. Es macht es auch einfacher AWS Support , Ihnen zu helfen, wenn Sie sie kontaktieren müssen. Die folgenden Abschnitte basieren auf der Annahme, dass Sie dieser Empfehlung folgen.
> Weitere Informationen zum Erstellen von Sicherheitsgruppen für VPC Lattice, die Sie mit Ihrer Auto-Scaling-Gruppe verwenden können, finden Sie unter Steuern des Datenverkehrs mithilfe von Sicherheitsgruppen im VPC Lattice Benutzerhandbuch. Weitere Informationen zur Behebung von Problemen mit dem Datenfluss finden Sie im VPC Lattice Benutzerhandbuch.

Informationen zum Erstellen einer Sicherheitsgruppe finden Sie unter Erstellen einer Sicherheitsgruppe im Amazon EC2 EC2-Benutzerhandbuch. Verwenden Sie die folgende Tabelle, um zu bestimmen, welche Optionen Sie auswählen müssen.

Option	Wert
Name	Ein Name, den Sie sich leicht merken können.
Beschreibung	Eine Beschreibung, die Ihnen hilft, die Sicherheitsgruppe zu identifizieren.
VPC	Dieselbe VPC wie die Auto-Scaling-Gruppe

Regeln für eingehenden Datenverkehr

Wenn Sie eine Sicherheitsgruppe erstellen, verfügt sie über keine Regeln für den eingehenden Datenverkehr. Es ist kein eingehender Verkehr von Clients innerhalb eines VPC-Lattice-Servicenetzwerks zu Ihrer Instance erlaubt, bis Sie der Sicherheitsgruppe Regeln für eingehenden Verkehr hinzufügen.

Damit Clients innerhalb eines VPC Lattice-Dienstnetzwerks eine Verbindung zu Instances in Ihrer Auto-Scaling-Gruppe herstellen können, muss die Sicherheitsgruppe für Ihre Auto Scaling-Gruppe

korrekt eingerichtet sein. Geben Sie ihm in diesem Fall eine Regel für eingehenden Datenverkehr, um Datenverkehr über den Namen der AWS verwalteten Präfixliste für VPC Lattice statt über eine bestimmte IP-Adresse zuzulassen. Die VPC Lattice-Präfixliste ist ein Bereich von IP-Adressen, die von VPC Lattice in CIDR-Notation verwendet werden. Weitere Informationen finden Sie unter Arbeiten mit AWS verwalteten Präfixlisten im Amazon VPC-Benutzerhandbuch.

Informationen zum Hinzufügen von Regeln zu einer Sicherheitsgruppe finden Sie unter Hinzufügen von Regeln zu Ihrer Sicherheitsgruppe im Amazon VPC Benutzerhandbuch, und verwenden Sie die folgende Tabelle, um die auszuwählenden Optionen zu bestimmen.

Option	Wert	
HTTP-Regel	Typ: HTTP Quelle: com.amazonaws.*region*.vpc-lattice	
HTTPS-Regel	Typ: HTTPS Quelle: com.amazonaws.*region*.vpc-lattice	

Die Sicherheitsgruppe ist zustandsbehaftet: Sie lässt den Datenverkehr von Clients innerhalb des VPC Lattice Service-Netzwerks zu Instances in Ihrer Auto-Scaling-Gruppe zu und sendet dann die Antwort an den Client zurück, den sie zuvor verlassen hat.

Regeln für ausgehenden Datenverkehr

Standardmäßig enthält eine Sicherheitsgruppe eine ausgehende Regel, die den gesamten ausgehenden Datenverkehr zulässt. Sie können diese Standardregel optional entfernen und eine Regel für ausgehenden Datenverkehr hinzufügen, um bestimmten Sicherheitsanforderungen gerecht zu werden.

Einschränkungen

- Gruppen mit gemischten Instances werden nicht unterstützt. Wenn Sie versuchen, eine VPC Lattice-Zielgruppe mit einer Auto Scaling-Gruppe zu verbinden, die eine Richtlinie für gemischte Instances hat, erhalten Sie die Fehlermeldung Derzeit können Auto Scaling-Gruppen mit gemischten Instances nicht in einen VPC Lattice-Service integriert werden.. Das liegt daran, dass

der Load-Balancing-Algorithmus die Last gleichmäßig auf alle verfügbaren Ressourcen verteilt und davon ausgeht, dass die Instances ähnlich genug sind, um gleiche Lasten zu bewältigen.

Hinzufügen einer VPC-Lattice-Zielgruppe zu Ihrer Auto-Scaling-Gruppe

In diesem Thema wird beschrieben, wie Sie eine VPC Lattice-Zielgruppe an eine Auto-Scaling-Gruppe anhängen. Außerdem wird beschrieben, wie VPC Lattice-Zustandsprüfungen aktiviert werden, damit Amazon EC2 Auto Scaling Instances ersetzen kann, die VPC Lattice als fehlerhaft meldet.

Standardmäßig ersetzt Amazon EC2 Auto Scaling nur Instances, die aufgrund von Amazon EC2-Zustandsprüfungen nicht zustandsbehaftet oder nicht erreichbar sind. Wenn Sie die VPC Lattice-Zustandsprüfungen aktivieren, kann Amazon EC2 Auto Scaling eine laufende Instance ersetzen, wenn eine der VPC Lattice-Zielgruppen, die Sie der Auto Scaling-Gruppe zuordnen, diese als fehlerhaft meldet. Weitere Informationen finden Sie unter Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe.

> **Important**
>
> Bevor Sie fortfahren, müssen Sie alle im vorherigen Abschnitt genannten Voraussetzungen erfüllen.

Fügen Sie eine VPC-Lattice-Zielgruppe hinzu

Sie können einer Auto Scaling Scaling-Gruppe eine oder mehrere Zielgruppen zuordnen, wenn Sie die Gruppe erstellen oder aktualisieren.

Console

Befolgen Sie die Schritte in diesem Abschnitt, um die Konsole für Folgendes zu verwenden:

- Anhängen einer VPC Lattice-Zielgruppe an eine Auto-Scaling-Gruppe
- Aktivieren der Zustandsprüfungen für VPC Lattice

So fügen Sie eine VPC Lattice-Zielgruppe einer neuen Auto Scaling-Gruppe hinzu

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.

2. Wählen Sie in der Navigationsleiste oben die AWS-Region aus, in der Sie Ihre Gruppe erstellt haben.
3. Wählen Sie Erstellen einer Auto-Scaling-Gruppe aus.
4. Wählen Sie in den Schritten 1 und 2 die gewünschten Optionen aus und fahren Sie mit Schritt 3: Konfigurieren von erweiterten Optionen fort.
5. Wählen Sie für VPC Lattice-Integrationsoptionen die Option An VPC Lattice-Dienst anhängen.
6. Wählen Sie unter VPC Lattice-Zielgruppe auswählen Ihre Zielgruppe aus.
7. (Optional) Wählen Sie für Zustandsprüfungen und Zusätzliche Zustandsprüfungstypen die Option VPC Lattice-Zustandsprüfungen aktivieren aus.
8. (Optional) Geben Sie unter Karenzzeit für die Zustandsprüfung die Zeit in Sekunden ein. So lange muss Amazon EC2 Auto Scaling warten, bevor der Zustand einer Instance überprüft wird nachdem Sie den Zustand `InService` erreicht hat. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.
9. Fahren Sie mit dem Erstellen der Auto-Scaling-Gruppe fort. Ihre Instances werden automatisch in der VPC Lattice-Zielgruppe registriert, nachdem die Auto-Scaling-Gruppe erstellt wurde.

So fügen Sie eine VPC Lattice-Zielgruppe einer bestehenden Auto Scaling-Gruppe hinzu

Gehen Sie wie folgt vor, um eine Zielgruppe für einen Dienst an eine vorhandene Auto-Scaling-Gruppe anzuhängen.

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben Ihrer Auto-Scaling-Gruppe.

   Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.
3. Wählen Sie auf der Registerkarte Details die Option VPC Lattice-Integrationsoptionen und Bearbeiten aus.
4. Wählen Sie für VPC Lattice-Integrationsoptionen die Option An VPC Lattice-Dienst anhängen aus.
5. Wählen Sie unter VPC Lattice-Zielgruppe auswählen Ihre Zielgruppe aus.
6. Wählen Sie Aktualisieren.

Wenn Sie mit dem Hinzufügen der Zielgruppe fertig sind, können Sie optional die Zustandsprüfungen aktivieren, die diese Zielgruppe verwenden.

So aktivieren Sie die VPC Lattice Zustandsprüfungen

1. Wählen Sie auf der Registerkarte Details die Option Zustandsprüfungen, Bearbeiten aus.
2. Wählen Sie für Zustandsprüfungen und Zusätzliche Zustandsprüfungstypen die Option VPC Lattice-Zustandsprüfungen aktivieren aus.
3. Geben Sie unter Frist für Zustandsprüfungen die Zeit in Sekunden ein. So lange muss Amazon EC2 Auto Scaling warten, bevor der Zustand einer Instance überprüft wird nachdem Sie den Zustand `InService` erreicht hat. Weitere Informationen finden Sie unter Legen Sie die Wartefrist für die Zustandsprüfung einer Auto-Scaling-Gruppe fest.
4. Wählen Sie Aktualisieren.

AWS CLI

Folgen Sie den Schritten in diesem Abschnitt, um Folgendes AWS CLI zu verwenden:

- Anhängen einer VPC Lattice-Zielgruppe an eine Auto-Scaling-Gruppe
- Aktivieren der Zustandsprüfungen für VPC Lattice

So fügen Sie eine VPC Lattice-Zielgruppe einer Auto Scaling-Gruppe zu

Verwenden Sie den folgenden Befehl create-auto-scaling-group, um eine Auto Scaling -Gruppe zu erstellen und gleichzeitig eine VPC Lattice-Zielgruppe anzuhängen, indem Sie ihren Amazon-Ressourcennamen (ARN) angeben.

Ersetzen Sie die Beispielwerte für `--auto-scaling-group-name`, `--vpc-zone-identifier`, `--min-size` und `--max-size`. Für die Option `--launch-template` ersetzen Sie *`my-launch-template`* und *`1`* durch den Namen und die Version der Startvorlage, die Sie für Instances erstellt haben, die in einer VPC Lattice-Zielgruppe registriert sind. Bei der Option `--traffic-sources` ersetzen Sie den Beispiel-ARN durch den ARN Ihrer VPC Lattice-Zielgruppe.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
  --min-size 1 --max-size 5 \
```

```
  --traffic-sources "Identifier=arn:aws:vpc-lattice:region:account-id:targetgroup/
tg-0e2f2665eEXAMPLE"
```

Verwenden Sie den folgenden Befehl attach-traffic-sources, um eine VPC Lattice-Zielgruppe an eine Auto-Scaling-Gruppe anzuhängen, nachdem sie bereits erstellt wurde.

```
aws autoscaling attach-traffic-sources --auto-scaling-group-name my-asg \
  --traffic-sources "Identifier=arn:aws:vpc-lattice:region:account-id:targetgroup/
tg-0e2f2665eEXAMPLE"
```

So aktivieren Sie die Zustandsprüfungen für VPC Lattice

Wenn Sie eine anwendungsbasierte Zustandsprüfung für Ihre VPC Lattice-Zielgruppe konfiguriert haben, können Sie diese Zustandsprüfung aktivieren. Verwenden Sie den Befehl create-auto-scaling-group oder update-auto-scaling-group mit der Option `--health-check-type` und einem Wert von **`VPC_LATTICE`**. Um den Kulanzzeitraum für die von Ihrer Auto-Scaling-Gruppe durchgeführten Integritätsprüfungen anzugeben, schließen Sie die Option `--health-check-grace-period` ein und geben Sie ihren Wert in Sekunden an.

```
--health-check-type "VPC_LATTICE" --health-check-grace-period 60
```

Eine VPC-Lattice-Zielgruppe abtrennen

Wenn Sie VPC Lattice nicht mehr benötigen, gehen Sie wie folgt vor, um die Zielgruppe von Ihrer Auto-Scaling-Gruppe zu trennen.

Console

Befolgen Sie die Schritte in diesem Abschnitt, um die Konsole für Folgendes zu verwenden:

- Trennen einer VPC Lattice-Zielgruppe von einer Auto-Scaling-Gruppe
- Schalten Sie die Zustandsprüfungen für VPC Lattice aus

So trennen Sie eine VPC Lattice-Zielgruppe von einer Auto-Scaling-Gruppe

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/ und wählen Sie im Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus.
2. Aktivieren Sie das Kontrollkästchen neben einer vorhandenen Gruppe.

Im unteren Teil der Seite wird ein geteilter Bereich geöffnet.

3. Wählen Sie auf der Registerkarte Details die Option VPC Lattice-Integrationsoptionen und Bearbeiten aus.
4. Wählen Sie unter VPC Lattice Integration Options das Löschsymbol (X) neben der Zielgruppe aus.
5. Wählen Sie Aktualisieren.

Wenn Sie mit dem Trennen der Zielgruppe fertig sind, können Sie die VPC Lattice-Zustandsprüfungen deaktivieren.

So deaktivieren Sie die VPC Lattice Zustandsprüfungen

1. Wählen Sie auf der Registerkarte Details die Option Zustandsprüfungen, Bearbeiten aus.
2. Deaktivieren Sie für Zustandsprüfungen und Zusätzliche Zustandsprüfungstypen die Option VPC Lattice-Zustandsprüfungen aktivieren.
3. Wählen Sie Aktualisieren.

AWS CLI

Folgen Sie den Schritten in diesem Abschnitt, um Folgendes AWS CLI zu verwenden:

- Trennen einer VPC Lattice-Zielgruppe von einer Auto-Scaling-Gruppe
- Schalten Sie die Zustandsprüfungen für VPC Lattice aus

Verwenden Sie den detach-traffic-sources-Befehl, um eine Zielgruppe von Ihrer Auto-Scaling-Gruppe zu trennen, wenn Sie sie nicht mehr benötigen.

```
aws autoscaling detach-traffic-sources --auto-scaling-group-name my-asg \
  --traffic-sources "Identifier=arn:aws:vpc-lattice:region:account-id:targetgroup/
tg-0e2f2665eEXAMPLE"
```

Um die Zustandsprüfungen einer Auto Scaling-Gruppe zu aktualisieren, so dass sie keine VPC Lattice-Zustandsprüfungen mehr verwendet, verwenden Sie den Befehl update-auto-scaling-group. Geben Sie die Option `--health-check-type` und den Wert **`EC2`** ein.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
```

```
  --health-check-type "EC2"
```

Überprüfen Sie den Anhangsstatus Ihrer VPC Lattice-Zielgruppe

Nachdem Sie eine VPC Lattice-Zielgruppe an eine Auto-Scaling-Gruppe angehängt haben, wechselt sie bei der Registrierung der Instances in der Gruppe in den `Adding`-Status. Wenn alle Instances der Gruppe angemeldet sind, wird sie in den Status `Added` versetzt. Besteht zumindest eine angemeldete Instance die Zustandsprüfungen, wird er in den Status `InService` versetzt. Wenn sich die Zielgruppe im Zustand `InService` befindet, kann Amazon EC2 Auto Scaling alle als fehlerhaft gemeldeten Instances beenden und ersetzen. Wenn keine registrierten Instances die Zustandsprüfungen bestehen (z. B. aufgrund einer falsch konfigurierten Zustandsprüfung), tritt die Zielgruppe nicht in den Zustand `InService` ein. Amazon EC2 Auto Scaling beendet und ersetzt die Instances nicht.

Wenn Sie eine Zielgruppe für einen Dienst abtrennen, wird sie in den Status `Removing` versetzt, solange sie die Instances der Gruppe abmeldet. Nach der Abmeldung werden die Instances weiterhin ausgeführt. Standardmäßig ist Connection Draining (Verzögerung der Registrierungsaufhebung) aktiviert. Ist Connection Draining aktiviert, wartet VPC Lattice darauf, dass aktive Anforderungen abgeschlossen werden oder das maximale Zeitlimit abgelaufen ist (je nachdem, was zuerst eintritt), bevor die Instances abgemeldet werden.

Sie können den Status des Anhangs mit den SDKs AWS Command Line Interface (AWS CLI) oder den AWS SDKs überprüfen. Sie können den Status des Anhangs nicht von der Konsole aus überprüfen.

Um den Status des Anhangs AWS CLI zu überprüfen

Der folgende Befehl describe-traffic-sources gibt den Anhangsstatus aller Traffic-Quellen für die angegebene Auto-Scaling-Gruppe zurück.

```
aws autoscaling describe-traffic-sources --auto-scaling-group-name my-asg
```

Das Beispiel gibt den ARN der VPC Lattice-Zielgruppe zurück, die an die Auto-Scaling-Gruppe angehängt ist, zusammen mit dem Anhangsstatus der Zielgruppe im Element `State`.

```
{
    "TrafficSources": [
        {
```

```
            "Identifier": "arn:aws:vpc-lattice:region:account-
id:targetgroup/tg-0e2f2665eEXAMPLE",
            "State": "InService",
            "Type": "vpc-lattice"
        }
    ]
}
```

Wird EventBridge zur Behandlung von Auto Scaling Scaling-Ereignissen verwendet

Amazon EventBridge, früher CloudWatch Events genannt, hilft Ihnen bei der Einrichtung ereignisgesteuerter Regeln, mit denen Ressourcen überwacht und Zielaktionen initiiert werden, die andere AWS Dienste nutzen.

Ereignisse von Amazon EC2 Auto Scaling werden nahezu EventBridge in Echtzeit übermittelt. Sie können EventBridge Regeln einrichten, die als Reaktion auf eine Vielzahl dieser Ereignisse programmgesteuerte Aktionen und Benachrichtigungen aufrufen. Während Instances beispielsweise gerade gestartet oder beendet werden, können Sie eine AWS Lambda Funktion aufrufen, um eine vorkonfigurierte Aufgabe auszuführen.

Zu den Zielen von EventBridge Regeln können AWS Lambda Funktionen, Amazon SNS SNS-Themen, API-Ziele, Event-Busse usw. gehören. AWS-Konten Informationen zu unterstützten Zielen finden Sie unter EventBridge Amazon-Ziele im EventBridge Amazon-Benutzerhandbuch.

Erstellen Sie zunächst EventBridge Regeln anhand eines Beispiels unter Verwendung eines Amazon SNS SNS-Themas und einer EventBridge Regel. Wenn ein Benutzer dann eine Aktualisierung der Instance startet, benachrichtigt Amazon SNS Sie per E-Mail, wenn ein Checkpoint erreicht wird. Weitere Informationen finden Sie unter Erstellen Sie EventBridge Regeln für Instance-Aktualisierungsereignisse.

Inhalt

Ereignis-Referenz für Amazon EC2 Auto Scaling

Mit Amazon können Sie Regeln erstellen EventBridge, die eingehenden Ereignissen entsprechen, und diese zur Verarbeitung an Ziele weiterleiten.

Inhalt

Lebenszyklus-Aktions-Ereignisse

Wenn Sie Ihrer Auto Scaling-Gruppe Lifecycle-Hooks hinzufügen, sendet Amazon EC2 Auto Scaling Ereignisse an den EventBridge Zeitpunkt, an dem eine Instance in einen Wartestatus übergeht. Ereignisse werden auf die bestmögliche Weise ausgegeben.

Ereignistypen

Aufskalierungs-Lebenszyklus-Aktion

Im folgenden Beispielereignis hat Amazon EC2 Auto Scaling eine Instance aufgrund eines Start-Lebenszyklus-Hooks in einen `Pending:Wait`-Status versetzt.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
```

```
    "LifecycleActionToken": "87654321-4321-4321-4321-210987654321",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
    "NotificationMetadata": "additional-info",
    "Origin": "EC2",
    "Destination": "AutoScalingGroup"
  }
}
```

Herunterskalierungs-Lebenszyklus-Aktion

Im folgenden Beispielereignis hat Amazon EC2 Auto Scaling eine Instance aufgrund eines Beendigungs-Lebenszyklus-Hooks in einen `Terminating:Wait`-Status versetzt.

> **Important**
>
> Wenn eine Auto-Scaling-Gruppe die Instances beim Abskalieren an einen warmen Pool zurückgibt, kann die Rückgabe von Instances an den warmen Pool auch `EC2 Instance-terminate Lifecycle Action`-Ereignisse erzeugen. Ereignisse, die ausgelöst werden, wenn eine Instance beim Abskalieren in den Wartestatus wechselt, haben `WarmPool` als den Wert für `Destination`. Weitere Informationen finden Sie unter Instance reuse policy.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-terminate Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "LifecycleActionToken":"87654321-4321-4321-4321-210987654321",
    "AutoScalingGroupName":"my-asg",
    "LifecycleHookName":"my-lifecycle-hook",
    "EC2InstanceId":"i-1234567890abcdef0",
    "LifecycleTransition":"autoscaling:EC2_INSTANCE_TERMINATING",
```

```
    "NotificationMetadata":"additional-info",
    "Origin": "AutoScalingGroup",
    "Destination": "EC2"
  }
}
```

Erfolgreiche Skalierungsereignisse

Die folgenden Beispiele zeigen die Ereignistypen für erfolgreiche Skalierungsereignisse. Ereignisse werden auf die bestmögliche Weise ausgegeben.

Ereignistypen

- Erfolgreiche Aufskalierungsereignisse
- Erfolgreiche Abskalierungsereignisse

Erfolgreiche Aufskalierungsereignisse

Im folgenden Beispielereignis hat Amazon EC2 Auto Scaling erfolgreich eine Instance gestartet.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn",
    "instance-arn"
  ],
  "detail": {
    "StatusCode": "InProgress",
    "Description": "Launching a new EC2 instance: i-12345678",
    "AutoScalingGroupName": "my-asg",
    "ActivityId": "87654321-4321-4321-4321-210987654321",
    "Details": {
      "Availability Zone": "us-west-2b",
      "Subnet ID": "subnet-12345678"
    },
    "RequestId": "12345678-1234-1234-1234-123456789012",
    "StatusMessage": "",
```

```
    "EndTime": "yyyy-mm-ddThh:mm:ssZ",
    "EC2InstanceId": "i-1234567890abcdef0",
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "Cause": "description-text",
    "Origin": "EC2",
    "Destination": "AutoScalingGroup"
  }
}
```

Erfolgreiche Abskalierungsereignisse

Im folgenden Beispielereignis hat Amazon EC2 Auto Scaling erfolgreich eine Instance beendet.

> **Important**
>
> Wenn eine Auto-Scaling-Gruppe die Instances beim Abskalieren an einen warmen Pool zurückgibt, kann die Rückgabe von Instances an den warmen Pool auch `EC2 Instance Terminate Successful`-Ereignisse erzeugen. Ereignisse, die ausgelöst werden, wenn eine Instance erfolgreich in den warmen Pool zurückkehrt, haben `WarmPool` als den Wert für `Destination`. Weitere Informationen finden Sie unter Instance reuse policy.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Terminate Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn",
    "instance-arn"
  ],
  "detail": {
    "StatusCode": "InProgress",
    "Description": "Terminating EC2 instance: i-12345678",
    "AutoScalingGroupName": "my-asg",
    "ActivityId": "87654321-4321-4321-4321-210987654321",
    "Details": {
      "Availability Zone": "us-west-2b",
      "Subnet ID": "subnet-12345678"
```

```
    },
    "RequestId": "12345678-1234-1234-1234-123456789012",
    "StatusMessage": "",
    "EndTime": "yyyy-mm-ddThh:mm:ssZ",
    "EC2InstanceId": "i-1234567890abcdef0",
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "Cause": "description-text",
    "Origin": "AutoScalingGroup",
    "Destination": "EC2"
  }
}
```

Fehlgeschlagene Skalierungsereignisse

Die folgenden Beispiele zeigen die Ereignistypen für fehlgeschlagene Skalierungsereignisse. Ereignisse werden auf die bestmögliche Weise ausgegeben.

Ereignistypen

- Fehlgeschlagene Aufskalierungsereignisse
- Fehlgeschlagene Abskalierungsereignisse

Fehlgeschlagene Aufskalierungsereignisse

Im folgenden Beispielereignis ist der Start einer Instance durch Amazon EC2 Auto Scaling fehlgeschlagen.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Launch Unsuccessful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn",
    "instance-arn"
  ],
  "detail": {
    "StatusCode": "Failed",
    "AutoScalingGroupName": "my-asg",
```

```
    "ActivityId": "87654321-4321-4321-4321-210987654321",
    "Details": {
      "Availability Zone": "us-west-2b",
      "Subnet ID": "subnet-12345678"
    },
    "RequestId": "12345678-1234-1234-1234-123456789012",
    "StatusMessage": "message-text",
    "EndTime": "yyyy-mm-ddThh:mm:ssZ",
    "EC2InstanceId": "i-1234567890abcdef0",
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "Cause": "description-text",
    "Origin": "EC2",
    "Destination": "AutoScalingGroup"
  }
}
```

Fehlgeschlagene Abskalierungsereignisse

Im folgenden Beispielereignis ist die Beendigung einer Instance durch Amazon EC2 Auto Scaling fehlgeschlagen.

> **Important**
>
> Wenn eine Auto-Scaling-Gruppe keine Instances beim Abskalieren an einen warmen Pool zurückgibt, kann die Rückgabe von Instances an den warmen Pool auch `EC2 Instance Terminate Unsuccessful`-Ereignisse erzeugen. Ereignisse, die ausgelöst werden, wenn eine Instance nicht erfolgreich in den warmen Pool zurückkehrt, haben `WarmPool` als den Wert für `Destination`. Weitere Informationen finden Sie unter Instance reuse policy.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Terminate Unsuccessful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn",
    "instance-arn"
  ],
```

```
  "detail": {
    "StatusCode": "Failed",
    "AutoScalingGroupName": "my-asg",
    "ActivityId": "87654321-4321-4321-4321-210987654321",
    "Details": {
      "Availability Zone": "us-west-2b",
      "Subnet ID": "subnet-12345678"
    },
    "RequestId": "12345678-1234-1234-1234-123456789012",
    "StatusMessage": "message-text",
    "EndTime": "yyyy-mm-ddThh:mm:ssZ",
    "EC2InstanceId": "i-1234567890abcdef0",
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "Cause": "description-text",
    "Origin": "AutoScalingGroup",
    "Destination": "EC2"
  }
}
```

Instance-Aktualisierungsereignisse

Die folgenden Beispiele zeigen Ereignisse für das Instance-Aktualisierungs-Feature. Ereignisse werden auf die bestmögliche Weise ausgegeben.

Ereignistypen

- Prüfpunkt erreicht
- Instance-Aktualisierung gestartet
- Instance-Aktualisierung erfolgreich
- Instance-Aktualisierung fehlgeschlagen
- Instance-Aktualisierung abgebrochen
- Das Rollback der Instance-Aktualisierung wurde gestartet
- Das Rollback der Instanzaktualisierung war erfolgreich
- Das Rollback der Instanzaktualisierung ist fehlgeschlagen

Prüfpunkt erreicht

Während einer Instance-Aktualisierung sendet Amazon EC2 Auto Scaling das folgende Ereignis, wenn die Anzahl der ersetzten Instances den für den Prüfpunkt definierten prozentualen Schwellenwert erreicht.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Checkpoint Reached",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "ab00cf8f-9126-4f3c-8010-dbb8cad6fb86",
    "AutoScalingGroupName": "my-asg",
    "CheckpointPercentage": "50",
    "CheckpointDelay": "300"
  }
}
```

Instance-Aktualisierung gestartet

Amazon EC2 Auto Scaling sendet das folgende Ereignis, wenn sich der Zustand einer Instance-Aktualisierung auf `InProgress` ändert.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Started",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
    "AutoScalingGroupName": "my-asg"
  }
}
```

Instance-Aktualisierung erfolgreich

Amazon EC2 Auto Scaling sendet das folgende Ereignis, wenn sich der Zustand einer Instance-Aktualisierung auf `Successful` ändert.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Succeeded",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
    "AutoScalingGroupName": "my-asg"
  }
}
```

Instance-Aktualisierung fehlgeschlagen

Amazon EC2 Auto Scaling sendet das folgende Ereignis, wenn sich der Zustand einer Instance-Aktualisierung auf `Failed` ändert.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Failed",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
    "AutoScalingGroupName": "my-asg"
  }
```

```
}
```

Instance-Aktualisierung abgebrochen

Amazon EC2 Auto Scaling sendet das folgende Ereignis, wenn sich der Zustand einer Instance-Aktualisierung auf `Cancelled` ändert.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Cancelled",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
    "AutoScalingGroupName": "my-asg"
  }
}
```

Das Rollback der Instance-Aktualisierung wurde gestartet

Amazon EC2 Auto Scaling sendet das folgende Ereignis, wenn sich der Zustand einer Instance-Aktualisierung auf `RollbackInProgress` ändert.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Rollback Started",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
    "AutoScalingGroupName": "my-asg"
```

```
  }
}
```

Das Rollback der Instanzaktualisierung war erfolgreich

Amazon EC2 Auto Scaling sendet das folgende Ereignis, wenn sich der Zustand einer Instance-Aktualisierung auf `RollbackSuccessful` ändert.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Rollback Succeeded",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
    "AutoScalingGroupName": "my-asg"
  }
}
```

Das Rollback der Instanzaktualisierung ist fehlgeschlagen

Amazon EC2 Auto Scaling sendet das folgende Ereignis, wenn sich der Zustand einer Instance-Aktualisierung auf `Failed` ändert.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Rollback Failed",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
```

```
    "AutoScalingGroupName": "my-asg"
  }
}
```

Beispielereignisse und -muster in einem warmen Pool

Amazon EC2 Auto Scaling unterstützt mehrere vordefinierte Muster in Amazon EventBridge. Dies vereinfacht die Erstellung eines Ereignismusters. Sie wählen Feldwerte in einem Formular aus und EventBridge generieren das Muster für Sie. Derzeit unterstützt Amazon EC2 Auto Scaling keine vordefinierten Muster für Ereignisse, die von einer Auto-Scaling-Gruppe mit einem Warm-Pool ausgelöst werden. Sie müssen das Muster als JSON-Objekt eingeben. Dieser Abschnitt und das Thema Erstellen Sie EventBridge Regeln für Ereignisse im warmen Pool zeigen Ihnen, wie Sie ein Ereignismuster verwenden, um Ereignisse auszuwählen und sie an Ziele zu senden.

Um EventBridge Regeln zu erstellen, die nach Ereignissen im Zusammenhang mit warmen Pools filtern, an die Amazon EC2 Auto Scaling sendet EventBridge, fügen Sie die `Destination` Felder `Origin` und aus dem `detail` Abschnitt des Ereignisses hinzu.

Bei den Werten `Origin` und `Destination` kann es sich um Folgendes handeln:

`EC2` | `AutoScalingGroup` | `WarmPool`

Inhalt

Beispielereignisse

Wenn Sie Ihrer Auto Scaling-Gruppe Lifecycle-Hooks hinzufügen, sendet Amazon EC2 Auto Scaling Ereignisse an den EventBridge Zeitpunkt, an dem eine Instance in einen Wartestatus übergeht. Weitere Informationen finden Sie unter Verwenden von Lebenszyklus-Hooks mit einem Warm Pool.

Dieser Abschnitt enthält Beispiele für diese Ereignisse, wenn Ihre Auto-Scaling-Gruppe über einen warmen Pool verfügt. Ereignisse werden auf die bestmögliche Weise ausgegeben.

> **Note**
>
> Informationen zu Ereignissen, an die Amazon EC2 Auto Scaling EventBridge bei erfolgreicher Skalierung sendet, finden Sie unterErfolgreiche Skalierungsereignisse. Informationen zu

Ereignissen, bei denen die Skalierung nicht erfolgreich ist, finden Sie unter Fehlgeschlagene Skalierungsereignisse.

Beispiele für Ereignisse

- Aufskalierungs-Lebenszyklus-Aktion
- Herunterskalierungs-Lebenszyklus-Aktion

Aufskalierungs-Lebenszyklus-Aktion

Ereignisse, die geliefert werden, wenn eine Instance beim Aufskalieren in den Wartestatus wechselt, haben `EC2 Instance-launch Lifecycle Action` als den Wert für `detail-type`. Im `detail`-Objekt zeigen die Werte für die Attribute `Origin` und `Destination`, woher die Instance kommt und wohin sie geht.

In diesem Beispiel für ein Abskalierungsereignis wird eine neue Instance gestartet und ihr Status wird in `Warmed:Pending:Wait` geändert, weil sie dem warmen Pool hinzugefügt wird. Weitere Informationen finden Sie unter Lebenszyklusstatusübergänge für Instances in einem Warm Pool.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2021-01-13T00:12:37.214Z",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "LifecycleActionToken": "71514b9d-6a40-4b26-8523-05e7eEXAMPLE",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-launch-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
    "NotificationMetadata": "additional-info",
    "Origin": "EC2",
    "Destination": "WarmPool"
  }
```

```
}
```

In diesem Beispiel für ein Aufskalierungsereignis ändert sich der Status der Instance in `Pending:Wait`, wenn sie aus dem warmen Pool zur Auto-Scaling-Gruppe hinzugefügt wird. Weitere Informationen finden Sie unter Lebenszyklusstatusübergänge für Instances in einem Warm Pool.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2021-01-19T00:35:52.359Z",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "LifecycleActionToken": "19cc4d4a-e450-4d1c-b448-0de67EXAMPLE",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-launch-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
    "NotificationMetadata": "additional-info",
    "Origin": "WarmPool",
    "Destination": "AutoScalingGroup"
  }
}
```

Herunterskalierungs-Lebenszyklus-Aktion

Ereignisse, die geliefert werden, wenn eine Instance beim Abskalieren in den Wartestatus wechselt, haben `EC2 Instance-terminate Lifecycle Action` als den Wert für `detail-type`. Im `detail`-Objekt zeigen die Werte für die Attribute `Origin` und `Destination`, woher die Instance kommt und wohin sie geht.

In diesem Beispiel für ein Abskalierungsereignis ändert sich der Status einer Instance in `Warmed:Pending:Wait`, wenn sie an den warmen Pool zurückgegeben wird. Weitere Informationen finden Sie unter Lebenszyklusstatusübergänge für Instances in einem Warm Pool.

```
{
```

```
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-terminate Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2022-03-28T00:12:37.214Z",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "LifecycleActionToken": "42694b3d-4b70-6a62-8523-09a1eEXAMPLE",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-termination-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",
    "NotificationMetadata": "additional-info",
    "Origin": "AutoScalingGroup",
    "Destination": "WarmPool"
  }
}
```

Beispiel für Ereignismuster

Im vorstehenden Abschnitt werden Beispiel-Ereignisse aufgeführt, die von Amazon EC2 Auto Scaling ausgegeben werden.

EventBridge Ereignismuster haben dieselbe Struktur wie die Ereignisse, denen sie entsprechen. Das Muster zitiert die Felder, die Sie abgleichen möchten, und liefert die Werte, nach denen Sie suchen.

Die folgenden Felder des Ereignisses bilden das in der Regel definierte Ereignismuster, das eine Aktion aufruft:

`"source": "aws.autoscaling"`

Gibt an, dass das Ereignis aus Amazon EC2 Auto Scaling stammt.

`"detail-type": "EC2 Instance-launch Lifecycle Action"`

Identifiziert den Ereignistyp.

`"Origin": "EC2"`

Gibt an, woher die Instance kommt.

`"Destination": "WarmPool"`

Gibt an, wohin die Instance geht.

Verwenden Sie das folgende Beispiel-Ereignismuster, um alle `EC2 Instance-launch Lifecycle Action`-Ereignisse zu erfassen, die Instances zugeordnet sind, die in den warmen Pool gelangen.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "Origin": [ "EC2" ],
      "Destination": [ "WarmPool" ]
   }
}
```

Verwenden Sie das folgende Beispiel-Ereignismuster, um alle `EC2 Instance-launch Lifecycle Action`-Ereignisse zu erfassen, die mit Instances verbunden sind, die den warmen Pool aufgrund eines Aufskalierungsereignisses verlassen.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "Origin": [ "WarmPool" ],
      "Destination": [ "AutoScalingGroup" ]
   }
}
```

Verwenden Sie das folgende Beispiel-Ereignismuster, um alle `EC2 Instance-launch Lifecycle Action`-Ereignisse zu erfassen, die mit Instances verknüpft sind, die direkt in der Auto-Scaling-Gruppe gestartet werden.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "Origin": [ "EC2" ],
      "Destination": [ "AutoScalingGroup" ]
```

```
   }
}
```

Verwenden Sie das folgende Beispiel-Ereignismuster, um alle `EC2 Instance-terminate Lifecycle Action`-Ereignisse zu erfassen, die Instances zugeordnet sind, die beim Abskalieren in den warmen Pool zurückkehren.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-terminate Lifecycle Action" ],
  "detail": {
      "Origin": [ "AutoScalingGroup" ],
      "Destination": [ "WarmPool" ]
   }
}
```

Verwenden Sie das folgende Beispiel-Ereignismuster, um alle Ereignisse zu erfassen, die mit `EC2 Instance-launch Lifecycle Action` assoziiert sind, unabhängig vom Ursprung oder Ziel.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ]
}
```

EventBridge Regeln erstellen

Wenn ein Ereignis von Amazon EC2 Auto Scaling ausgelöst wird, wird eine Ereignisbenachrichtigung EventBridge als JSON-Datei an Amazon gesendet. Sie können eine EventBridge Regel schreiben, um zu automatisieren, welche Aktionen ergriffen werden, wenn ein Ereignismuster mit der Regel übereinstimmt. Wenn ein Ereignismuster EventBridge erkannt wird, das einem in einer Regel definierten Muster entspricht, EventBridge ruft es das in der Regel angegebene Ziel (oder die Ziele) auf.

Sie können die Beispielprozeduren in diesem Abschnitt als Ausgangspunkt verwenden.

Sie könnten auch die folgende Dokumentation nützlich finden.

- Informationen zum Ausführen benutzerdefinierter Aktionen für Instances beim Starten oder bevor sie mit einer Lambda-Funktion beendet werden, finden Sie unter Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft.

- Informationen zum Aufrufen einer Lambda-Funktion bei API-Aufrufen, die mit protokolliert CloudTrail wurden, finden Sie unter Tutorial: AWS API-Aufrufe protokollieren EventBridge im EventBridge Amazon-Benutzerhandbuch.
- Weitere Informationen zum Erstellen von Ereignisregeln finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erstellen von EventBridge Amazon-Regeln, die auf Ereignisse reagieren.

Themen

- Erstellen Sie EventBridge Regeln für Instance-Aktualisierungsereignisse
- Erstellen Sie EventBridge Regeln für Ereignisse im warmen Pool

Erstellen Sie EventBridge Regeln für Instance-Aktualisierungsereignisse

Im folgenden Beispiel wird eine EventBridge Regel zum Senden einer E-Mail-Benachrichtigung erstellt. Dies geschieht jedes Mal, wenn Ihre Auto-Scaling-Gruppe ein Ereignis auslöst, wenn während einer Instance-Aktualisierung ein Checkpoint erreicht wird. Das Verfahren zum Einrichten von E-Mail-Benachrichtigungen über Amazon SNS ist enthalten. Damit Sie Amazon SNS zum Versenden von E-Mail-Benachrichtigungen verwenden können, müssen Sie zunächst ein Thema erstellen und es mit Ihren E-Mail-Adressen abonnieren.

Weitere Informationen über die Instance-Aktualisierungsfunktion finden Sie unter Verwenden Sie eine Instanzaktualisierung, um Instances in einer Auto Scaling Scaling-Gruppe zu aktualisieren.

Erstellen Sie ein Amazon SNS-Thema.

Ein SNS-Thema ist ein logischer Zugriffspunkt, ein Kommunikationskanal der Auto-Scaling-Gruppe zum Versenden von Benachrichtigungen. Sie erstellen ein Thema, indem Sie einen Namen dafür angeben.

Themen-Namen müssen die folgenden Anforderungen erfüllen:

- 1-256 Zeichen enthalten
- Er muss ASCII-Buchstaben mit Groß- und Kleinschreibung, Zahlen, Unterstriche oder Bindestriche enthalten.

Weitere Informationen finden Sie unter Amazon SNS-Thema anlegen im Amazon Simple Notification Service-Entwicklerhandbuch.

Amazon SNS-Thema abonnieren

Zum Empfangen der Benachrichtigungen, die die Auto-Scaling-Gruppe an das Thema sendet, müssen Sie einen Endpunkt für das Thema abonnieren. In diesem Verfahren, für Endpoint, geben Sie die E-Mailadresse an, unter der Sie die Benachrichtigungen von Amazon EC2 Auto Scaling erhalten möchten.

Weitere Informationen finden Sie unter Amazon SNS-Thema abonnieren im Amazon Simple Notification Service-Entwicklerhandbuch.

Bestätigen Ihres Amazon SNS-Abonnements

Amazon SNS sendet eine Bestätigungs-E-Mail an die E-Mail-Adresse, die Sie im vorherigen Schritt angegeben haben.

Stellen Sie sicher, dass Sie die E-Mail unter AWS Benachrichtigungen öffnen und den Link zur Bestätigung des Abonnements auswählen, bevor Sie mit dem nächsten Schritt fortfahren.

Sie erhalten eine Bestätigungsnachricht von. AWS Amazon SNS ist jetzt so konfiguriert, dass Benachrichtigungen empfangen und als E-Mail an die angegebene E-Mail-Adresse gesendet werden.

Weiterleiten von Ereignissen an Ihr Amazon SNS-Thema

Erstellen Sie eine Regel, die den ausgewählten Ereignissen entspricht, und leiten Sie sie an Ihr Amazon SNS-Thema weiter, um abonnierte E-Mail-Adressen zu benachrichtigen.

So erstellen Sie eine Regel, die Benachrichtigungen an Ihr Amazon-SNS-Thema sendet

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.
3. Wählen Sie Regel erstellen aus.
4. Zum Define rule detail (Festlegen der Regeldetails) gehen Sie folgendermaßen vor:

    a. Geben Sie für die Regel einen Name (Namen) und optional eine Beschreibung ein.

        Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.

    b. Bei Event bus (Ereignisbus) wählen Sie default (Standard) aus. Wenn ein AWS Service in Ihrem Konto ein Ereignis generiert, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.

c. Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus.

d. Wählen Sie Weiter aus.

5. Bei Build event pattern (Ereignis-Muster erstellen) gehen Sie wie folgt vor:

   a. Wählen Sie als Eventquelle AWS Events oder EventBridge Partnerevents aus.

   b. Bei Build event pattern (Ereignis-Muster erstellen) gehen Sie wie folgt vor:

      i. Wählen Sie für Ereignisquelle die Option AWS-Services aus.

      ii. Für AWS-Service wählen Sie Auto Scaling aus.

      iii. Wählen Sie für Event type (Ereignistyp) die Option Instance Refresh (Instance-Aktualisierung aus.

      iv. Standardmäßig entspricht die Regel jedem Instance-Aktualisierungsereignis. Um eine Regel zu erstellen, die Sie benachrichtigt, wenn während einer Instance-Aktualisierung ein Checkpoint erreicht wird, wählen Sie Specific instance event(s) (Spezifische Instance-Ereignisse) und dann Auto-Scaling-Instance-Aktualisierung für EC2 aus.

      v. Standardmäßig stimmt die Regel mit jeder Auto-Scaling-Gruppe in der Region überein. Damit die Regel mit einer bestimmten Auto-Scaling-Gruppe übereinstimmt, wählen Sie Specific group name(s) und dann eine oder mehrere Auto-Scaling-Gruppen aus.

      vi. Wählen Sie Weiter aus.

6. Bei Select target(s) (Ziel(e) auswählen) gehen Sie wie folgt vor:

   a. Für Target types (Zieltypen), wählen Sie AWS-Service aus.

   b. Für Select a target (Wählen Sie ein Ziel aus), wählen Sie SNS-Thema aus.

   c. Für Topic (Thema), wählen Sie Ihr Amazon-SNS-Thema aus.

   d. (Optional) Unter Additional settings (Zusätzliche Einstellungen) können Sie optional zusätzliche Einstellungen konfigurieren. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erstellen von EventBridge Amazon-Regeln, die auf Ereignisse reagieren.

   e. Wählen Sie Weiter aus.

7. (Optional) Bei Tags können Sie Ihrer Regel optional einen Tag oder mehrere Tags hinzufügen und dann Next (Weiter) auswählen.

8. Für Review and create (Überprüfen und erstellen), überprüfen Sie die Details der Regel und ändern Sie sie nach Bedarf. Wählen Sie dann Create rule (Regel erstellen).

Erstellen Sie EventBridge Regeln für Ereignisse im warmen Pool

Im folgenden Beispiel wird eine EventBridge Regel zum Aufrufen programmatischer Aktionen erstellt. Dies geschieht jedes Mal, wenn Ihre Auto-Scaling-Gruppe ein Ereignis ausgibt, wenn eine neue Instance zum Warm-Pool hinzugefügt wird.

Bevor Sie die Regel erstellen, erstellen Sie die AWS Lambda Funktion, die die Regel als Ziel verwenden soll. Sie müssen diese Funktion als Ziel für die Regel angeben. Das folgende Verfahren enthält nur die Schritte zum Erstellen der EventBridge Regel, die wirksam wird, wenn neue Instanzen in den warmen Pool gelangen. Ein einführendes Tutorial, das Ihnen zeigt, wie Sie eine einfache Lambda-Funktion erstellen, die aufgerufen wird, wenn ein eingehendes Ereignis einer Regel entspricht, finden Sie unter Tutorial: Konfigurieren eines Lebenszyklus-Hook, der eine Lambda-Funktion aufruft.

Weitere Informationen zum Erstellen und Arbeiten mit Warm-Pools finden Sie unter Warm-Pools für Amazon EC2 Auto Scaling.

So erstellen Sie eine Ereignisregel, die eine Lambda-Funktion aufruft

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.
3. Wählen Sie Regel erstellen aus.
4. Zum Define rule detail (Festlegen der Regeldetails) gehen Sie folgendermaßen vor:
    a. Geben Sie für die Regel einen Name (Namen) und optional eine Beschreibung ein.

       Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
    b. Bei Event bus (Ereignisbus) wählen Sie default (Standard) aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis generiert, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.
    c. Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus.
    d. Wählen Sie Weiter aus.
5. Bei Build event pattern (Ereignis-Muster erstellen) gehen Sie wie folgt vor:
    a. Wählen Sie als Eventquelle AWS Events oder EventBridge Partnerevents aus.

b. Für Event pattern (Ereignismuster), wählen Sie Custom pattern (JSON editor) (Benutzerdefiniertes Muster (JSON-Editor)) und fügen Sie das folgende Muster in das Event pattern (Ereignismuster) und ersetzt den Text in *Kursivschrift* mit dem Namen Ihrer Auto-Scaling-Gruppe.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "AutoScalingGroupName": [ "my-asg" ],
      "Origin": [ "EC2" ],
      "Destination": [ "WarmPool" ]
   }
}
```

Um eine Regel zu erstellen, die mit anderen Ereignissen übereinstimmt, ändern Sie das Ereignismuster. Weitere Informationen finden Sie unter Beispiel für Ereignismuster.

c. Wählen Sie Weiter.

6. Bei Select target(s) (Ziel(e) auswählen) gehen Sie wie folgt vor:

a. Für Target types (Zieltypen), wählen Sie AWS-Service aus.

b. Für Select a target (Ein Ziel auswählen), wählen Sie Lambda function (Lambda-Funktion) aus.

c. Für Function (Funktion) wählen Sie die Funktion aus, an die Sie die Ereignisse senden möchten.

d. (Optional) Für Configure version/alias (Version/Alias konfigurieren), geben Sie Versions- und Aliaseinstellungen für die Ziel-Lambda-Funktion ein.

e. (Optional) Für Additional settings (Zusätzliche Einstellungen), geben Sie je nach Bedarf zusätzliche Einstellungen für Ihre Anwendung ein. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erstellen von EventBridge Amazon-Regeln, die auf Ereignisse reagieren.

f. Wählen Sie Weiter aus.

7. (Optional) Bei Tags können Sie Ihrer Regel optional einen Tag oder mehrere Tags hinzufügen und dann Next (Weiter) auswählen.

8. Für Review and create (Überprüfen und erstellen), überprüfen Sie die Details der Regel und ändern Sie sie nach Bedarf. Wählen Sie dann Create rule (Regel erstellen).

Stellen Sie Netzwerkkonnektivität für Ihre Auto-Scaling-Instances mit Amazon VPC bereit

Amazon Virtual Private Cloud (Amazon VPC) ist ein Service, mit dem Sie AWS Ressourcen wie Auto Scaling Scaling-Gruppen in einem logisch isolierten virtuellen Netzwerk starten können, das Sie definieren.

Ein Subnetz in einer Amazon VPC ist eine Unterteilung innerhalb einer Availability Zone, die durch ein Segment des IP-Adressbereichs der VPC definiert wird. Mithilfe von Subnetzen können Sie Instances auf Grundlage Ihrer Sicherheits- und Betriebsanforderungen gruppieren. Subnetze befinden sich vollständig innerhalb der Availability Zone, in der sie erstellt wurden. Sie starten Auto-Scaling-Instances innerhalb der Subnetze.

Sie müssen ein Internet-Gateway erstellen und Ihrem VPC hinzufügen, um die Kommunikation zwischen dem Internet und den Instances in den Subnetzen zu aktivieren. Ein Internet-Gateway ermöglicht den Ressourcen innerhalb der Subnetze das Herstellen einer Verbindung mit dem Internet durch den Edge des Amazon EC2-Netzwerks. Wird der Datenverkehr eines Subnetzes an ein Internet-Gateway weitergeleitet, wird das Subnetz als öffentliches Subnetz bezeichnet. Wird der Datenverkehr eines Subnetzes nicht an ein Internet-Gateway weitergeleitet, wird das Subnetz als privates Subnetz bezeichnet. Verwenden Sie öffentliche Subnetze für Ressourcen, die mit dem Internet verbunden sein müssen, und private Subnetze für Ressourcen, die nicht mit dem Internet verbunden sein müssen. Weitere Informationen zur Bereitstellung von Internetzugriff auf Instances in einer VPC finden Sie unter Zugriff auf das Internet im Amazon VPC-Benutzerhandbuch.

Inhalt

Standard-VPC

Wenn Sie Ihre Auto Scaling-Gruppe AWS-Konto nach dem 4. Dezember 2013 erstellt haben oder wenn Sie Ihre Auto Scaling Scaling-Gruppe in einer neuen erstellen AWS-Region, erstellen wir eine Standard-VPC für Sie. Diese Standard-VPC verfügt über jeweils ein Standard-Subnetz pro Availability Zone. Verfügen Sie über eine Standard-VPC, wird die Auto-Scaling-Gruppe standardmäßig in der Standard-VPC erstellt.

Sie können Ihre VPCs auf der Seite Ihre VPCs der Amazon-VPC-Konsole anzeigen.

Weitere Informationen über die Standard-VPC finden Sie unter Standard-VPC im Benutzerhandbuch zu Amazon VPC.

Nicht standardmäßige VPC

Sie können zusätzliche VPCs erstellen, indem Sie die Seite VPC-Dashboard im AWS Management Console aufrufen und Create VPC (VPC erstellen) auswählen.

Weitere Informationen finden Sie im Amazon VPC-Benutzerhandbuch.

> **Note**
>
> Eine VPC umfasst alle Availability Zones in ihrer AWS-Region. Wenn Sie Ihrer VPC-Subnetze hinzufügen, wählen Sie mehrere Availability Zones aus, um sicherzustellen, dass die in diesen Subnetzen gehosteten Anwendungen hochverfügbar sind. Eine Availability Zone ist eines oder mehrere diskrete Rechenzentren mit redundanter Stromversorgung, Vernetzung und Konnektivität in einem AWS-Region. Availability Zones helfen Ihnen dabei, Produktionsanwendungen hochverfügbar, fehlertolerant und skalierbar zu machen.

Überlegungen bei der Auswahl von VPC-Subnetzen

Beachten Sie die folgenden Überlegungen bei der Auswahl von VPC-Subnetzen für Ihre Auto-Scaling-Gruppe:

- Wenn Sie Ihrer Auto-Scaling-Gruppe einen Elastic Load Balancing-Load Balancer zuordnen, können die Instances entweder in öffentlichen oder privaten Subnetzen gestartet werden. Der Load Balancer muss jedoch in öffentlichen Subnetzen erstellt werden, um die DNS-Auflösung zu unterstützen.

- Wenn Sie direkt über SSH auf Ihre Auto-Scaling-Instances zugreifen, können die Instances nur in öffentlichen Subnetzen gestartet werden.
- Wenn Sie mit AWS Systems Manager Session Manager auf Auto Scaling Scaling-Instances ohne Zugriff zugreifen, können die Instances entweder in öffentlichen oder privaten Subnetzen gestartet werden.
- Wenn Sie private Subnetze verwenden, können Sie den Auto-Scaling-Instances erlauben, über ein öffentliches NAT-Gateway auf das Internet zuzugreifen.
- Standardmäßig sind die Standardsubnetze in einer Standard-VPC öffentliche Subnetze.

IP-Adressierung in einer VPC

Beim Starten von Auto-Scaling-Instances in einer VPC wird den Instances automatisch eine private IP-Adresse aus dem CIDR-Breich des Subnetzes, in dem die Instance gestartet ist, zugewiesen. Dies ermöglicht den Instances die Kommunikation mit anderen Instances in der VPC.

Sie können eine Startvorlage oder Startkonfiguration so konfigurieren, dass sie Instances öffentliche IPv4-Adressen zuweist. Wenn Sie Ihren Instances öffentliche IP-Adressen zuweisen, können sie mit dem Internet oder anderen Diensten kommunizieren. AWS

Werden öffentliche IP-Adressen für Instances aktiviert und in einem Subnetz gestartet, das für die automatische Zuweisung von IPv6-Adressen an Instances konfiguriert ist, empfangen sie IPv4- und IPv6-Adressen. Andernfalls empfangen sie nur IPv4-Adressen. Weitere Informationen finden Sie unter IPv6-Adressen im Amazon-EC2-Benutzerhandbuch.

Weitere Informationen zum Angeben von CIDR-Bereichen für Ihre VPC oder ein Subnetz finden Sie im Amazon VPC-Benutzerhandbuch.

Amazon EC2 Auto Scaling kann beim Start der Instance automatisch zusätzliche private IP-Adressen zuweisen, wenn Sie eine Startvorlage verwenden, die zusätzliche Netzwerkschnittstellen angibt. Für jede Netzwerkschnittstelle wird eine einzelne private IP-Adresse aus dem CIDR-Bereich des Subnetzes zugewiesen, in dem die Instance gestartet wird. In diesem Fall kann das System der primären Netzwerkschnittstelle keine öffentliche IPv4-Adresse mehr automatisch zuweisen. Sie können die Instances nur dann über eine öffentliche IPv4-Adresse verbinden, wenn Sie den Auto-Scaling-Instances verfügbare elastische IP-Adressen zuweisen.

Netzwerkschnittstellen in einer VPC

Jede Instance in Ihrer VPC verfügt über eine Standard-Netzwerkschnittstelle (die primäre Netzwerkschnittstelle). Sie können eine primäre Netzwerkschnittstelle nicht von einer Instance trennen. Sie können eine zusätzliche Netzwerkschnittstelle erstellen und diese an eine Instance in Ihrer VPC anfügen. Die Anzahl der anfügbaren Netzwerkschnittstellen ist je nach Instance-Typ unterschiedlich.

Beim Starten einer Instance mithilfe einer Startvorlage können Sie zusätzliche Netzwerkschnittstellen angeben. Beim Starten einer Auto-Scaling-Instance mit mehreren Netzwerkschnittstellen wird jedoch automatisch jede Schnittstelle im selben Subnetz wie die Instance erstellt. Dies liegt daran, dass Amazon EC2 Auto Scaling die in der Startvorlage definierten Subnetze zugunsten der in der Auto-Scaling-Gruppe angegebenen Subnetze ignoriert. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.

Wenn Sie eine oder mehrere Netzwerkschnittstellen aus demselben Subnetz erstellen oder an eine Instance anfügen, kann es zu Netzwerkproblemen, z. B. asymmetrischem Routing, kommen. Dies gilt insbesondere bei Instances, die eine Variante von Linux verwenden, die nicht von Amazon stammt. Wenn Sie diese Art von Konfiguration benötigen, müssen Sie die sekundäre Netzwerkschnittstelle innerhalb des Betriebssystems konfigurieren. Ein Beispiel finden Sie unter Wie kann ich dafür sorgen, dass meine sekundäre Netzwerkschnittstelle in meiner Ubuntu EC2-Instance funktioniert? im AWS Knowledge Center.

Tenancy zur Instance-Platzierung

Standardmäßig werden alle Instances in der VPC als gemeinsame Tenancy-Instances ausgeführt. Amazon EC2 Auto Scaling unterstützt auch Dedicated Instances und Dedicated Hosts. Weitere Informationen finden Sie unter Erstellen einer Startvorlage mithilfe erweiterter Einstellungen.

AWS Outposts

AWS Outposts erweitert eine Amazon-VPC von einer AWS Region zu einem Outpost mit den VPC-Komponenten, auf die in der Region zugegriffen werden kann, darunter Internet-Gateways, virtuelle private Gateways, Amazon VPC Transit Gateways und VPC-Endpunkte. Ein Außenposten ist einer Availability Zone in der Region zugeordnet und ist eine Erweiterung dieser Availability Zone, die Sie für die Ausfallsicherheit verwenden können.

Weitere Informationen finden Sie im AWS Outposts -Benutzerhandbuch.

Ein Beispiel für die Bereitstellung einer Auto-Scaling-Gruppe, die Datenverkehr von einem Application Load Balancer innerhalb eines Außenposten bereitstellt, finden Sie im folgenden Blogbeitrag Konfigurieren eines Application Load Balancer auf AWS Outposts.

Weitere Ressourcen für Informationen über VPCs

Die folgenden Themen enthalten weitere Informationen über VPCs und Subnetze.

- Private Subnetze in einer VPC
  - Beispiel: VPC mit Servern in privaten Subnetzen und NAT
  - NAT-Gateways
- Öffentliche Subnetze in einer VPC
  - Beispiel: VPC für eine Testumgebung
  - Beispiel: VPC für Web- und Datenbankserver
- Subnetze für Ihren Application Load Balancer
  - Subnetze für Ihren Load Balancer
- Allgemeine VPC-Informationen
  - Amazon VPC User Guide
  - Verbinden von VPCs mit VPC-Peering
  - Elastic Network-Schnittstelle
  - Verwenden Sie VPC-Endpunkte für private Konnektivität

Sicherheit in Amazon EC2 Auto Scaling

Cloud-Sicherheit bei AWS hat höchste Priorität. Als - AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die entwickelt wurde, um die Anforderungen der sicherheitskritischsten Organisationen zu erfüllen.

Sicherheit ist eine geteilte Verantwortung zwischen 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 für den Schutz der Infrastruktur verantwortlich, die AWS Services in der AWS Cloud ausführt. stellt Ihnen AWS außerdem Services bereit, die Sie sicher nutzen können. Externe Prüfer testen und überprüfen im Rahmen der AWS Compliance-Programme Compliance-. Weitere Informationen zu den Compliance-Programmen, die für Amazon EC2 Auto Scaling gelten, finden Sie unter Im AWS Rahmen des Compliance-Programms zugelassene - ServicesIm.
- Sicherheit in der Cloud – Ihre Verantwortung wird durch den - AWS Service bestimmt, den Sie verwenden. 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 zeigt Ihnen, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Amazon EC2 Auto Scaling einsetzen können. Die folgenden Themen veranschaulichen, wie Sie Amazon EC2 Auto Scaling zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren auch, wie Sie andere - AWS Services verwenden, die Sie bei der Überwachung und Sicherung Ihrer Amazon EC2 Auto Scaling-Ressourcen unterstützen.

Themen

Infrastruktursicherheit in Amazon EC2 Auto Scaling

Als verwalteter Service ist Amazon EC2 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 Amazon EC2 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.

Sie können auch einen Virtual Private Cloud (VPC)-Endpunkt für Amazon EC2 Auto Scaling verwenden. Schnittstellen-VPC-Endpunkte ermöglichen es Ihren Amazon-VPC-Ressourcen, ihre privaten IP-Adressen zu verwenden, um auf Amazon EC2 Auto Scaling zuzugreifen, ohne dem öffentlichen Internet ausgesetzt zu sein. Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling und Schnittstellen-VPC-Endpunkte.

Zugehörige Ressourcen

Informationen zu Funktionen zur Isolierung des von Amazon EC2 bereitgestellten Serviceverkehrs finden Sie unter Infrastruktursicherheit in Amazon EC2 im Amazon EC2 EC2-Benutzerhandbuch.

Ausfallsicherheit in Amazon EC2 Auto Scaling

Die AWS globale -Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit niedriger Latenz, hohem Durchsatz und hoher Redundanz 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.

Gehen Sie wie folgt vor, um von der geografischen Redundanz des Availability-Zone-Designs zu profitieren:

- Verteilen Sie Ihre Auto-Scaling-Gruppe auf mehrere Availability Zones.
- Haben Sie mindestens eine funktionierende Instance in jeder Availability Zone.
- Hängen Sie einen Load Balancer an, um eingehenden Datenverkehr auf dieselben Availability Zones zu verteilen. Wenn Sie einen Application Load Balancer verwenden, stellen Sie sicher, dass jede EC2-Instance eine ähnliche Menge an Datenverkehr erhält, indem Sie zonenübergreifendes Load Balancing aktiviert lassen. Dies trägt dazu bei, die Auswirkungen einer erhöhten Last auf bestehende Instances während eines Failover-Ereignisses zu begrenzen, und führt zu einer höheren Ausfallsicherheit als ohne zonenübergreifendes Load Balancing.
- Stellen Sie sicher, dass die Elastic-Load-Balancing-Zustandsprüfungen korrekt konfiguriert sind, und dass sie für die Auto-Scaling-Gruppe aktiviert sind. Wenn eine Instance ihre Zustandsprüfung nicht besteht, wird Elastic Load Balancing den Datenverkehr nicht mehr an sie senden, sondern an fehlerfreie Instances umleiten, während Amazon EC2 Auto Scaling die fehlerhafte Instance ersetzt.

Amazon EC2 Auto Scaling unterstützt Ihre Anforderungen an die Ausfallsicherheit Ihrer Anwendungen auf folgende Weise:

- Überprüft Instances auf Zustands- und Erreichbarkeitsprobleme. Wenn eine Instance fehlerhaft wird, wird sie automatisch beendet und eine neue gestartet.
- Wenn dynamische Skalierungsrichtlinien in Kraft sind, wird die Kapazität automatisch entsprechend dem eingehenden Datenverkehr skaliert.
- Erkennt Probleme mit der Zuverlässigkeit der Amazon- CloudWatch Metriken, die Skalierungsrichtlinien unterstützen, und unterbricht Abskalierungsaktivitäten, wenn keine zuverlässigen Metriken verfügbar sind, z. B. wenn Datenpunkte fehlen.
- Versucht automatisch, die gleiche Anzahl von Instances in jeder aktivierten Availability Zone aufrechtzuerhalten, wenn Ihre Gruppe skaliert wird.

- Nutzt Availability Zones, um eine hohe Verfügbarkeit zu gewährleisten. Wenn eine Availability Zone fehlerhaft wird, geht Amazon EC2 Auto Scaling wie folgt vor:
  - Startet neue Instances in einer anderen Availability Zone, die für Ihre Auto-Scaling-Gruppe aktiviert ist.
  - Verteilt Instances gleichmäßig auf alle aktivierten Availability Zones, wenn die fehlerhafte Availability Zone in einen fehlerfreien Zustand zurückkehrt.
- Versucht weiterhin, Instances in anderen aktivierten Availability Zones zu starten, wenn eine Instance in einer bestimmten Availability Zone nicht gestartet werden kann.
- Registriert und deregistriert Instances automatisch bei den Load Balancern, die Ihrer Auto-Scaling-Gruppe zugeordnet sind. So müssen Sie Instances nicht separat registrieren und deregistrieren.

Zugehörige Ressourcen

Informationen zu Funktionen zur Unterstützung Ihrer Anforderungen an die Datenausfallsicherheit von Amazon EBS finden Sie unter Ausfallsicherheit in Amazon Elastic Block Store im Amazon-EBS-Benutzerhandbuch.

Datenschutz in Amazon EC2 Auto Scaling

Das Modell der AWS gemeinsamen Verantwortung gilt für den Datenschutz in Amazon EC2 Auto Scaling. Wie in diesem Modell beschrieben, AWS ist es verantwortlich für den Schutz der globalen Infrastruktur, auf der die AWS Cloud gesamte Infrastruktur läuft. 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 Amazon EC2 Auto Scaling oder anderen Geräten arbeiten und die Konsole AWS CLI, API oder AWS SDKs AWS-Services verwenden. 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.

Wenn Sie eine Amazon EC2 EC2-Instance starten, haben Sie die Möglichkeit, Benutzerdaten an die Instance zu übergeben, um beim Booten der Instance zusätzliche Konfigurationen vorzunehmen. Wir empfehlen außerdem, niemals vertrauliche oder sensible Informationen in die Benutzerdaten aufzunehmen, die an eine Instance weitergegeben werden.

Wird AWS KMS keys zum Verschlüsseln von Amazon EBS-Volumes verwendet

Sie können Ihre Auto Scaling-Gruppe so konfigurieren, dass Amazon EBS-Volume-Daten, die in der Cloud gespeichert sind, mit AWS KMS keys verschlüsselt werden. Amazon EC2 Auto Scaling unterstützt AWS verwaltete und vom Kunden verwaltete Schlüssel zur Verschlüsselung Ihrer Daten. Beachten Sie, dass die `KmsKeyId`-Option zum Angeben eines kundenverwalteten Schlüssels nicht verfügbar ist, wenn Sie eine Startkonfiguration verwenden. Verwenden Sie stattdessen eine Startvorlage, um den kundenverwalteten Schlüssel anzugeben. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe. Informationen zur Erstellung, Speicherung und Verwaltung Ihrer AWS KMS Verschlüsselungsschlüssel finden Sie im AWS Key Management Service Entwicklerhandbuch.

Sie können auch einen kundenverwalteten Schlüssel in Ihrem EBS-unterstützten AMI konfigurieren, bevor Sie die Startvorlage oder die Startkonfiguration einrichten. Sie können die Verschlüsselung standardmäßig verwenden, um die Verschlüsselung der neuen EBS-Volumes und Snapshot-Kopien zu erzwingen, die Sie erstellen. Weitere Informationen finden Sie unter Verschlüsselung mit EBS-gestützten AMIs verwenden im Amazon EC2 EC2-Benutzerhandbuch und Standardverschlüsselung im Amazon EBS-Benutzerhandbuch.

> **Note**
>
> Informationen zum Einrichten der Schlüsselrichtlinie, die Sie zum Starten von Auto Scaling-Instances benötigen, wenn Sie einen kundenverwalteten Schlüssel für die Verschlüsselung verwenden, finden Sie unter Erforderliche AWS KMS Schlüsselrichtlinie für die Verwendung mit verschlüsselten Volumes.

Zugehörige Ressourcen

Die von Amazon EBS bereitgestellten Datenschutzrichtlinien finden Sie unter Datenschutz im Amazon Elastic Block Store im Amazon EBS-Benutzerhandbuch.

Erforderliche AWS KMS Schlüsselrichtlinie für die Verwendung mit verschlüsselten Volumes

Amazon EC2 Auto Scaling verwendet serviceverknüpfte Rollen, um Berechtigungen an andere zu delegieren. AWS-Services Servicebezogene Rollen in Amazon EC2 Auto Scaling sind vordefiniert und beinhalten Berechtigungen, die Amazon EC2 Auto Scaling benötigt, um andere AWS-Services in Ihrem Namen anzurufen. Die vordefinierten Berechtigungen beinhalten auch den Zugriff auf Ihre. Von AWS verwaltete Schlüssel Sie enthalten jedoch keinen Zugriff auf Ihre kundenverwalteten Schlüssel, sodass Sie die volle Kontrolle über diese Schlüssel behalten können.

In diesem Thema wird beschrieben, wie Sie die Schlüsselrichtlinie einrichten, die Sie zum Starten von Auto Scaling-Instances benötigen, wenn Sie einen kundenverwalteten Schlüssel für die Amazon EBS-Verschlüsselung angeben.

> **Note**
>
> Amazon EC2 Auto Scaling benötigt keine zusätzliche Autorisierung für die Verwendung des standardmäßigen Von AWS verwalteter Schlüssel zum Schützen der verschlüsselten Volumes in Ihrem Konto.

Inhalt

Übersicht

Folgendes AWS KMS keys kann für die Amazon EBS-Verschlüsselung verwendet werden, wenn Amazon EC2 Auto Scaling Instances startet:

- Von AWS verwalteter Schlüssel— Ein Verschlüsselungsschlüssel in Ihrem Konto, das Amazon EBS erstellt, besitzt und verwaltet. Dies ist der Standardverschlüsselungsschlüssel für ein neues Konto. Der Von AWS verwalteter Schlüssel wird für die Verschlüsselung verwendet, sofern Sie keinen vom Kunden verwalteten Schlüssel angeben.
- Vom Kunden verwalteter Schlüssel — Ein benutzerdefinierter Verschlüsselungsschlüssel, den Sie erstellen, besitzen und verwalten. Weitere Informationen finden Sie unter Erstellen von Schlüsseln im AWS Key Management Service -Entwicklerhandbuch.

  Hinweis: Der Schlüssel muss symmetrisch sein. Amazon EBS unterstützt keine asymmetrischen vom Kunden verwalteten Schlüssel.

Sie konfigurieren kundenverwaltete Schlüssel, wenn Sie verschlüsselte Snapshots oder eine Startvorlage, die verschlüsselte Volumes angibt, erstellen oder wenn Sie die Verschlüsselung standardmäßig aktivieren.

Konfigurieren von Schlüsselrichtlinien

Ihre KMS-Schlüssel müssen über eine Schlüsselrichtlinie verfügen, mit der Amazon EC2 Auto Scaling-Instances mit Amazon EBS-Volumes starten kann, die mit einem kundenverwalteten Schlüssel verschlüsselt sind.

Nutzen Sie die Beispiele auf dieser Seite zum Konfigurieren einer Schlüsselrichtlinie, um Amazon EC2 Auto Scaling Zugriff auf Ihren kundenverwalteten Schlüssel zu erteilen. Sie können die Schlüsselrichtlinie des kundenverwalteten Schlüssels entweder bei Erstellung des Schlüssels oder zu einem späteren Zeitpunkt ändern.

Sie müssen Ihrer Schlüsselrichtlinie mindestens zwei Richtlinienanweisungen hinzufügen, damit er mit Amazon EC2 Auto Scaling verwendet werden kann.

- Die erste Anweisung ermöglicht, dass die im `Principal`-Element angegebene IAM-Identität den kundenverwalteten Schlüssel direkt verwendet. Er umfasst Berechtigungen zur Ausführung der AWS KMS `Encrypt`,`Decrypt`, `ReEncrypt*`,`GenerateDataKey*`, und `DescribeKey` -Operationen mit dem Schlüssel.
- Die zweite Anweisung ermöglicht es der im `Principal` Element angegebenen IAM-Identität, den `CreateGrant` Vorgang zum Generieren von Zuschüssen zu verwenden, die eine Teilmenge ihrer eigenen Berechtigungen an Personen delegieren AWS-Services , die in AWS KMS oder einen anderen Principal integriert sind. Auf diese Weise können sie den Schlüssel verwenden, um in Ihrem Namen verschlüsselte Ressourcen zu erstellen.

Ändern Sie beim Hinzufügen der neuen Richtlinienanweisungen zur Schlüsselrichtlinie keinen der vorhandenen Anweisungen in der Richtlinie.

Für jedes der folgenden Beispiele werden Argumente, die ersetzt werden müssen, wie z. B. eine Schlüssel-ID oder der Name einer dienstbezogenen Rolle, als Benutzerplatzhaltertext angezeigt. In den meisten Fällen können Sie den Namen der serviceverknüpften Rolle durch den Namen einer serviceverknüpften Amazon EC2 Auto Scaling-Rolle ersetzen.

Weitere Informationen finden Sie in den folgenden Ressourcen:

- Informationen zum Erstellen eines Schlüssels mit dem finden Sie unter AWS CLI create-key.
- Informationen zum Aktualisieren einer Schlüsselrichtlinie mit dem finden Sie unter put-key-policy. AWS CLI

- Informationen zum Ermitteln einer Schlüssel-ID und des Amazon-Ressourcennamens (ARN) finden Sie unter Die Schlüssel-ID und den ARN finden und im AWS Key Management Service - Benutzerhandbuch.
- Weitere Informationen zu Amazon EC2 Auto Scaling-Rollen finden Sie unter Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling.
- Informationen zur Amazon EBS-Verschlüsselung und KMS im Allgemeinen sowie zur Amazon EBS-Verschlüsselung finden Sie im Amazon EBS-Benutzerhandbuch und im AWS Key Management Service Entwicklerhandbuch.

Beispiel 1: Schlüsselrichtlinienabschnitte, welche Zugriff auf den kundenverwalteten Schlüssel erlauben

Fügen Sie der Schlüsselrichtlinie des kundenverwalteten Schlüssel die folgenden beiden Richtlinienanweisungen hinzu und ersetzen Sie dabei den Beispiel ARN durch den ARN der entsprechenden serviceverknüpften Rolle, der Zugriff auf den Schlüssel gewährt wird. In diesem Beispiel gewähren die Richtlinienabschnitte der mit dem Service verknüpften Rolle mit dem Namen AWSServiceRoleForAutoScalingBerechtigungen zur Verwendung des vom Kunden verwalteten Schlüssels.

```
{
   "Sid": "Allow service-linked role use of the customer managed key",
   "Effect": "Allow",
   "Principal": {
       "AWS": [
           "arn:aws:iam::account-id:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
       ]
   },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
   ],
   "Resource": "*"
}
```

```
{
```

```
    "Sid": "Allow attachment of persistent resources",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::account-id:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
        ]
    },
    "Action": [
        "kms:CreateGrant"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": true
        }
     }
}
```

Beispiel 2: Schlüsselrichtlinienabschnitte, welche Zugriff auf den kundenverwalteten Schlüssel über mehrere Konten erlauben

Wenn Sie einen vom Kunden verwalteten Schlüssel in einem anderen Konto als der Auto-Scaling-Gruppe erstellen, müssen Sie eine Berechtigung in Kombination mit der Schlüsselrichtlinie verwenden, um den kontoübergreifenden Zugriff auf den Schlüssel zu ermöglichen.

Es gibt zwei Schritte, die in der folgenden Reihenfolge ausgeführt werden müssen:

1. Fügen Sie zunächst die folgenden beiden Richtlinienerklärungen zur Schlüsselrichtlinie des vom Kunden verwalteten Schlüssels hinzu. Ersetzen Sie den Beispiel-ARN durch den ARN des anderen Kontos und achten Sie darauf, *111122223333* durch die tatsächliche Konto-ID des Kontos zu ersetzen, in dem Sie AWS-Konto die Auto Scaling Scaling-Gruppe erstellen möchten. Damit können Sie einem IAM-Benutzer oder einer IAM-Rolle im angegebenen Konto die Berechtigung erteilen, mit dem folgenden CLI-Befehl eine Berechtigung für den Schlüssel zu erstellen. Dies gewährt jedoch keinen Benutzern Zugriff auf den Schlüssel.

   ```
   {
       "Sid": "Allow external account 111122223333 use of the customer managed key",
       "Effect": "Allow",
       "Principal": {
           "AWS": [
   ```

```
            "arn:aws:iam::111122223333:root"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

```
{
    "Sid": "Allow attachment of persistent resources in external
 account 111122223333",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::111122223333:root"
        ]
    },
    "Action": [
        "kms:CreateGrant"
    ],
    "Resource": "*"
}
```

2. Erstellen Sie dann von dem Konto aus, in dem Sie die Auto-Scaling-Gruppe erstellen möchten, eine Berechtigung, welche die relevanten Berechtigungen an die entsprechende serviceverknüpfte Rolle delegiert. Das `Grantee Principal`-Element der Berechtigung ist der ARN der dazugehörigen serviceverknüpften Rolle. `key-id` ist der ARN des Schlüssels.

   *Im Folgenden finden Sie ein Beispiel für einen CLI-Befehl create-grant, der der **AWSServiceRoleForAutoScaling**im Konto 111122223333 genannten dienstverknüpften Rolle die Berechtigung erteilt, den vom Kunden verwalteten Schlüssel im Konto 444455556666 zu verwenden.*

```
aws kms create-grant \
  --region us-west-2 \
  --key-id arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d \
```

```
  --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling \
  --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"
```

Damit dieser Befehl erfolgreich ist, muss der Benutzer, der die Anforderung stellt, über Berechtigungen für die `CreateGrant`-Aktion verfügen.

Die folgende IAM-Beispielrichtlinie ermöglicht es einer IAM-Identität (Benutzer oder Rolle) im Konto *111122223333*, eine Berechtigung für den vom Kunden verwalteten Schlüssel im Konto *444455556666* zu erstellen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreationOfGrantForTheKMSKeyinExternalAccount444455556666",
      "Effect": "Allow",
      "Action": "kms:CreateGrant",
      "Resource": "arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    }
  ]
}
```

Weitere Informationen über die Erstellung eines Zuschusses für einen KMS-Schlüssel in einem anderen AWS-Konto, finden Sie unter Berechtigungserteilungen in AWS KMS im AWS Key Management Service -Entwicklerhandbuch.

> **Important**
>
> Der Name der serviceverknüpften Rolle, der als Prinzipal des Empfängers angegeben wird, muss der Name einer vorhandenen Rolle sein. Um nach dem Erstellen der Erteilung sicherzustellen, dass die Erteilung Amazon EC2 Auto Scaling erlaubt, den angegebenen KMS-Schlüssel zu verwenden, löschen Sie die serviceverknüpfte Rolle nicht und erstellen Sie diese nicht neu.

Bearbeiten von Schlüsselrichtlinien in der AWS KMS -Konsole

Die Beispiele in den vorherigen Abschnitten zeigen nur, wie einer Schlüsselrichtlinie Anweisungen hinzugefügt werden, was nur eine Möglichkeit darstellt, eine Schlüsselrichtlinie zu ändern. Die einfachste Möglichkeit, eine Schlüsselrichtlinie zu ändern, besteht darin, die Standardansicht der AWS KMS Konsole für wichtige Richtlinien zu verwenden und eine IAM-Identität (Benutzer oder Rolle) zu einem der Hauptbenutzer für die entsprechende Schlüsselrichtlinie zu machen. Weitere Informationen finden Sie im AWS Key Management Service Entwicklerhandbuch unter Verwenden der AWS Management Console Standardansicht.

> **Important**
>
> Gehen Sie vorsichtig vor. Die Standardansichtsrichtlinien der Konsole beinhalten Berechtigungen zur Ausführung von AWS KMS `Revoke` Vorgängen mit dem vom Kunden verwalteten Schlüssel. Wenn Sie AWS-Konto Zugriff auf einen vom Kunden verwalteten Schlüssel in Ihrem Konto gewähren und Sie versehentlich die Erteilung widerrufen, mit der sie ihnen diese Berechtigung erteilt haben, können externe Benutzer nicht mehr auf ihre verschlüsselten Daten oder den Schlüssel, der zur Verschlüsselung ihrer Daten verwendet wurde, zugreifen.

Identity and Access Management für Amazon EC2 Auto Scaling

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) ist, Amazon EC2 Auto Scaling-Ressourcen zu nutzen. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Um Amazon EC2 Auto Scaling verwenden zu können, benötigen Sie eine AWS-Konto und Ihre Sicherheitsanmeldedaten für die Anmeldung bei Ihrem Konto. Weitere Informationen finden Sie unter AWS Sicherheitsanmeldedaten im IAM-Benutzerhandbuch.

Eine umfassende IAM-Dokumentation finden Sie im IAM User Guide.

Zugriffskontrolle

Auch wenn Sie über gültige Anmeldeinformationen zur Authentifizierung Ihrer Anfragen verfügen, können Sie die Amazon EC2 Auto Scaling-Ressourcen nur mit entsprechenden Berechtigungen erstellen oder darauf zugreifen. Beispielsweise müssen Sie über Berechtigungen zum Erstellen von Auto-Scaling-Gruppen, zum Starten von Instances mit Startvorlagen usw. verfügen.

Dieses Thema enthält Informationen dazu, wie ein IAM-Administrator Ihre Amazon EC2 Auto Scaling-Ressourcen mithilfe von IAM sichern kann, indem er steuert, wer Amazon EC2 Auto Scaling-Aktionen durchführen darf.

Wir empfehlen Ihnen, die Amazon EC2-Themen zuerst zu lesen. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für Amazon EC2 im Amazon EC2-Benutzerhandbuch. Nachdem Sie die Themen in diesem Abschnitt gelesen haben, sollten Sie eine gute Vorstellung davon haben, welche Zugriffsberechtigungen Amazon EC2 bietet, und wie diese zu Ihren Amazon EC2 Auto Scaling-Ressourcenberechtigungen passen können.

Themen

Funktionsweise von Amazon EC2 Auto Scaling mit IAM

Informieren Sie sich zunächst darüber, welche IAM-Features mit Amazon EC2 Auto Scaling verwendet werden können, bevor Sie IAM verwenden, um den Zugriff auf Amazon EC2 Auto Scaling zu verwalten.

IAM-Features, die mit Amazon EC2 Auto Scaling verwendet werden können

IAM-Feature	Amazon EC2 Auto Scaling-Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services pezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Ja

Einen Überblick über das AWS-Services Zusammenwirken von Amazon EC2 Auto Scaling und anderen mit den meisten IAM-Funktionen finden Sie unter , AWS-Services die mit IAM funktionieren im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien für Amazon EC2 Auto Scaling

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.

Ressourcenbasierte Richtlinien in Amazon EC2 Auto Scaling

Unterstützt ressourcenbasierte Richtlinien	Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder umfassen AWS-Services.

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource in unterschiedlichen befinden AWS-Konten, muss ein IAM-Administrator im vertrauenswürdigen Konto auch der Prinzipal-Entität (Benutzer oder Rolle) die Berechtigung für den Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden im IAM-Benutzerhandbuch.

Richtlinienaktionen für Amazon EC2 Auto Scaling

Unterstützt Richtlinienaktionen	Ja

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

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

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der Amazon EC2 Auto Scaling-Aktionen finden Sie in der Service Authorization Reference unter Von Amazon EC2 Auto Scaling definierte Aktionen.

Bei Richtlinienaktionen in Amazon EC2 Auto Scaling wird der Aktion das folgende Präfix vorangestellt:

```
autoscaling
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
  "autoscaling:action1",
  "autoscaling:action2"
]
```

Mithilfe von Platzhaltern (*) können mehrere Aktionen angegeben werden. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "autoscaling:Describe*"
```

Richtlinienressourcen für Amazon EC2 Auto Scaling

Unterstützt Richtlinienressourcen	Ja

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

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

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Sie können ARNs verwenden, um die Auto-Scaling-Gruppen und Startkonfigurationen zu identifizieren, für welche die IAM-Richtlinie gilt.

Eine Auto Scaling-Gruppe hat den folgenden ARN.

```
"Resource": "arn:aws:autoscaling:region:account-
id:autoScalingGroup:uuid:autoScalingGroupName/asg-name"
```

Eine Startkonfiguration hat den folgenden ARN.

```
"Resource": "arn:aws:autoscaling:region:account-
id:launchConfiguration:uuid:launchConfigurationName/lc-name"
```

Um eine Auto-Scaling-Gruppe mit der Aktion `CreateAutoScalingGroup` anzugeben, müssen Sie die UUID durch einen Platzhalter (*) ersetzen, wie im folgenden Beispiel zu sehen.

```
"Resource": "arn:aws:autoscaling:region:account-
id:autoScalingGroup:*:autoScalingGroupName/asg-name"
```

Um eine Startkonfiguration mit der Aktion `CreateLaunchConfiguration` anzugeben, müssen Sie die UUID durch einen Platzhalter (*) ersetzen, wie im folgenden Beispiel zu sehen.

```
"Resource": "arn:aws:autoscaling:region:account-
id:launchConfiguration:*:launchConfigurationName/lc-name"
```

Weitere Informationen zu Amazon EC2 Auto Scaling-Ressourcentypen und zu ihren ARNs finden Sie in der Service Authorization Reference unter Von Amazon EC2 Auto Scaling definierte Ressourcen. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von Amazon EC2 Auto Scaling definierte Aktionen.

> **Note**
>
> Ein Beispiel für eine IAM-Richtlinie, die ARNs zur Steuerung des Zugriffs auf Auto-Scaling-Gruppen verwendet, finden Sie unter Steuern Sie, welche Auto-Scaling-Gruppen gelöscht werden können.

Nicht alle Amazon EC2 Auto Scaling-Aktionen unterstützen Berechtigungen auf Ressourcenebene. Für Aktionen, die Berechtigungen auf Ressourcenebene nicht unterstützen, muss ein Platzhalter (*) als Ressource verwendet werden.

Die folgenden Amazon EC2 Auto Scaling-Aktionen unterstützen keine Berechtigungen auf Ressourcenebene.

- `DescribeAccountLimits`
- `DescribeAdjustmentTypes`
- `DescribeAutoScalingGroups`
- `DescribeAutoScalingInstances`
- `DescribeAutoScalingNotificationTypes`
- `DescribeInstanceRefreshes`
- `DescribeLaunchConfigurations`
- `DescribeLifecycleHooks`
- `DescribeLifecycleHookTypes`
- `DescribeLoadBalancers`
- `DescribeLoadBalancerTargetGroups`

- `DescribeMetricCollectionTypes`
- `DescribeNotificationConfigurations`
- `DescribePolicies`
- `DescribeScalingActivities`
- `DescribeScalingProcessTypes`
- `DescribeScheduledActions`
- `DescribeTags`
- `DescribeTerminationPolicyTypes`
- `DescribeWarmPool`

Richtlinienbedingungsschlüssel für Amazon EC2 Auto Scaling

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja

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

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

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

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

AWS unterstützt globale Bedingungsschlüssel und servicespezifische Bedingungsschlüssel. Informationen zum Anzeigen aller AWS globalen Bedingungsschlüssel finden Sie unter AWS Globale Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

Amazon EC2 Auto Scaling unterstützt die folgenden Bedingungsschlüssel, die verwendet werden können, um den Zugriff auf unterstützte Aktionen zu kontrollieren und die Konfiguration von Auto-Scaling-Gruppen zu erzwingen:

- `autoscaling:InstanceTypes`
- `autoscaling:LaunchConfigurationName`
- `autoscaling:LaunchTemplateVersionSpecified`
- `autoscaling:LoadBalancerNames`
- `autoscaling:MaxSize`
- `autoscaling:MinSize`
- `autoscaling:ResourceTag/key-name: tag-value`
- `autoscaling:TargetGroupARNs`
- `autoscaling:VPCZoneIdentifiers`

Die folgenden Bedingungsschlüssel gelten speziell für die Erstellung von Startkonfigurationsanforderungen:

- `autoscaling:ImageId`
- `autoscaling:InstanceType`
- `autoscaling:MetadataHttpEndpoint`
- `autoscaling:MetadataHttpPutResponseHopLimit`
- `autoscaling:MetadataHttpTokens`
- `autoscaling:SpotPrice`

Amazon EC2 Auto Scaling unterstützt auch die folgenden globalen Bedingungsschlüssel, mit denen Sie Berechtigungen basierend auf den Tags in der Anforderung definieren können oder die in der Auto Scaling-Gruppe vorhanden sind. Weitere Informationen finden Sie unter Tagging von Auto-Scaling-Gruppen und Instances.

- `aws:RequestTag/key-name: tag-value`

- `aws:ResourceTag/key-name: tag-value`
- `aws:TagKeys: [tag-key, ...]`

Eine Liste der Amazon EC2 Auto Scaling-Aktionen finden Sie in der Service Authorization Reference unter Von Amazon EC2 Auto Scaling definierte Aktionen. Weitere Informationen zu Amazon EC2 Auto Scaling-Bedingungsschlüsseln finden Sie unter Bedingungsschlüssel für Amazon EC2 Auto Scaling.

> **Note**
>
> Beispiele für IAM-Richtlinien, die Bedingungsschlüssel verwenden, um den Zugriff auf unterstützte Aktionen zu kontrollieren und die Konfiguration von Auto-Scaling-Gruppen zu erzwingen, finden Sie in den folgenden Ressourcen:
>
> - Eine Startvorlage und eine Versionsnummer verlangen – In diesem Beispiel wird erzwungen, dass beim Erstellen oder Aktualisieren von Auto Scaling-Gruppen eine Startvorlage und die Versionsnummer der Startvorlage angegeben werden müssen.
> - Steuern Sie die Größe der Auto-Scaling-Gruppen, die erstellt werden können – In diesem Beispiel werden Einschränkungen für die möglichen Werte für die `MaxSize` Eigenschaften `MinSize` und beim Erstellen oder Aktualisieren von Auto Scaling-Gruppen mit einem bestimmten Tag erzwungen.
> - Steuern, welche Skalierungsrichtlinien gelöscht werden können – In diesem Beispiel wird erzwungen, dass das Löschen von Skalierungsrichtlinien nur für Auto Scaling-Gruppen ohne ein bestimmtes Tag zulässig ist.

ACLs in Amazon EC2 Auto Scaling

Unterstützt ACLs	Nein

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

ABAC mit Amazon EC2 Auto Scaling

Unterstützt ABAC (Tags in Richtlinien)	Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In werden AWS diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anfügen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

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

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Was ist ABAC? im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

ABAC ist für Ressourcen möglich, die Tags unterstützen. Tags werden jedoch nicht von allen Ressourcen unterstützt. Startkonfigurationen und Skalierungsrichtlinien unterstützen keine Tags, Auto–Scaling-Gruppen dagegen schon.

Weitere Informationen finden Sie unter Tagging von Auto-Scaling-Gruppen und Instances.

Verwenden temporärer Anmeldeinformationen mit Amazon EC2 Auto Scaling

Unterstützt temporäre Anmeldeinformationen	Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich der , die mit temporären Anmeldeinformationen

AWS-Services funktionieren, finden Sie unter AWS-Services , die mit IAM funktionieren im IAM-Benutzerhandbuch.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich AWS Management Console mit einer anderen Methode als einem Benutzernamen und einem Passwort bei der anmelden. Wenn Sie beispielsweise AWS über den SSO-Link (Single Sign-On) Ihres Unternehmens auf zugreifen, erstellt dieser Prozess automatisch temporäre Anmeldeinformationen. 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.

Sie können temporäre Anmeldeinformationen manuell mit der AWS CLI oder der AWS API erstellen. Sie können diese temporären Anmeldeinformationen dann verwenden, um auf zuzugreifen AWS. AWS empfohlen, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Servicerollen für Amazon EC2 Auto Scaling

Unterstützt Servicerollen	Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

Wenn Sie einen Lebenszyklus-Hook erstellen, der ein Amazon SNS-Thema oder eine Amazon SQS-Warteschlange benachrichtigt, müssen Sie eine Rolle angeben, damit Amazon EC2 Auto Scaling in Ihrem Namen auf Amazon SNS oder Amazon SQS zugreifen kann. Verwenden Sie die IAM-Konsole, um die Servicerolle für Ihren Lebenszyklus-Hook einzurichten. Die Konsole unterstützt Sie bei der Erstellung einer Rolle mit ausreichenden Berechtigungen mit einer verwalteten Richtlinie. Weitere Informationen finden Sie unter Benachrichtigungen über Amazon SNS erhalten und Benachrichtigungen über Amazon SQS erhalten.

Wenn Sie eine Auto Scaling-Gruppe erstellen, können Sie optional eine Servicerolle übergeben, damit Amazon EC2-Instances in AWS-Services Ihrem Namen auf andere zugreifen können. Die Servicerolle für Amazon EC2-Instances (auch Amazon EC2-Instance-Profil für eine Startvorlage oder

Startkonfiguration genannt) ist eine spezielle Art von Servicerolle, die jeder EC2-Instance in einer Auto-Scaling-Gruppe zugewiesen wird, wenn die Instance startet. Sie können die IAM-Konsole und verwenden AWS CLI , um diese Servicerolle zu erstellen oder zu bearbeiten. Weitere Informationen finden Sie unter IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden.

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle kann dazu führen, dass Amazon EC2 Auto Scaling nicht mehr funktioniert. Bearbeiten Sie Servicerollen nur, wenn Sie von Amazon EC2 Auto Scaling dazu angeleitet werden.

Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling

Unterstützt serviceverknüpfte Rollen	Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem angezeigt AWS-Konto und gehören dem Service. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Amazon EC2 Auto Scaling-Rollen finden Sie unter Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling.

API-Berechtigungen für Amazon EC2 Auto Scaling

Sie müssen den Benutzern die Berechtigung erteilen, die benötigten Amazon EC2 Auto Scaling-API-Aktionen aufzurufen, wie in Richtlinienaktionen für Amazon EC2 Auto Scaling beschrieben. Darüber hinaus müssen Sie Benutzern für einige Amazon EC2 Auto Scaling Scaling-Aktionen die Erlaubnis erteilen, bestimmte Aktionen von anderen AWS APIs aus aufzurufen.

Erforderliche Berechtigungen von anderen AWS APIs

Zusätzlich zu den Amazon EC2 Auto Scaling Scaling-API-Berechtigungen müssen Benutzer über die folgenden Berechtigungen von anderen AWS APIs verfügen, um die zugehörige Aktion erfolgreich ausführen zu können.

Erstellen einer Auto-Scaling-Gruppe (`autoscaling:CreateAutoScalingGroup`)

- `iam:CreateServiceLinkedRole`— Um die standardmäßige serviceverknüpfte Rolle zu erstellen, falls diese Rolle noch nicht existiert.
- `iam:PassRole`— Um beim Start eine IAM-Rolle an den Service oder an EC2-Instances zu übergeben. Wird benötigt, wenn eine nicht standardmäßige serviceverknüpfte Rolle, eine IAM-Rolle für einen Lebenszyklus-Hook oder eine Startvorlage, die ein Instance-Profil spezifiziert (ein Container für eine IAM-Rolle), bereitgestellt wird.
- `ec2:RunInstances`— Um Instances zu starten, wenn eine Startvorlage bereitgestellt wird.
- `ec2:CreateTags`— Um Instances und Volumes beim Start zu taggen, wenn eine Startvorlage mit einer Tag-Spezifikation bereitgestellt wird.

Erstellen eines Lebenszyklus-Hooks (`autoscaling:PutLifecycleHook`)

- `iam:PassRole`— Um eine IAM-Rolle an den Service zu übergeben. Wird benötigt, wenn eine IAM-Rolle bereitgestellt wird.

Fügen Sie eine VPC-Lattice-Zielgruppe hinzu (`autoscaling:AttachTrafficSources`)

- `vpc-lattice:RegisterTargets`— Um Instanzen automatisch bei der Zielgruppe zu registrieren.

Eine VPC-Lattice-Zielgruppe abtrennen (`autoscaling:DetachTrafficSources`)

- `vpc-lattice:DeregisterTargets`— Um Instanzen automatisch bei der Zielgruppe abzumelden.

Erstellen einer Startkonfiguration (`autoscaling:CreateLaunchConfiguration`)

- `ec2:DescribeImages`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceAttribute`
- `ec2:DescribeKeyPairs`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSpotInstanceRequests`
- `ec2:DescribeVpcClassicLink`
- `iam:PassRole`— Um beim Start eine IAM-Rolle an EC2-Instances zu übergeben. Erforderlich, wenn eine Startkonfiguration ein Instance-Profil (einen Container für eine IAM-Rolle) angibt.

AWS verwaltete Richtlinien für Amazon EC2 Auto Scaling

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet AWS wird. AWS Verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie 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.

Von Amazon EC2 Auto Scaling verwaltete Richtlinien

Sie können die folgenden verwalteten Richtlinien an Ihre AWS Identity and Access Management (IAM-) Identitäten (Benutzer oder Rollen) anhängen. Jede Richtlinie gewährt Zugriff auf alle oder einige der API-Aktionen für Amazon EC2 Auto Scaling.

- AutoScalingConsoleFullZugriff — Gewährt vollen Zugriff auf Amazon EC2 Auto Scaling mithilfe von. AWS Management Console Diese Richtlinie funktioniert, wenn Sie Startkonfigurationen verwenden, aber nicht, wenn Sie Startvorlagen verwenden.
- AutoScalingConsoleReadOnlyAccess— Gewährt schreibgeschützten Zugriff auf Amazon EC2 Auto Scaling mithilfe von. AWS Management Console Diese Richtlinie funktioniert, wenn Sie Startkonfigurationen verwenden, aber nicht, wenn Sie Startvorlagen verwenden.
- AutoScalingFullAccess— Gewährt vollen Zugriff auf Amazon EC2 Auto Scaling für IAM-Identitäten, die vollen Amazon EC2 Auto Scaling Scaling-Zugriff über die SDKs benötigen, AWS CLI aber keinen Zugriff. AWS Management Console
- AutoScalingReadOnlyZugriff — Gewährt schreibgeschützten Zugriff auf Amazon EC2 Auto Scaling für IAM-Identitäten, die nur Aufrufe an die SDKs oder tätigen. AWS CLI

Bei der Verwendung von Startvorlagen über die Konsole müssen Sie zusätzliche, für Startvorlagen spezifische Berechtigungen erteilen, die unter Support für Startvorlagen erläutert werden. Die Amazon EC2 Auto Scaling-Konsole benötigt Berechtigungen für `ec2`-Aktionen, damit Informationen über Startvorlagen angezeigt und Instances mithilfe von Startvorlagen gestartet werden können.

AutoScalingServiceRole AWS Von Richtlinien verwaltete Richtlinie

Diese Richtlinie ist mit einer servicebezogenen Rolle verknüpft, die es Amazon EC2 Auto Scaling ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling.

Die Berechtigungen für diese Richtlinie finden Sie unter AutoScalingServiceRoleRichtlinie in der Referenz für AWS verwaltete Richtlinien.

Amazon EC2 Auto Scaling Scaling-Updates für AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amazon EC2 Auto Scaling an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen. Um automatische Warnungen über Änderungen an dieser Seite zu erhalten, abonnieren Sie den RSS-Feed auf der Amazon-EC2-Auto-Scaling Seite „Document history“ (Dokumentverlauf).

Änderung	Beschreibung	Datum
Amazon EC2 Auto Scaling fügt seiner service-verknüpften Rolle Berechtigungen zu	Die `AutoScalingServiceRolePolicy` Richtlinie gewährt nun Berechtigungen zum Aufrufen der Amazon EC2 GetSecurityGroupsForVPC-API-Aktion, um alle Sicherheitsgruppen für eine VPC abzurufen, um die Validierung zu verbessern, und der Amazon EC2 GetInstanceTypesFromInstanceRequirementsEC2-API-Aktion, um Informationen darüber abzurufen, welche Instance-Typen bestimmte	29. Februar 2024

Änderung	Beschreibung	Datum
	Instance-Anforderungen erfüllen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling.	

Änderung	Beschreibung	Datum
Amazon EC2 Auto Scaling fügt seiner service-verknüpften Rolle Berechtigungen zu	Die Richtlinie `AutoScalingServiceRolePolicy` gewährt dem Service nun Berechtigungen für den Zugriff auf die API-Aktionen, die er für eine Integration mit VPC Lattice benötigt. • `GetTargetGroup` - und `ListTargetGroup` - Aktionen. Erforderlich zum Abrufen von Informationen zu den VPC-Lattice-Zielgruppen. • `RegisterTargets` - und `DeregisterTargets` - Aktionen. Erforderlich zum Registrieren und Deregistrieren von Instances bei VPC-Lattice-Zielgruppen. • `ListTargets` . Ermöglicht Amazon EC2 Auto Scaling das Abrufen von Zustandsinformationen für Instances, die bei VPC-Lattice-Zielgruppen registriert sind. Weitere Informationen finden Sie unter Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling.	6. Dezember 2022

Änderung	Beschreibung	Datum
Amazon EC2 Auto Scaling fügt seiner service-verknüpften Rolle Berechtigungen zu	Um die Verwendung eines AWS Systems Manager Parameters als Alias für eine AMI-ID beim Erstellen einer Startvorlage zu unterstützen, gewährt die `AutoScalingServiceRolePolicy` Richtlinie jetzt die Erlaubnis, die AWS Systems Manager GetParametersAPI-Aktion aufzurufen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling.	28. März 2022
Amazon EC2 Auto Scaling fügt seiner service-verknüpften Rolle Berechtigungen zu	Um die prädiktive Skalierung zu unterstützen, beinhaltet die `AutoScalingServiceRolePolicy` Richtlinie jetzt die Erlaubnis, die CloudWatch GetMetricDaten-API-Aktion aufzurufen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling.	19. Mai 2021
Amazon EC2 Auto Scaling startet Nachverfolgung von Änderungen	Amazon EC2 Auto Scaling begann, Änderungen an seinen AWS verwalteten Richtlinien nachzuverfolgen.	19. Mai 2021

Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling nutzt serviceverknüpfte Rollen für die Berechtigungen, die für den Aufruf anderer AWS-Services -Services in Ihrem Namen benötigt werden. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit einer verknüpft ist. AWS-Service

Serviceverknüpfte Rollen bieten eine sichere Möglichkeit, um Berechtigungen zu AWS-Services -Services zu delegieren, da nur der verknüpfte Service eine serviceverknüpfte Rolle annehmen kann. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen im -IAM-Benutzerhandbuch. Servicebezogene Rollen ermöglichen außerdem, dass alle API-Aufrufe sichtbar sind. AWS CloudTrail Das hilft bei Überwachungs- und Prüfungsanforderungen, da Sie alle in Ihrem Namen von Amazon EC2 Auto Scaling ausgeführten Aktionen nachverfolgen können. Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling API-Aufrufe protokollieren mit AWS CloudTrail.

In den folgenden Abschnitten wird die Erstellung und Verwaltung von serviceverknüpften Amazon EC2 Auto Scaling-Rollen beschrieben. Beginnen Sie mit dem Konfigurieren von Berechtigungen, damit eine IAM-Identität (z. B. ein Benutzer oder eine Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen im -IAM-Benutzerhandbuch.

Inhalt

Übersicht

Es gibt zwei Typen von serviceverknüpften Amazon EC2 Auto Scaling-Rollen:

- Die standardmäßige dienstbezogene Rolle für Ihr Konto, benannt. AWSServiceRoleForAutoScaling Diese Rolle wird automatisch Ihren Auto Scaling-Gruppen zugewiesen, es sei denn, Sie geben eine andere serviceverknüpfte Rolle an.

- ***Eine dienstbezogene Rolle mit einem benutzerdefinierten Suffix, das Sie bei der Erstellung der Rolle angeben, AWSServiceRoleForAutoScaling z. B. _ mysuffix.***

Eine serviceverknüpfte Rolle mit benutzerdefiniertem Suffix hat dieselben Berechtigungen wie die standardmäßige serviceverknüpfte Rolle. In beiden Fällen können Sie die Rollen nicht bearbeiten und auch nicht löschen, wenn sie noch von einer Auto Scaling-Gruppe verwendet werden. Der einzige Unterschied ist das Suffix des Rollennamens.

Sie können beide Rollen angeben, wenn Sie Ihre AWS Key Management Service Schlüsselrichtlinien bearbeiten, sodass Instances, die von Amazon EC2 Auto Scaling gestartet werden, mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt werden können. Wenn Sie jedoch vorhaben, einem bestimmten kundenverwalteten Schlüssel individuell Zugriff zu gewähren, sollten Sie eine serviceverknüpfte Rolle mit benutzerdefiniertem Suffix verwenden. Eine serviceverknüpfte Rolle mit benutzerdefiniertem Suffix bietet Ihnen:

- Mehr Kontrolle über den kundenverwalteten Schlüssel
- Die Möglichkeit, in Ihren CloudTrail Protokollen nachzuverfolgen, welche Auto Scaling Scaling-Gruppe einen API-Aufruf getätigt hat

Wenn Sie kundenverwaltete Schlüssel erstellen, auf die nicht alle Benutzer Zugriff haben sollen, führen Sie diese Schritte aus, um die Verwendung einer serviceverknüpften Rolle mit benutzerdefiniertem Suffix zuzulassen:

1. Erstellen Sie eine serviceverknüpfte Rolle mit einem benutzerdefinierten Suffix. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle (manuell).
2. Erteilen Sie der serviceverknüpften Rolle Zugriff auf einen kundenverwalteten Schlüssel. Weitere Informationen über die Schlüsselrichtlinie, die zulässt, dass der Schlüssel von einer serviceverknüpften Rolle verwendet wird, finden Sie unter Erforderliche AWS KMS Schlüsselrichtlinie für die Verwendung mit verschlüsselten Volumes.
3. Geben Sie Benutzern Zugriff auf die von Ihnen erstellte serviceverknüpfte Rolle. Weitere Informationen zum Erstellen der IAM-Richtlinie finden Sie unter Steuern Sie, welche serviceverknüpfte Rolle übergeben werden kann (mit) PassRole. Wenn Benutzer versuchen, eine serviceverknüpfte Rolle ohne Berechtigung anzugeben, diese Rolle an den Service weiterzugeben, wird eine Fehlermeldung angezeigt.

Von der serviceverknüpften Rolle erteilte Berechtigungen

Amazon EC2 Auto Scaling verwendet die benannte serviceverknüpfte Rolle AWSServiceRoleForAutoScalingoder Ihr benutzerdefiniertes Suffix für serviceverknüpfte Rolle.

Die serviceverknüpfte Rolle vertraut darauf, dass der folgende Service die Rolle annimmt:

- `autoscaling.amazonaws.com`

Die Rollenberechtigungsrichtlinie, AutoScalingServiceRolePolicy, ermöglicht Amazon EC2 Auto Scaling, die folgenden Aktionen durchzuführen:

- `ec2`— EC2-Instances erstellen, beschreiben, ändern, starten/stoppen und beenden.
- `iam`— Übergeben Sie IAM-Rollen an EC2-Instances, sodass Anwendungen, die auf den Instances ausgeführt werden, auf temporäre Anmeldeinformationen für die Rolle zugreifen können.
- `iam`— Erstellen Sie die AWSServiceRoleForEC2Spotserviceverknüpfte Rolle, damit Amazon EC2 Auto Scaling Spot-Instances in Ihrem Namen starten kann.
- `elasticloadbalancing`— Registrieren und deregistrieren Sie Instances mit Elastic Load Balancing und überprüfen Sie den Zustand registrierter Ziele.
- `cloudwatch`— CloudWatch Alarme für Skalierungsrichtlinien erstellen, beschreiben, ändern und löschen und Metriken abrufen, die für die prädiktive Skalierung verwendet werden.
- `sns`— Veröffentlichen Sie Benachrichtigungen auf Amazon SNS, wenn Instances gestartet oder beendet werden.
- `events`— EventBridge Regeln in Ihrem Namen erstellen, beschreiben, aktualisieren und löschen.
- `ssm`— Liest Parameter aus dem Parameterspeicher, wenn Sie einen Systems Manager Manager-Parameter als Alias für eine AMI-ID in einer Startvorlage verwenden.
- `vpc-lattice`— Registrieren und deregistrieren Sie Instances bei VPC Lattice und überprüfen Sie den Zustand der registrierten Ziele.

Erstellen einer serviceverknüpften Rolle (automatisch)

Amazon EC2 Auto Scaling erstellt die AWSServiceRoleForAutoScalingserviceverknüpfte Rolle für Sie, wenn Sie zum ersten Mal eine Auto Scaling Scaling-Gruppe erstellen, es sei denn, Sie erstellen manuell eine serviceverknüpfte Rolle mit benutzerdefiniertem Suffix und geben sie bei der Erstellung der Gruppe an.

> **Important**
>
> Sie müssen über IAM-Berechtigungen zum Erstellen der serviceverknüpften Rolle verfügen. Andernfalls schlägt das automatische Erstellen fehl. Weitere Informationen finden Sie unter Berechtigungen von serviceverknüpften Rollen im IAM-Benutzerhandbuch und unter Erstellen einer serviceverknüpften Rolle in diesem Handbuch.

Amazon EC2 Auto Scaling unterstützt serviceverknüpfte Rollen seit März 2018. Wenn Sie zuvor eine Auto Scaling-Gruppe erstellt haben, hat Amazon EC2 Auto Scaling die AWSServiceRoleForAutoScalingRolle in Ihrem Konto erstellt. Weitere Informationen finden Sie unter In meinem AWS-Konto wird eine neue Rolle angezeigt im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle (manuell)

So erstellen Sie eine serviceverknüpfte Rolle (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und Create Role (Rolle erstellen) aus.
3. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die Option AWS - Dienst.
4. Wählen Sie bei Choose the service that will use this role (Service auswählen, der diese Rolle verwendet) die Option EC2 Auto Scaling und den Anwendungsfall EC2 Auto Scaling aus.
5. Wählen Sie Next: Permissions (Nächster Schritt: Berechtigungen), Next: Tags (Nächster Schritt: Tags) und dann Next: Review (Nächster Schritt: Prüfen) aus. Hinweis: Während der Erstellung können keine Tags an serviceverknüpfte Rollen angefügt werden.
6. ***Lassen Sie auf der Seite „Überprüfen“ das Feld Rollenname leer, um eine servicebezogene Rolle mit dem Namen zu erstellen AWSServiceRoleForAutoScaling, oder geben Sie ein Suffix ein, um eine dienstbezogene Rolle mit dem Suffix name _ zu erstellen. AWSServiceRoleForAutoScaling***
7. (Optional:) Bearbeiten Sie in Role description (Rollenbeschreibung) die Beschreibung für die neue serviceverknüpfte Rolle.
8. Wählen Sie Rolle erstellen aus.

So erstellen Sie eine serviceverknüpfte Rolle (AWS CLI)

Verwenden Sie den folgenden CLI-Befehl create-service-linked-role, um eine serviceverknüpfte Rolle für Amazon EC2 Auto Scaling mit dem Suffix Name _ zu erstellen. AWSServiceRoleForAutoScaling

```
aws iam create-service-linked-role --aws-service-name autoscaling.amazonaws.com --
custom-suffix suffix
```

Die Ausgabe dieses Befehls umfasst den ARN der serviceverknüpften Rolle, den Sie verwenden können, um der serviceverknüpften Rolle Zugriff auf Ihren vom Kunden verwalteten Schlüssel zu erteilen.

```
{
    "Role": {
        "RoleId": "ABCDEF0123456789ABCDEF",
        "CreateDate": "2018-08-30T21:59:18Z",
        "RoleName": "AWSServiceRoleForAutoScaling_suffix",
        "Arn": "arn:aws:iam::123456789012:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling_suffix",
        "Path": "/aws-service-role/autoscaling.amazonaws.com/",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "sts:AssumeRole"
                    ],
                    "Principal": {
                        "Service": [
                            "autoscaling.amazonaws.com"
                        ]
                    },
                    "Effect": "Allow"
                }
            ]
        }
    }
}
```

Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Bearbeiten der serviceverknüpften Rolle

Die für Amazon EC2 Auto Scaling erstellten serviceverknüpften Rollen können nicht bearbeitet werden. Nach dem Erstellen einer serviceverknüpften Rolle können Sie weder den Namen der Rolle noch ihre Berechtigungen ändern. Sie können jedoch die Beschreibung der Rolle bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen der serviceverknüpften Rolle

Wenn Sie eine Auto Scaling-Gruppe nicht verwenden, empfehlen wir, deren serviceverknüpfte Rolle zu löschen. Das Löschen der Rolle verhindert, dass Sie eine Entität haben, die nicht verwendet oder aktiv überwacht und verwaltet wird.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem die zugehörigen abhängigen Ressourcen gelöscht wurden. Dies schützt Sie vor der versehentlichen Aufkündigung von Amazon EC2 Auto Scaling-Berechtigungen für Ihre Ressourcen. Wenn eine serviceverknüpfte Rolle mit mehreren Auto Scaling-Gruppen verwendet wird, müssen Sie zunächst alle Auto Scaling-Gruppen, welche die serviceverknüpfte Rolle verwenden, löschen, bevor Sie sie löschen können. Weitere Informationen finden Sie unter Löschen der Auto-Scaling-Infrastruktur.

Sie können IAM zum Löschen der serviceverknüpften Rolle verwenden. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Wenn Sie die AWSServiceRoleForAutoScalingserviceverknüpfte Rolle löschen, erstellt Amazon EC2 Auto Scaling die Rolle erneut, wenn Sie eine Auto Scaling Scaling-Gruppe erstellen und keine andere serviceverknüpfte Rolle angeben.

Unterstützte Regionen für Amazon EC2 Auto Scaling serviceverknüpfte Rollen

Amazon EC2 Auto Scaling unterstützt die Verwendung von serviceverknüpften Rollen überall AWS-Regionen dort, wo der Service verfügbar ist.

Beispiele für identitätsbasierte Amazon EC2 Auto Scaling-Richtlinien

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 Amazon EC2 Auto Scaling gewähren.

Informationen dazu, wie Sie unter Verwendung dieser Beispiel-JSON-Richtliniendokumente eine IAM-Richtlinie erstellen, finden Sie unter Erstellen von Richtlinien auf der JSON-Registerkarte im IAM-Benutzerhandbuch.

Dies ist ein Beispiel für eine Berechtigungsrichtlinie.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
          "autoscaling:CreateAutoScalingGroup",
          "autoscaling:UpdateAutoScalingGroup",
          "autoscaling:DeleteAutoScalingGroup"
      ],
      "Resource": "*",
      "Condition": {
          "StringEquals": { "autoscaling:ResourceTag/purpose": "testing" }
      }
   },
   {
      "Effect": "Allow",
      "Action": "autoscaling:Describe*",
      "Resource": "*"
   }]
}
```

Diese Beispielrichtlinie gewährt Benutzern Berechtigungen zum Erstellen, Aktualisieren und Löschen von Auto-Scaling-Gruppen, jedoch nur, wenn die Gruppe das Tag **`purpose=testing`** verwendet. Da `Describe`-Aktionen keine Berechtigungen auf Ressourcenebene unterstützen, müssen Sie sie in einer separaten Anweisung ohne Bedingungen angeben. Um Instances mit einer Startvorlage zu starten, muss der Benutzer auch über die `ec2:RunInstances`-Berechtigung verfügen. Weitere Informationen finden Sie unter Support für Startvorlagen.

> **Note**
>
> Sie können auch eigene benutzerdefinierte IAM-Richtlinien erstellen, um IAM-Identitäten (Benutzern oder Rollen) Berechtigungen zum Ausführen von Amazon-EC2-Auto-Scaling-Aktionen zu gewähren oder zu verweigern. Die benutzerdefinierten Richtlinien können Sie dann den IAM-Identitäten zuweisen, die die angegebenen Berechtigungen fordern. Die folgenden Beispiele zeigen Berechtigungen für einige häufige Anwendungsfälle.

Bei einigen Amazon EC2 Auto Scaling-API-Aktionen lassen sich bestimmte Auto Scaling-Gruppen, die mit der Aktion erstellt oder geändert werden können, in die Richtlinie einbinden. Sie können die Zielressourcen für diese Aktionen einschränken, indem Sie einzelne Auto Scaling-Gruppen-ARNs angeben. Als bewährte Methode empfehlen wir jedoch, tagbasierte Richtlinien zu verwenden, die Aktionen für Auto Scaling-Gruppen mit einem bestimmten Tag zulassen (oder ablehnen).

Beispiele

- Steuern Sie die Größe der Auto-Scaling-Gruppen, die erstellt werden können
- Steuern, welche Tag-Schlüssel und Tag-Werte verwendet werden können
- Steuern Sie, welche Auto-Scaling-Gruppen gelöscht werden können
- Steuern, welche Skalierungsrichtlinien gelöscht werden können
- Steuern Sie den Zugriff auf Aktionen zur Instance-Aktualisierung
- Erstellen einer serviceverknüpften Rolle
- Steuern Sie, welche serviceverknüpfte Rolle übergeben werden kann (mit) PassRole

Steuern Sie die Größe der Auto-Scaling-Gruppen, die erstellt werden können

Die folgende Richtlinie gewährt Berechtigungen zum Erstellen und Aktualisieren aller Auto-Scaling-Gruppen mit dem Tag **`environment=development`**, sofern der Anforderer nicht eine Mindestgröße kleiner als **`1`** oder eine maximale Größe größer als **`10`** angibt. Verwenden Sie nach Möglichkeit Tags, um den Zugriff auf die Auto-Scaling-Gruppen in Ihrem Konto zu steuern.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
          "autoscaling:CreateAutoScalingGroup",
          "autoscaling:UpdateAutoScalingGroup"
      ],
      "Resource": "*",
      "Condition": {
          "StringEquals": { "autoscaling:ResourceTag/environment": "development" },
          "NumericGreaterThanEqualsIfExists": { "autoscaling:MinSize": 1 },
          "NumericLessThanEqualsIfExists": { "autoscaling:MaxSize": 10 }
```

```
      }
   }]
}
```

Wenn Sie den Zugriff auf Auto-Scaling-Gruppen nicht über Tags kontrollieren, können Sie alternativ ARNs verwenden, um die Auto-Scaling-Gruppen zu identifizieren, für welche die IAM-Richtlinie gilt.

Eine Auto Scaling-Gruppe hat den folgenden ARN.

```
"Resource": "arn:aws:autoscaling:region:account-
id:autoScalingGroup:*:autoScalingGroupName/my-asg"
```

Sie können auch mehrere ARNs angeben, indem Sie sie in eine Liste einschließen. Weitere Informationen zum Angeben der ARNs von Amazon EC2 Auto-Scaling-Ressourcen im `Resource`-Element finden Sie unter Richtlinienressourcen für Amazon EC2 Auto Scaling.

Steuern, welche Tag-Schlüssel und Tag-Werte verwendet werden können

Sie können auch Bedingungen in Ihren IAM-Richtlinien verwenden, um die Tag-Schlüssel und Tag-Werte zu steuern, die auf Auto-Scaling-Gruppen angewendet werden können. Verwenden Sie den `aws:RequestTag`-Bedingungsschlüssel, um Berechtigungen zum Erstellen oder Markieren einer Auto-Scaling-Gruppe nur dann zu gewähren, wenn der Anforderer bestimmte Tags angibt. Um nur bestimmte Tag-Schlüssel zuzulassen, verwenden Sie den Bedingungsschlüssel `aws:TagKeys` mit dem Modifikator `ForAllValues`.

Die folgende Richtlinie erfordert, dass der Anforderer in der Anfrage ein Tag mit dem Schlüssel **`environment`** angibt. Der Wert **`"?*"`** erzwingt, dass ein Wert für den Tag-Schlüssel vorhanden ist. Bei der Verwenden eines Platzhalters müssen Sie den `StringLike`-Bedingungsoperator verwenden.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
          "autoscaling:CreateAutoScalingGroup",
          "autoscaling:CreateOrUpdateTags"
      ],
      "Resource": "*",
      "Condition": {
```

```
            "StringLike": { "aws:RequestTag/environment": "?*" }
        }
    }]
}
```

Die folgende Richtlinie legt fest, dass der Antragssteller nur Auto-Scaling-Gruppen mit den Tags **purpose=webserver** und **cost-center=cc123** kennzeichnen kann und nur die Tags **purpose** und **cost-center** zulässt (keine anderen Tags können angegeben werden).

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "autoscaling:CreateAutoScalingGroup",
            "autoscaling:CreateOrUpdateTags"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/purpose": "webserver",
                "aws:RequestTag/cost-center": "cc123"
            },
            "ForAllValues:StringEquals": { "aws:TagKeys": ["purpose", "cost-center"] }
        }
    }]
}
```

Die folgende Richtlinie erfordert, dass der Anforderer mindestens ein Tag in der Anfrage angibt, und lässt nur die **cost-center**- und **owner**-Schlüssel zu.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "autoscaling:CreateAutoScalingGroup",
            "autoscaling:CreateOrUpdateTags"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": { "aws:TagKeys": ["cost-center", "owner"] }
```

```
      }
   }]
}
```

> **Note**
>
> Bei Bedingungen gilt, dass die Groß- und Kleinschreibung für den Bedingungsschlüssel nicht berücksichtigt und für den Bedingungswert beachtet wird. Verwenden Sie aus diesem Grund den `aws:TagKeys`-Bedingungsschlüssel und geben Sie den Tag (Markierung)-Schlüssel als Wert dieser Bedingung an, wenn Sie die Berücksichtigung der Groß- und Kleinschreibung für einen Tag (Markierung)-Schlüssel erzwingen möchten.

Steuern Sie, welche Auto-Scaling-Gruppen gelöscht werden können

Die folgende Richtlinie erlaubt das Löschen einer Auto-Scaling-Gruppe nur, wenn die Gruppe über das Tag verfügt**`environment=development`**.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "autoscaling:DeleteAutoScalingGroup",
      "Resource": "*",
      "Condition": {
          "StringEquals": { "aws:ResourceTag/environment": "development" }
      }
   }]
}
```

Wenn Sie keine Bedingungsschlüssel verwenden, um den Zugriff auf Auto-Scaling-Gruppen zu steuern, können Sie stattdessen die ARNs der Ressourcen in dem `Resource`-Element angeben, um den Zugriff zu kontrollieren.

Die folgende Richtlinie gibt Benutzern die Erlaubnis, die `DeleteAutoScalingGroup` API-Aktion zu verwenden, jedoch nur für Auto-Scaling-Gruppen, deren Name mit **`devteam-`** beginnt.

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

```
      "Effect": "Allow",
      "Action": "autoscaling:DeleteAutoScalingGroup",
      "Resource": "arn:aws:autoscaling:region:account-
id:autoScalingGroup:*:autoScalingGroupName/devteam-*"
   }]
}
```

Sie können auch mehrere ARNs angeben, indem Sie sie in eine Liste einschließen. Einbeziehung der UUID stellt sicher, dass Zugriff zu der spezifischen Auto Scaling-Gruppe gewährt ist. Die UUID für eine neue Gruppe unterscheidet sich von der UUID für eine gelöschte Gruppe mit demselben Namen.

```
"Resource": [
    "arn:aws:autoscaling:region:account-
id:autoScalingGroup:uuid:autoScalingGroupName/devteam-1",
    "arn:aws:autoscaling:region:account-
id:autoScalingGroup:uuid:autoScalingGroupName/devteam-2",
    "arn:aws:autoscaling:region:account-
id:autoScalingGroup:uuid:autoScalingGroupName/devteam-3"
]
```

Steuern, welche Skalierungsrichtlinien gelöscht werden können

Die folgende Richtlinie gewährt Berechtigungen zum Verwenden der `DeletePolicy`-Aktion zum Löschen einer Skalierungsrichtlinie. Es lehnt die Aktion jedoch auch dann ab, wenn die Auto-Scaling-Gruppe, auf welche die Aktion abzielt, über das Tag **`environment=production`** verfügt. Verwenden Sie nach Möglichkeit Tags, um den Zugriff auf die Auto-Scaling-Gruppen in Ihrem Konto zu steuern.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "autoscaling:DeletePolicy",
      "Resource": "*"
   },
   {
      "Effect": "Deny",
      "Action": "autoscaling:DeletePolicy",
      "Resource": "*",
      "Condition": {
          "StringEquals": { "autoscaling:ResourceTag/environment": "production" }
```

```
      }
   }]
}
```

Steuern Sie den Zugriff auf Aktionen zur Instance-Aktualisierung

Die folgende Richtlinie erteilt nur dann Berechtigungen zum Starten, Zurücksetzen und Abbrechen einer Instance-Aktualisierung, wenn die Auto-Scaling-Gruppe, auf welche die Aktion abzielt, über das Tag **`environment=testing`** verfügt. Da `Describe`-Aktionen keine Berechtigungen auf Ressourcenebene unterstützen, müssen Sie sie in einer separaten Anweisung ohne Bedingungen angeben.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
          "autoscaling:StartInstanceRefresh",
          "autoscaling:CancelInstanceRefresh",
          "autoscaling:RollbackInstanceRefresh"
      ],
      "Resource": "*",
      "Condition": {
          "StringEquals": { "autoscaling:ResourceTag/environment": "testing" }
      }
   },
   {
      "Effect": "Allow",
      "Action": "autoscaling:DescribeInstanceRefreshes",
      "Resource": "*"
   }]
}
```

Um eine gewünschte Konfiguration im `StartInstanceRefresh`-Aufruf anzugeben, benötigen Benutzer möglicherweise einige zugehörige Berechtigungen, wie beispielsweise:

- ec2: RunInstances — Um EC2-Instances mithilfe einer Startvorlage zu starten, muss der Benutzer über die `ec2:RunInstances` entsprechende Berechtigung in einer IAM-Richtlinie verfügen. Weitere Informationen finden Sie unter Support für Startvorlagen.
- ec2: CreateTags — Um EC2-Instances von einer Startvorlage aus zu starten, die den Instances und Volumes bei der Erstellung Tags hinzufügt, muss der Benutzer über die `ec2:CreateTags`

entsprechende Berechtigung in einer IAM-Richtlinie verfügen. Weitere Informationen finden Sie unter Erforderliche Berechtigungen zum Markieren von Instances und Volumes.

- iam: PassRole — Um EC2-Instances von einer Startvorlage aus zu starten, die ein Instance-Profil (einen Container für eine IAM-Rolle) enthält, muss der Benutzer auch über die `iam:PassRole` entsprechende Berechtigung in einer IAM-Richtlinie verfügen. Weitere Informationen und eine IAM-Beispielrichtlinie finden Sie unter IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden.
- ssm: GetParameters — Um EC2-Instances von einer Startvorlage aus zu starten, die einen AWS Systems Manager Parameter verwendet, muss der Benutzer auch über die `ssm:GetParameters` entsprechende Berechtigung in einer IAM-Richtlinie verfügen. Weitere Informationen finden Sie unter Verwenden Sie AWS Systems Manager Parameter anstelle von AMI-IDs in Startvorlagen.

Erstellen einer serviceverknüpften Rolle

Amazon EC2 Auto Scaling benötigt Berechtigungen zum Erstellen einer serviceverknüpften Rolle, wenn ein Benutzer in Ihnen zum ersten Mal Amazon EC2 Auto Scaling Scaling-API-Aktionen AWS-Konto aufruft. Wenn die serviceverknüpfte Rolle noch nicht vorhanden ist, erstellt Amazon EC2 Auto Scaling diese in Ihrem Konto. Die serviceverknüpfte Rolle erteilt Amazon EC2 Auto Scaling Berechtigungen, sodass Amazon EC2 Auto Scaling andere in AWS-Services Ihrem Namen anrufen kann.

Damit diese automatische Rollenerstellung möglich ist, müssen Benutzer über Berechtigungen für die Aktion `iam:CreateServiceLinkedRole` verfügen.

```
"Action": "iam:CreateServiceLinkedRole"
```

Im Folgenden wird ein Beispiel für eine Berechtigungsrichtlinie gezeigt, mit der ein Benutzer eine Amazon EC2 Auto Scaling-Rolle für Amazon EC2 Auto Scaling erstellen kann.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling",
      "Condition": {
          "StringLike": { "iam:AWSServiceName":"autoscaling.amazonaws.com" }
```

```
      }
   }]
}
```

Steuern Sie, welche serviceverknüpfte Rolle übergeben werden kann (mit) PassRole

Benutzer, die Auto-Scaling-Gruppen erstellen oder aktualisieren und in der Anfrage eine serviceverknüpfte Rolle mit benutzerdefiniertem Suffix angeben, benötigen die `iam:PassRole`-Berechtigung.

Sie können die `iam:PassRole` Berechtigung verwenden, um die Sicherheit Ihrer vom AWS KMS Kunden verwalteten Schlüssel zu schützen, indem Sie verschiedenen dienstbezogenen Rollen Zugriff auf verschiedene Schlüssel gewähren. Abhängig von den Anforderungen Ihrer Organisation haben Sie möglicherweise einen Schlüssel für das Entwicklungsteam, einen weiteren für das QA-Team und einen weiteren für das Finanzteam. Erstellen Sie zunächst eine dienstbezogene Rolle, die Zugriff auf den erforderlichen Schlüssel hat, z. B. eine dienstbezogene Rolle mit dem Namen. AWSServiceRoleForAutoScaling_devteamkeyaccess Fügen Sie dann diese Richtlinie an eine IAM-Identität an, z. B. einen Benutzer oder eine Rolle.

Die folgende Richtlinie gewährt die Berechtigung, die **`AWSServiceRoleForAutoScaling_devteamkeyaccess`** Rolle an jede Auto-Scaling-Gruppe weiterzugeben, deren Name mit **`devteam-`** beginnt. Wenn die IAM-Identität, die die Auto-Scaling-Gruppe erstellt, versucht, eine andere serviceverknüpfte Rolle anzugeben, wird eine Fehlermeldung ausgegeben. Wenn sie sich dafür entscheiden, keine dienstbezogene Rolle anzugeben, wird stattdessen die AWSServiceRoleForAutoScalingStandardrolle verwendet.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling_devteamkeyaccess",
      "Condition": {
          "StringEquals": { "iam:PassedToService": [ "autoscaling.amazonaws.com" ] },
          "StringLike": { "iam:AssociatedResourceARN":
 [ "arn:aws:autoscaling:region:account-
id:autoScalingGroup:*:autoScalingGroupName/devteam-*" ] }
      }
   }]
```

```
}
```

Weitere Informationen zu serviceverknüpfte Rollen mit benutzerdefiniertem Suffix finden Sie unter Serviceverknüpfte Rollen für Amazon EC2 Auto Scaling.

Serviceübergreifende Confused-Deputy-Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann.

Bei AWS dienstübergreifendem Identitätswechsel kann es zu einem Problem mit dem verwirrten Stellvertreter kommen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der Anruf-Dienst kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte.

Um dies zu verhindern, AWS bietet Tools, mit denen Sie Ihre Daten für alle Dienste mit Dienstprinzipalen schützen können, denen Zugriff auf Ressourcen in Ihrem Konto gewährt wurde. Wir empfehlen den Einsatz des`aws:SourceArn`und`aws:SourceAccount`globale Bedingungskontext-Schlüssel in Vertrauensrichtlinien für Amazon EC2 Auto Scaling Servicerollen. Diese Schlüssel beschränken die Berechtigungen, die Amazon EC2 Auto Scaling der Ressource einem anderen Service gewährt.

Die Werte für die `SourceAccount` Felder `SourceArn` und werden festgelegt, wenn Amazon EC2 Auto Scaling AWS Security Token Service (AWS STS) verwendet, um eine Rolle in Ihrem Namen zu übernehmen.

Um die globalen Bedingungsschlüssel `aws:SourceArn` oder `aws:SourceAccount` zu verwenden, legen Sie den Wert auf den Amazon-Ressourcennamen (ARN) oder das Konto der Ressource, die Amazon EC2 Auto Scaling speichert. Nutzen Sie, wann immer möglich, den spezifischeren Wert `aws:SourceArn`. Legen Sie den Wert auf den ARN oder ein ARN-Muster mit Platzhalterzeichen fest (*) für die unbekannten Teile des ARN. Wenn Sie den ARN der Ressource nicht kennen, verwenden Sie stattdessen `aws:SourceAccount`.

Das folgende Beispiel zeigt, wie Sie die globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in Amazon EC2 Auto Scaling verwenden können, um das Problem des verwirrten Stellvertreters zu vermeiden.

Beispiel: Verwenden`aws:SourceArn`und`aws:SourceAccount`-Bedingungsschlüssel

Eine Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen durchzuführen, wird als Servicerolle bezeichnet. In Fällen, in denen Sie Lifecycle-Hooks erstellen möchten, die Benachrichtigungen an einen anderen Ort als Amazon senden EventBridge, müssen Sie eine Servicerolle erstellen, damit Amazon EC2 Auto Scaling in Ihrem Namen Benachrichtigungen an ein Amazon SNS SNS-Thema oder eine Amazon SQS SQS-Warteschlange senden kann. Wenn Sie nur eine Auto Scaling-Gruppe mit dem betriebsübergreifenden Zugriff verknüpfen möchten, können Sie die Vertrauensrichtlinie der Servicerolle wie folgt angeben.

In dieser Beispiel-Vertrauensrichtlinie werden Bedingungsanweisungen verwendet, um die `AssumeRole`-Fähigkeit für die Servicerolle nur für die Aktionen, die sich auf die angegebene Auto Scaling-Gruppe im angegebenen Konto auswirken. Die Bedingungen `aws:SourceArn` und `aws:SourceAccount` werden unabhängig ausgewertet. Jede Anforderung, die Servicerolle zu verwenden, muss beide Bedingungen erfüllen.

Bevor Sie diese Schlüsselrichtlinie verwenden, ersetzen Sie die Beispiel-Konto-ID, die Region und den Trail-Namen durch gültige Werte aus Ihrem Konto.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "autoscaling.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn":
 "arn:aws:autoscaling:region:account_id:autoScalingGroup:uuid:autoScalingGroupName/my-
asg"
      },
      "StringEquals": {
        "aws:SourceAccount": "account_id"
      }
    }
  }
}
```

Für das obige Beispiel gilt:

- Das `Principal`-Element gibt den Auftraggeber des Dienstes an (`autoscaling.amazonaws.com`).
- Das `Action`-Element spezifiziert die `sts:AssumeRole` Aktion.
- Das `Condition`-Element spezifiziert die globalen Bedingungsschlüssel `aws:SourceArn` und `aws:SourceAccount`. Der ARN der Quelle enthält die Konto-ID, daher ist es nicht erforderlich, `aws:SourceAccount` mit `aws:SourceArn` zu verwenden.

Zusätzliche Informationen

Weitere Informationen finden Sie unterAWS Globale Bedingungskontextschlüssel, Das Problem des verwirrten Stellvertreters, und Ändern einer Rollenvertrauensrichtlinie (Konsole) in der IAM User Guide.

Support für Startvorlagen

Amazon EC2 Auto Scaling unterstützt die Verwendung von Amazon EC2-Startvorlagen mit Ihren Auto Scaling-Gruppen. Es wird empfohlen, Benutzern das Erstellen von Auto Scaling-Gruppen anhand von Startvorlagen zu gestatten, da sie auf diese Weise die neuesten Funktionen von Amazon EC2 Auto Scaling und Amazon EC2 verwenden können. Beispielsweise müssen Benutzer eine Startvorlage angeben, um eine Richtlinie für gemischte Instances verwenden zu können.

Sie können die `AmazonEC2FullAccess`-Richtlinie verwenden, um Benutzern Vollzugriff für die Arbeit mit Amazon EC2 Auto Scaling-Ressourcen, Startvorlagen und anderen EC2-Ressourcen in ihrem Konto zu gewähren. Sie können auch eigene benutzerdefinierte IAM-Richtlinien erstellen, um Benutzern detaillierte Berechtigungen zum Arbeiten mit Startvorlagen zu erteilen, wie in diesem Thema beschrieben.

Eine Beispielrichtlinie, die Sie für Ihre eigene Verwendung anpassen können

Das folgende Beispiel zeigt eine grundlegende Berechtigungsrichtlinie, die Sie für Ihre eigene Verwendung anpassen können. Die Richtlinie gewährt Berechtigungen zum Erstellen, Aktualisieren und Löschen aller Auto-Scaling-Gruppen, jedoch nur, wenn die Gruppe das Tag **`purpose=testing`** verwendet. Diese gewährt dann Berechtigung für alle `Describe`-Aktionen. Da `Describe`-Aktionen keine Berechtigungen auf Ressourcenebene unterstützen, müssen Sie sie in einer separaten Anweisung ohne Bedingungen angeben.

IAM-Identitäten (Benutzer oder Rollen) mit dieser Richtlinie verfügen über die Berechtigung zum Erstellen oder Aktualisieren einer Auto-Scaling-Gruppe mithilfe einer Startvorlage, da sie auch über die Berechtigung zur Verwendung der `ec2:RunInstances`-Aktion verfügen.

```
{
    "Version": "2012-10-17",
    "Statement": [
          {
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:DeleteAutoScalingGroup"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": { "autoscaling:ResourceTag/purpose": "testing" }
            }
         },
         {
            "Effect": "Allow",
            "Action": [
                "autoscaling:Describe*",
                "ec2:RunInstances"
            ],
            "Resource": "*"
        }
    ]
}
```

Benutzer, die Auto-Scaling-Gruppen erstellen oder aktualisieren, benötigen möglicherweise einige zugehörige Berechtigungen, wie beispielsweise:

- ec2: CreateTags — Um den Instances und Volumes bei der Erstellung Tags hinzuzufügen, muss der Benutzer über die `ec2:CreateTags` entsprechende Berechtigung in einer IAM-Richtlinie verfügen. Weitere Informationen finden Sie unter Erforderliche Berechtigungen zum Markieren von Instances und Volumes.
- iam: PassRole — Um EC2-Instances von einer Startvorlage aus zu starten, die ein Instance-Profil (einen Container für eine IAM-Rolle) enthält, muss der Benutzer auch über die `iam:PassRole` entsprechende Berechtigung in einer IAM-Richtlinie verfügen. Weitere Informationen und eine IAM-

Beispielrichtlinie finden Sie unter IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden.

- ssm: GetParameters — Um EC2-Instances von einer Startvorlage aus zu starten, die einen AWS Systems Manager Parameter verwendet, muss der Benutzer auch über die `ssm:GetParameters` entsprechende Berechtigung in einer IAM-Richtlinie verfügen. Weitere Informationen finden Sie unter Verwenden Sie AWS Systems Manager Parameter anstelle von AMI-IDs in Startvorlagen.

Diese Berechtigungen für Aktionen, die beim Starten von Instances ausgeführt werden sollen, werden überprüft, wenn der Benutzer mit einer Auto-Scaling-Gruppe interagiert. Weitere Informationen finden Sie unter Überprüfung der Berechtigungen für `ec2:RunInstances` und `iam:PassRole`.

Die folgenden Beispiele zeigen Richtlinienanweisungen, die Sie verwenden können, um den Zugriff von IAM-Benutzern auf Startvorlagen zu steuern.

Themen

Verlangen, dass Startvorlagen ein bestimmtes Tag haben

Beim Gewähren von `ec2:RunInstances`-Berechtigungen können Sie festlegen, dass Benutzer nur Startvorlagen mit bestimmten Tags oder bestimmten IDs verwenden können, um Berechtigungen beim Starten von Instances mit einer Startvorlage zu beschränken. Sie können auch das AMI und andere Ressourcen steuern, auf die jeder, der Startvorlagen verwendet, beim Starten von Instances verweisen und diese verwenden kann, indem Sie zusätzliche Berechtigungen auf Ressourcenebene für den `RunInstances`-Aufruf angeben.

Das folgende Beispiel schränkt die Berechtigungen für die Aktion `ec2:RunInstances` auf das Starten von Vorlagen ein, die sich in der angegebenen Region befinden und die das Tag **`purpose=testing`** haben. Außerdem erhalten Benutzer Zugriff auf die in einer Startvorlage

angegebenen Ressourcen: AMIs, Instance-Typen, Volumes, Schlüsselpaare, Netzwerkschnittstellen und Sicherheitsgruppen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:region:account-id:launch-template/*",
            "Condition": {
                "StringEquals": { "aws:ResourceTag/purpose": "testing" }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region::image/ami-*",
                "arn:aws:ec2:region:account-id:instance/*",
                "arn:aws:ec2:region:account-id:subnet/*",
                "arn:aws:ec2:region:account-id:volume/*",
                "arn:aws:ec2:region:account-id:key-pair/*",
                "arn:aws:ec2:region:account-id:network-interface/*",
                "arn:aws:ec2:region:account-id:security-group/*"
            ]
        }
    ]
}
```

Weitere Informationen zur Verwendung von tagbasierten Richtlinien mit Startvorlagen finden Sie unter Steuern des Zugriffs auf Startvorlagen mit IAM-Berechtigungen im Amazon EC2 EC2-Benutzerhandbuch.

Eine Startvorlage und eine Versionsnummer verlangen

Sie können IAM-Berechtigungen auch verwenden, um zu erzwingen, dass beim Erstellen oder Aktualisieren von Auto-Scaling-Gruppen eine Startvorlage und die Versionsnummer der Startvorlage angegeben werden müssen.

Im folgenden Beispiel können Benutzer Auto-Scaling-Gruppen nur erstellen und aktualisieren, wenn eine Startvorlage und die Versionsnummer der Startvorlage angegeben sind. Wenn Benutzer mit

dieser Richtlinie die Versionsnummer weglassen, um entweder die Version der Startvorlage `$Latest` oder `$Default` anzugeben, oder versuchen, stattdessen eine Startkonfiguration zu verwenden, schlägt die Aktion fehl.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": { "autoscaling:LaunchTemplateVersionSpecified": "true" }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup"
            ],
            "Resource": "*",
            "Condition": {
                "Null": { "autoscaling:LaunchConfigurationName": "false" }
            }
        }
    ]
}
```

Vorschreiben der Verwendung von Instance Metadata Service Version 2 (IMDSv2)

Für zusätzliche Sicherheit können Sie die Berechtigungen Ihrer Benutzer so festlegen, dass eine Startvorlage verwendet werden muss, für die IMDSv2 erforderlich ist. Weitere Informationen finden Sie unter Konfiguration des Instance-Metadaten-Service im Amazon EC2 EC2-Benutzerhandbuch.

Das folgende Beispiel legt fest, dass der Benutzer die Aktion `ec2:RunInstances` nur aufrufen kann, wenn für die Instance ebenfalls die Verwendung von IMDSv2 (angegeben durch `"ec2:MetadataHttpTokens":"required"`) erforderlich ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RequireImdsV2",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringNotEquals": { "ec2:MetadataHttpTokens": "required" }
            }
        }
    ]
}
```

> **Tip**
>
> Um zu erzwingen, dass Ersatz-Instances von Auto Scaling gestartet werden, die eine neue Startvorlage oder eine neue Version einer Startvorlage mit den konfigurierten Instance-Metadatenoptionen verwenden, können Sie eine Instance-Aktualisierung starten. Weitere Informationen finden Sie unter Aktualisieren von Auto-Scaling-Instances.

Beschränken des Zugriffs auf Amazon EC2-Ressourcen

Das folgende Beispiel steuert die Konfiguration der Instances, die ein Benutzer starten kann, indem der Zugriff auf Amazon EC2-Ressourcen eingeschränkt wird. Um Berechtigungen auf Ressourcenebene für Ressourcen festzulegen, die in einer Startvorlage angegeben sind, müssen Sie die Ressourcen in die `RunInstances`-Aktionsanweisung aufnehmen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account-id:launch-template/*",
                "arn:aws:ec2:region::image/ami-04d5cc9b88example",
                "arn:aws:ec2:region:account-id:subnet/subnet-1a2b3c4d",
                "arn:aws:ec2:region:account-id:volume/*",
```

```
                "arn:aws:ec2:region:account-id:key-pair/*",
                "arn:aws:ec2:region:account-id:network-interface/*",
                "arn:aws:ec2:region:account-id:security-group/sg-903004f88example"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:region:account-id:instance/*",
            "Condition": {
                "StringEquals": { "ec2:InstanceType": ["t2.micro", "t2.small"] }
            }
        }
    ]
}
```

In diesem Beispiel gibt es zwei Anweisungen:

- Die erste Anweisung erfordert, dass Benutzer Instances in einem bestimmten Subnetz (**`subnet-1a2b3c4d`**) starten und dabei eine bestimmte Sicherheitsgruppe (**`sg-903004f88example`**) und ein bestimmtes AMI (**`ami-04d5cc9b88example`**) verwenden. Außerdem erhalten die Benutzer Zugriff auf die in einer Startvorlage angegebenen Ressourcen: Netzwerkschnittstellen, Schlüsselpaare und Volumes.
- Die zweite Anweisung ermöglicht es den Benutzern, Instances nur mit den Instance-Typen **`t2.micro`** und **`t2.small`** zu starten, was aus Kostengründen sinnvoll ist.

  Beachten Sie jedoch, dass es derzeit keine effektive Möglichkeit gibt, Benutzer, die berechtigt sind, Instances mit einer Startvorlage zu starten, vollständig daran zu hindern, andere Instance-Typen zu starten. Dies liegt daran, dass ein in einer Startvorlage angegebener Instance-Typ überschrieben werden kann, sodass Instance-Typen verwendet werden, die mithilfe der attributbasierten Instance-Typauswahl definiert wurden.

Eine vollständige Liste der Berechtigungen auf Ressourcenebene, mit denen Sie die Konfiguration der Instances steuern können, die ein Benutzer starten kann, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EC2 in der Referenz der Serviceberechtigung.

Erforderliche Berechtigungen zum Markieren von Instances und Volumes

Das folgende Beispiel ermöglicht es Benutzern, Instances und Volumes bei der Erstellung zu kennzeichnen. Diese Richtlinie wird benötigt, wenn in der Startvorlage Tags angegeben sind.

Weitere Informationen finden Sie unter Erteilen der Erlaubnis, Ressourcen während der Erstellung zu kennzeichnen im Amazon EC2 EC2-Benutzerhandbuch.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:region:account-id:*/*",
            "Condition": {
                "StringEquals": { "ec2:CreateAction": "RunInstances" }
            }
        }
    ]
}
```

Zusätzliche Berechtigungen für Startvorlagen

Sie müssen Ihren Konsolenbenutzern Berechtigungen für die `ec2:DescribeLaunchTemplates`- und `ec2:DescribeLaunchTemplateVersions`-Aktionen gewähren. Ohne diese Berechtigungen können Startvorlagendaten nicht im Auto Scaling-Gruppen-Assistenten geladen werden, und Benutzer können den Assistenten nicht durchlaufen, um Instances mithilfe einer Startvorlage zu starten. Sie können diese zusätzlichen Aktionen im `Action`-Element einer IAM-Richtlinienanweisung angeben.

Überprüfung der Berechtigungen für **`ec2:RunInstances`** und **`iam:PassRole`**

Benutzer können angeben, welche Version einer Startvorlage ihre Auto-Scaling-Gruppe verwendet. Je nach ihren Berechtigungen kann es sich dabei um eine bestimmte nummerierte Version oder um die Version `$Latest` oder `$Default` der Startvorlage handeln. Wenn Letzteres der Fall ist, seien Sie besonders vorsichtig. Dies kann die Berechtigungen für `ec2:RunInstances` und `iam:PassRole`, die Sie einschränken wollten, außer Kraft setzen.

In diesem Abschnitt wird das Szenario der Verwendung der neuesten oder Standardversion der Startvorlage mit einer Auto-Scaling-Gruppe erläutert.

Wenn ein Benutzer die `CreateAutoScalingGroup`-, `UpdateAutoScalingGroup`- oder `StartInstanceRefresh`-APIs aufruft, prüft Amazon EC2 Auto Scaling seine Berechtigungen anhand der Version der Startvorlage, die zu diesem Zeitpunkt die neueste oder die Standardversion ist, bevor es mit der Anforderung fortfährt. Dadurch werden die Berechtigungen für Aktionen

validiert, die beim Starten von Instances abgeschlossen werden müssen, wie z. B. die Aktionen `ec2:RunInstances` und `iam:PassRole`. Um dies zu erreichen, führen wir einen Amazon EC2 RunInstancesEC2-Probelauf durch, um zu überprüfen, ob der Benutzer über die erforderlichen Berechtigungen für die Aktion verfügt, ohne die Anfrage tatsächlich zu stellen. Wenn eine Antwort zurückgegeben wird, wird sie von Amazon EC2 Auto Scaling gelesen. Wenn die Berechtigungen des Benutzers eine bestimmte Aktion nicht zulassen, lässt Amazon EC2 Auto Scaling die Anfrage fehlschlagen und gibt dem Benutzer eine Fehlermeldung mit Informationen über die fehlende Berechtigung zurück.

Nach Abschluss der ersten Überprüfung und Anforderung startet Amazon EC2 Auto Scaling Instances bei jedem Start mit der neuesten Version oder Standardversion, auch wenn sie sich geändert hat. Dabei werden die Berechtigungen der serviceverknüpften Rolle verwendet. Das bedeutet, dass ein Benutzer, der die Startvorlage verwendet, diese möglicherweise aktualisieren kann, um eine IAM-Rolle an eine Instance zu übergeben, auch wenn er nicht über die `iam:PassRole`-Berechtigung verfügt.

Verwenden Sie den `autoscaling:LaunchTemplateVersionSpecified`-Bedingungsschlüssel, wenn Sie einschränken möchten, wer Zugriff auf die Konfiguration von Gruppen hat, um die Version `$Latest` oder `$Default` zu verwenden. Dadurch wird sichergestellt, dass die Auto-Scaling-Gruppe nur eine bestimmte nummerierte Version akzeptiert, wenn ein Benutzer die APIs `CreateAutoScalingGroup` und `UpdateAutoScalingGroup` aufruft. Ein Beispiel, das zeigt, wie dieser Bedingungsschlüssel zu einer IAM-Richtlinie hinzugefügt wird, finden Sie unter Eine Startvorlage und eine Versionsnummer verlangen.

Für Auto-Scaling-Gruppen, die für die Verwendung der Startvorlagenversion `$Latest` oder `$Default` konfiguriert sind, sollten Sie einschränken, wer Versionen der Startvorlage erstellen und verwalten kann, einschließlich der `ec2:ModifyLaunchTemplate`-Aktion, die es einem Benutzer ermöglicht, die Standardversion der Startvorlage anzugeben. Weitere Informationen finden Sie unter Steuern von Versionsberechtigungen im Amazon EC2 EC2-Benutzerhandbuch.

Zugehörige Ressourcen

Weitere Informationen zu Berechtigungen zum Anzeigen, Erstellen und Löschen von Startvorlagen und Startvorlagenversionen finden Sie unter Steuern des Zugriffs auf Startvorlagen mit IAM-Berechtigungen im Amazon EC2 EC2-Benutzerhandbuch.

Weitere Informationen zu Berechtigungen auf Ressourcenebene, mit denen Sie den Zugriff auf den `RunInstances`-Aufruf steuern können, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EC2 in der Referenz der Serviceberechtigung.

IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden

Anwendungen, die in Amazon EC2-Instances ausgeführt werden, benötigen Anmeldeinformationen, um auf andere AWS-Services-Services zugreifen zu können. Verwenden Sie eine IAM-Rolle, um diese Anmeldeinformationen auf sichere Weise bereitstellen zu können. Die Rolle stellt temporäre Berechtigungen bereit, die von der Anwendung verwendet werden können, wenn sie auf andere AWS -Ressourcen zugreift. Die Berechtigungen der Rolle bestimmen, welche Aktionen die Anwendung durchführen darf.

Für Instances in einer Auto Scaling-Gruppe müssen Sie eine Startkonfiguration oder Vorlage erstellen und ein Instance-Profil wählen, das mit den Instances verknüpft werden soll. Ein Instance-Profil ist ein Container für eine IAM-Rolle, mit dem Amazon EC2 einer Instance die IAM-Rolle übergibt, wenn die Instance gestartet wird. Erstellen Sie zunächst eine IAM-Rolle, die über alle für den Zugriff auf die Ressourcen erforderlichen Berechtigungen verfügt. AWS Erstellen Sie anschließend das Instance-Profil und weisen Sie diesem die Rolle zu.

> **Note**
>
> Als bewährte Methode empfehlen wir dringend, die Rolle so zu erstellen, dass sie über die Mindestberechtigungen für andere Benutzer verfügt AWS-Services , die für Ihre Anwendung erforderlich sind.

Inhalt

Voraussetzungen

Erstellen Sie die IAM-Rolle, die Ihre unter Amazon EC2 ausgeführte Anwendung verwenden kann. Wählen Sie die entsprechenden Berechtigungen, so dass die Anwendung, der die Rolle im Anschluss übertragen wird, die benötigten API-Aufrufe durchführen kann.

Wenn Sie die IAM-Konsole anstelle des AWS CLI oder eines der AWS SDKs verwenden, erstellt die Konsole automatisch ein Instanzprofil und weist diesem denselben Namen zu wie der Rolle, der es entspricht.

So erstellen Sie eine IAM-Rolle (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich auf der linken Seite Roles (Rollen).
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die Option AWS - Service.
5. Wählen Sie für Ihren Anwendungsfall die Option EC2 und dann Next (Weiter) aus.
6. Wenn möglich, wählen Sie die Richtlinie aus, die für die Berechtigungsrichtlinie verwendet werden soll, oder wählen Create policy (Richtlinie erstellen), um eine neue Registerkarte im Browser zu öffnen und eine vollständig neue Richtlinie zu erstellen. Weitere Informationen finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch. Nachdem Sie die Richtlinie erstellt haben, schließen Sie die Registerkarte und kehren zur ursprünglichen Registerkarte zurück. Aktivieren Sie die Kontrollkästchen neben den Berechtigungsrichtlinien, die der Service haben soll.
7. (Optional) Legen Sie eine Berechtigungsgrenze fest. Dies ist eine erweiterte Funktion, die für Servicerollen verfügbar ist. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
8. Wählen Sie Weiter aus.
9. Geben Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen), für Role name (Rollenname) einen Rollennamen ein, mit dem der Zweck dieser Rolle einfach zu erkennen ist. Dieser Name muss innerhalb Ihres AWS-Konto eindeutig sein. Da andere AWS Ressourcen möglicherweise auf die Rolle verweisen, können Sie den Namen der Rolle nicht bearbeiten, nachdem sie erstellt wurde.
10. Prüfen Sie die Rolle und klicken Sie dann auf Create Role (Rolle erstellen).

IAM-Berechtigungen

Verwenden Sie eine identitätsbasierte IAM-Richtlinie, um den Zugriff auf Ihre neue IAM-Rolle zu steuern. Die `iam:PassRole`-Berechtigung ist für die IAM-Identität (Benutzer oder Rolle) erforderlich,

die eine Auto-Scaling-Gruppe mithilfe einer Startvorlage erstellt oder aktualisiert, die ein Instance-Profil angibt.

Die folgende Beispielrichtlinie gewährt die Berechtigung, nur IAM-Rollen weiterzugeben, deren Name mit **`qateam-`** beginnt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::account-id:role/qateam-*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn"
                    ]
                }
            }
        }
    ]
}
```

> **Important**
>
> Informationen darüber, wie Amazon EC2 Auto Scaling Berechtigungen für die `iam:PassRole`-Aktion für eine Auto-Scaling-Gruppe validiert, die eine Startvorlage verwendet, finden Sie unter Überprüfung der Berechtigungen für `ec2:RunInstances` und `iam:PassRole`.

Erstellen einer Startvorlage

Wenn Sie die Startvorlage mithilfe von erstellen AWS Management Console, wählen Sie im Abschnitt Erweiterte Details die Rolle aus dem IAM-Instanzprofil aus. Weitere Informationen finden Sie unter Erstellen einer Startvorlage mithilfe erweiterter Einstellungen.

Wenn Sie die Startvorlage mit dem Befehl create-launch-template aus dem erstellen AWS CLI, geben Sie den Instanzprofilnamen Ihrer IAM-Rolle an, wie im folgenden Beispiel gezeigt.

```
aws ec2 create-launch-template --launch-template-name my-lt-with-instance-profile --
version-description version1 \
--launch-template-data
 '{"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro","IamInstanceProfile":
{"Name":"my-instance-profile"}}'
```

Weitere Informationen finden Sie auch unter

Weitere Informationen, die Sie beim Lernen von IAM-Rollen für Amazon EC2 unterstützen, finden Sie unter:

- IAM-Rollen für Amazon EC2 im Amazon EC2 EC2-Benutzerhandbuch
- Verwenden von Instance-Profilen und Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch

Compliance-Validierung für Amazon EC2 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 Implementierung 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— Auf diese AWS-Service Weise erhalten Sie 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.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- 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.

Compliance mit PCI DSS

Amazon EC2 Auto Scaling unterstützt die Verarbeitung, Speicherung und Übertragung von Kreditkartendaten durch einen Händler oder Dienstanbieter. Außerdem wurde seine Konformität mit dem Payment Card Industry (PCI) Data Security Standard (DSS) bestätigt. Weitere Informationen zu

PCI DSS, einschließlich der Möglichkeit, eine Kopie des AWS PCI Compliance Package anzufordern, finden Sie unter PCI DSS Level 1.

Informationen zum Erreichen der PCI-DSS-Konformität für Ihre AWS Workloads finden Sie im folgenden Compliance-Leitfaden:

- Payment Card Industry Data Security Standard (PCI DSS) 3.2.1 auf AWS

Amazon EC2 Auto Scaling und Schnittstellen-VPC-Endpunkte

Sie können die Sicherheit Ihrer VPC erhöhen, indem Sie Amazon EC2 Auto Scaling so konfigurieren, dass ein Schnittstellen-VPC-Endpunkt verwendet wird. Schnittstellenendpunkte werden von einer Technologie unterstützt AWS PrivateLink, die es Ihnen ermöglicht, privat auf Amazon EC2 Auto Scaling-APIs zuzugreifen, indem der gesamte Netzwerkverkehr zwischen Ihrer VPC und Amazon EC2 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 AWS PrivateLink ist nicht erforderlich, 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 einer VPC-Endpunktrichtlinie

Erstellen eines Schnittstellen-VPC-Endpunkts

Erstellen Sie einen Endpunkt für Amazon EC2 Auto Scaling mit dem folgenden Service-Namen:

```
com.amazonaws.region.autoscaling
```

Weitere Informationen finden Sie im AWS PrivateLink Handbuch unter Zugreifen auf einen AWS Dienst über einen Schnittstellen-VPC-Endpunkt.

Sie müssen keine Amazon EC2 Auto Scaling-Einstellungen ändern. Amazon EC2 Auto Scaling ruft andere AWS Services entweder über Service-Endpunkte oder VPC-Endpunkte mit privater Schnittstelle auf, je nachdem, welche verwendet werden.

Erstellen einer VPC-Endpunktrichtlinie

Sie können Ihrem VPC-Endpunkt eine Richtlinie anfügen, um den Zugriff auf die Amazon EC2 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": "autoscaling:DeleteScalingPolicy",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

Weitere Informationen finden Sie im Handbuch unter Steuern des Zugriffs auf VPC-Endpunkte mithilfe von Endpunktrichtlinien.AWS PrivateLink

Fehlersuche bei Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling liefert genaue Fehlermeldungen mit einer kurzen Beschreibung, damit Sie Probleme leichter beheben können. Sie finden die Fehlermeldungen in der Beschreibung der Skalierungen.

Themen

Abrufen einer Fehlermeldung aus Skalierungen

Wenn Sie eine Fehlermeldung aus der Beschreibung der Skalierungen abrufen möchten, verwenden Sie den Befehl describe-scaling-activities. Sie haben eine Aufzeichnung von Skalierungsaktivitäten, die sechs Wochen zurückreicht. Skalierungsaktivitäten werden nach Startzeit sortiert, wobei die neuesten Skalierungsaktivitäten zuerst aufgelistet werden.

> **Note**
>
> Die Skalierungen werden auch im Aktivitätsverlauf in der Amazon EC2 Auto Scaling-Konsole auf der Registerkarte Activity (Aktivität) für die Auto-Scaling-Gruppe angezeigt.

Verwenden Sie den folgenden Befehl, um die Skalierungsaktivitäten für eine bestimmte Auto-Scaling-Gruppe anzuzeigen.

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

Im Folgenden sehen Sie eine Beispielantwort, in der der aktuelle Status der Aktivität unter `StatusCode` und die Fehlermeldung unter `StatusMessage` zu finden ist.

```
{
    "Activities": [
        {
            "ActivityId": "3b05dbf6-037c-b92f-133f-38275269dc0f",
            "AutoScalingGroupName": "my-asg",
            "Description": "Launching a new EC2 instance: i-003a5b3ffe1e9358e.  Status
 Reason: Instance failed to complete user's Lifecycle Action: Lifecycle Action with
 token e85eb647-4fe0-4909-b341-a6c42d8aba1f was abandoned: Lifecycle Action Completed
 with ABANDON Result",
            "Cause": "At 2021-01-11T00:35:52Z a user request created an
 AutoScalingGroup changing the desired capacity from 0 to 1.  At 2021-01-11T00:35:53Z
 an instance was started in response to a difference between desired and actual
 capacity, increasing the capacity from 0 to 1.",
            "StartTime": "2021-01-11T00:35:55.542Z",
            "EndTime": "2021-01-11T01:06:31Z",
            "StatusCode": "Cancelled",
            "StatusMessage": "Instance failed to complete user's Lifecycle Action:
 Lifecycle Action with token e85eb647-4fe0-4909-b341-a6c42d8aba1f was abandoned:
 Lifecycle Action Completed with ABANDON Result",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-
west-2b\"...}",
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
        },
      ...
    ]
}
```

Eine Beschreibung der Felder in der Ausgabe finden Sie unterAktivität in der Amazon EC2 Auto Scaling API-Referenz.

Anzeigen von Skalierungsaktivitäten für eine gelöschte-Gruppe

Um die Skalierungsaktivitäten anzuzeigen, nachdem die Auto-Scaling-Gruppe gelöscht wurde, fügen Sie dem describe-scaling-activities-Befehl die Option `--include-deleted-groups` wie folgt hinzu.

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg --include-
deleted-groups
```

Nachfolgend finden Sie eine Beispielantwort mit einer Skalierungsaktivität für eine gelöschte Gruppe.

```
{
    "Activities": [
        {
            "ActivityId": "e1f5de0e-f93e-1417-34ac-092a76fba220",
            "AutoScalingGroupName": "my-asg",
            "Description": "Launching a new EC2 instance.  Status Reason: Your Spot
 request price of 0.001 is lower than the minimum required Spot request fulfillment
 price of 0.0031. Launching EC2 instance failed.",
            "Cause": "At 2021-01-13T20:47:24Z a user request update of AutoScalingGroup
 constraints to min: 1, max: 5, desired: 3 changing the desired capacity from 0 to 3.
  At 2021-01-13T20:47:27Z an instance was started in response to a difference between
 desired and actual capacity, increasing the capacity from 0 to 3.",
            "StartTime": "2021-01-13T20:47:30.094Z",
            "EndTime": "2021-01-13T20:47:30Z",
            "StatusCode": "Failed",
            "StatusMessage": "Your Spot request price of 0.001 is lower than the
 minimum required Spot request fulfillment price of 0.0031. Launching EC2 instance
 failed.",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-
west-2b\"...}",
            "AutoScalingGroupState": "Deleted",
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
        },
     ...
    ]
}
```

Skalierungsaktivitäten ausschalten

Sie haben die folgenden Optionen, wenn Sie ein Problem untersuchen möchten, ohne dass es zu Störungen durch Skalierungsrichtlinien oder geplante Aktionen kommt:

- Verhindern Sie, dass alle dynamischen Skalierungsrichtlinien und geplanten Aktionen Änderungen an der gewünschten Kapazität der Gruppe bewirken, indem Sie die `AlarmNotification` `ScheduledActions` Endprozesse aussetzen. Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling Scaling-Prozesse aussetzen und fortsetzen.

- Deaktivieren Sie einzelne dynamische Skalierungsrichtlinien, damit sie nicht die gewünschte Kapazität der Gruppe als Reaktion auf Laständerungen ändern. Weitere Informationen finden Sie unter Eine Skalierungsrichtlinie für eine Auto-Scaling-Gruppe deaktivieren.
- Aktualisieren Sie die Skalierungsrichtlinien für die individuelle Zielverfolgung so, dass sie nur horizontal skalieren (Kapazität hinzufügen), indem Sie den Scale-In-Teil der Richtlinie deaktivieren. Diese Methode verhindert, dass die gewünschte Kapazität der Gruppe schrumpft, ermöglicht es jedoch, sie bei steigender Auslastung zu erhöhen. Weitere Informationen finden Sie unter Skalierungsrichtlinien für die Ziel-Nachverfolgung für Amazon EC2 Auto Scaling.
- Aktualisieren Sie Ihre Richtlinie zur vorausschauenden Skalierung auf den Modus „Nur Prognosen“. Im Modus „Nur Prognose“ generiert die vorausschauende Skalierung zwar weiterhin Prognosen, erhöht aber nicht automatisch die Kapazität. Weitere Informationen finden Sie unter Erstellen Sie eine Richtlinie zur vorausschauenden Skalierung.

Weitere Ressourcen zur Fehlerbehebung

Auf den folgenden Seiten finden Sie zusätzliche Informationen zur Behebung von Problemen mit Amazon EC2 Auto Scaling.

- Eine Skalierung für eine Auto-Scaling-Gruppe überprüfen
- Überwachungsgrafiken in der Amazon EC2 Auto Scaling-Konsole anzeigen
- Zustandsprüfungen für Instances in einer Auto-Scaling-Gruppe
- Überlegungen zu und Einschränkungen für Lebenszyklus-Hooks
- Eine Lebenszyklus-Aktion abschließen
- Stellen Sie Netzwerkkonnektivität für Ihre Auto-Scaling-Instances mit Amazon VPC bereit
- Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe
- Eine Skalierungsrichtlinie für eine Auto-Scaling-Gruppe deaktivieren
- Amazon EC2 Auto Scaling Scaling-Prozesse aussetzen und fortsetzen
- Steuern welche Auto-Scaling-Instances beim Abskalieren beendet werden
- Löschen der Auto-Scaling-Infrastruktur
- Kontingente für Auto Scaling Scaling-Ressourcen und Gruppen

Die folgenden AWS Ressourcen können ebenfalls hilfreich sein:

- Themen zu Amazon EC2 Auto Scaling im AWS Knowledge Center

- Fragen zu Amazon EC2 Auto Scaling auf re:POST AWS
- Beiträge zu Amazon EC2 Auto Scaling im AWS Compute-Blog
- Fehlerbehebung CloudFormation im AWS CloudFormation Benutzerhandbuch

Die Fehlerbehebung erfordert oft eine iterative Abfrage und Erkennung durch einen Experten oder eine Community von Helfern. Wenn Sie nach dem Ausprobieren der Vorschläge in diesem Abschnitt weiterhin Probleme haben, wenden Sie sich an AWS Support (klicken Sie auf Support AWS Management Console, Support Center) oder stellen Sie mithilfe des Amazon EC2 Auto Scaling-Tags eine Frage zu AWS re:POST.

Fehlersuche bei Amazon EC2 Auto Scaling: Startfehler von EC2-Instance

Auf dieser Seite finden Sie Informationen zu Ihren EC2-Instances, die nicht gestartet werden konnten, mögliche Ursachen und Maßnahmen, die Sie zum Lösen der Probleme ergreifen können.

Wie Sie eine Fehlermeldung abrufen, erfahren Sie unter Abrufen einer Fehlermeldung aus Skalierungen.

Falls Ihre EC2-Instances nicht gestartet werden können, erscheinen eine oder mehrere der folgenden Fehlermeldungen:

Startprobleme

- Die angefragte Konfiguration wird derzeit nicht unterstützt.
- Die Sicherheitsgruppe <Name der Sicherheitsgruppe> ist nicht vorhanden. Die EC2-Instance konnte nicht gestartet werden.
- Das Schlüsselpaar <mit Ihrer EC2-Instance verbundenes Schlüsselpaar> ist nicht vorhanden. Die EC2-Instance konnte nicht gestartet werden.
- Der von Ihnen angeforderte Instance-Typ (<Instance type>) wird in der von Ihnen angeforderten Availability Zone (<instance Availability Zone>) nicht unterstützt...
- Ihr Spot-Anfragepreis von 0,015 ist niedriger als der erforderliche Mindestpreis für Spot-Anfragen von 0,0735...
- Ungültiger Gerätename <Gerätename> / Ungültiger Gerätename beim Hochladen. Die EC2-Instance konnte nicht gestartet werden.

- Der Wert (<Name des verbundenen Instance-Speichergeräts>) für den Parameter virtualName ist ungültig... Die EC2-Instance konnte nicht gestartet werden.
- EBS-Blockgerät-Zuweisungen werden für Instance-Speicher-AMIs nicht unterstützt.
- Platzierungsgruppen dürfen nicht mit Instanzen des Typs '<Instance type>' verwendet werden. Die EC2-Instance konnte nicht gestartet werden.
- Kunde. InternalError: Client-Fehler beim Start.
- Wir haben derzeit nicht genügend <instance type>-Kapazität in der Availability Zone, die Sie angefragt haben. Die EC2-Instance konnte nicht gestartet werden.
- Die angefragte Reservierung ist nicht ausreichend kompatibel und hat nicht genügend freie Kapazität für diese Anfrage. Die EC2-Instance konnte nicht gestartet werden.
- Ihre Kapazitätsblock-Reservierung <reservation id> ist noch nicht aktiv. Die EC2-Instance konnte nicht gestartet werden.
- Es ist keine Spot-Kapazität verfügbar, die Ihrer Anforderung entspricht. Die EC2-Instance konnte nicht gestartet werden.
- <number of instances> Instance wird/Instances werden bereits ausgeführt. Die EC2-Instance konnte nicht gestartet werden.

Die angefragte Konfiguration wird derzeit nicht unterstützt.

Ursache: Einige Optionen in Ihrer Startvorlage oder Startkonfiguration sind möglicherweise nicht mit dem Instance-Typ kompatibel, oder die Instance-Konfiguration wird in der von Ihnen angeforderten AWS Region oder Availability Zones möglicherweise nicht unterstützt.

Lösung: Versuchen Sie es mit einer anderen Instanzkonfiguration. Informationen zur Suche nach einem Instance-Typ, der Ihren Anforderungen entspricht, finden Sie unter Suchen nach einem Amazon EC2 EC2-Instance-Typ im Amazon EC2 EC2-Benutzerhandbuch.

Weitere Informationen zum Beheben dieses Problems finden Sie unter:

- Stellen Sie sicher, dass Sie ein AMI ausgewählt haben, das von Ihrem Instance-Typ unterstützt wird. Wenn der Instance-Typ beispielsweise einen ARM-basierten AWS Graviton-Prozessor anstelle eines Intel Xeon-Prozessors verwendet, benötigen Sie ein ARM-kompatibles AMI. Weitere Informationen zur Auswahl eines kompatiblen Instance-Typs finden Sie unter Kompatibilität bei der Änderung des Instance-Typs im Amazon EC2 EC2-Benutzerhandbuch.
- Testen Sie, ob der Instance-Typ in Ihrer angeforderten Region und Availability Zones verfügbar ist. Die Instance-Typen der neuesten Generation sind möglicherweise noch nicht in einer

bestimmten Region oder Availability Zone verfügbar. Die Instance-Typen der älteren Generation sind möglicherweise nicht in neueren Regionen oder Availability Zones verfügbar. Um nach Instanztypen zu suchen, die nach Standort (Region oder Availability Zone) angeboten werden, verwenden Sie den Befehl describe-instance-type-offerings. Weitere Informationen finden Sie unter Suchen nach einem Amazon EC2 EC2-Instance-Typ im Amazon EC2 EC2-Benutzerhandbuch.

- Wenn Sie Dedicated Instances oder Dedicated Hosts verwenden, stellen Sie sicher, dass Sie einen Instance-Typ ausgewählt haben, der als Dedicated Instance oder Dedicated Host unterstützt wird.

Die Sicherheitsgruppe <Name der Sicherheitsgruppe> ist nicht vorhanden. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Die Sicherheitsgruppe in Ihrer Startkonfiguration wurde möglicherweise gelöscht.

Solution (Lösung):

1. Verwenden Sie den Befehl describe-security-groups, um eine Liste der Sicherheitsgruppen abzurufen, die Ihrem Konto zugeordnet sind.
2. Wählen Sie in der Liste die Sicherheitsgruppen aus, die verwendet werden sollen. Wenn Sie stattdessen eine Sicherheitsgruppe erstellen möchten, verwenden Sie den Befehl create-security-group.
3. Erstellen Sie eine neue Startvorlage oder Startkonfiguration.
4. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit der neuen Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

Das Schlüsselpaar <mit Ihrer EC2-Instance verbundenes Schlüsselpaar> ist nicht vorhanden. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Das Schlüsselpaar, mit dem die Instance gestartet wurde, wurde möglicherweise gelöscht.

Solution (Lösung):

1. Verwenden Sie den Befehl describe-key-pairs, um eine Liste der Schlüsselpaare abzurufen, die Ihnen zur Verfügung stehen.
2. Wählen Sie in der Liste das Schlüsselpaar aus, das verwendet werden soll. Wenn Sie stattdessen ein Schlüsselpaar erstellen möchten, verwenden Sie den Befehl create-key-pair.

3. Erstellen Sie eine neue Startvorlage oder Startkonfiguration.
4. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit der neuen Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

Der von Ihnen angeforderte Instance-Typ (<Instance type>) wird in der von Ihnen angeforderten Availability Zone (<instance Availability Zone>) nicht unterstützt...

Fehlermeldung: Der von Ihnen angeforderte Instance-Typ (<instance type>) wird in der von Ihnen angeforderten Availability Zone (<Instance Availability Zone>) nicht unterstützt...(Launching EC2 instance failed)… Starten der EC2-Instance fehlgeschlagen.

Ursache: Die in Ihrer Auto Scaling-Gruppe angegebenen Availability Zones unterstützen den von Ihnen gewählten Instance-Typ nicht.

Solution (Lösung):

1. Überprüfen Sie, welche Availability Zones den von Ihnen gewählten Instance-Typ unterstützen, indem Sie den Befehl describe-instance-type-offerings verwenden oder von der Amazon EC2-Konsole aus den Wert für Availability Zones auf dem Netzwerkbereich der Seite Instance types überprüfen.
2. Aktualisieren oder entfernen Sie das Subnetz für alle nicht unterstützten Zonen in den Einstellungen Ihrer Auto-Scaling-Gruppe mithilfe des Befehls update-auto-scaling-group. Weitere Informationen finden Sie unter Hinzufügen oder Entfernen von Availability Zones.

Ihr Spot-Anfragepreis von 0,015 ist niedriger als der erforderliche Mindestpreis für Spot-Anfragen von 0,0735...

Ursache: Der Spot-Höchstpreis in Ihrer Anfrage ist niedriger als der Spot-Preis für den ausgewählten Instance-Typ.

Lösung: Senden Sie eine neue Anforderung mit einem höheren Spot-Höchstpreis (möglicherweise dem On-Demand-Preis). Bisher basierte der von Ihnen gezahlte Spot-Preis auf Geboten. Heute zahlen Sie den aktuellen Spot-Preis. Wenn Sie Ihren Höchstpreis höher setzen, ist die Chance höher, dass der Amazon EC2 Spot-Service Ihre erforderliche Kapazität startet und erhält.

Ungültiger Gerätename <Gerätename> / Ungültiger Gerätename beim Hochladen. Die EC2-Instance konnte nicht gestartet werden.

Ursache 1: Die Blockgerät-Zuweisungen in Ihrer Startvorlage enthalten möglicherweise Blockgerätenamen, die nicht verfügbar sind oder derzeit nicht unterstützt werden.

Solution (Lösung):

1. Überprüfen Sie, welche Gerätenamen für Ihre spezielle Instance-Konfiguration verfügbar sind. Weitere Informationen zur Benennung von Geräten finden Sie unter Gerätenamen auf Linux-Instances im Amazon EC2 EC2-Benutzerhandbuch.
2. Erstellen Sie manuell eine Amazon EC2-Instance, die nicht Teil der Auto-Scaling-Gruppe ist, und untersuchen Sie das Problem. Wenn die Konfiguration für die Benennung der Blockgeräte nicht mit den Namen im Amazon Machine Image (AMI) übereinstimmt, schlägt die Instance beim Start fehl. Weitere Informationen finden Sie unter Gerätezuordnungen blockieren im Amazon EC2 EC2-Benutzerhandbuch.
3. Nachdem Sie bestätigt haben, dass Ihre Instance erfolgreich gestartet wurde, verwenden Sie den Befehl describe-volumes, um zu sehen, wie die Volumes der Instance ausgesetzt sind.
4. Erstellen Sie eine neue Startvorlage oder Startkonfiguration mit dem Gerätenamen, der in der Volume-Beschreibung aufgeführt ist.
5. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit der neuen Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

Der Wert (<Name des verbundenen Instance-Speichergeräts>) für den Parameter virtualName ist ungültig... Die EC2-Instance konnte nicht gestartet werden.

Ursache: Das Format, in dem der virtuelle Name des Blockgeräts angegeben wurde, ist falsch.

Solution (Lösung):

1. Erstellen Sie eine neue Startvorlage oder Startkonfiguration, indem Sie die Gerätenamen im Parameter `virtualName` angeben. Informationen zum Gerätenamenformat finden Sie unter Gerätebenennung auf Linux-Instances im Amazon EC2 EC2-Benutzerhandbuch.
2. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit der neuen Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

EBS-Blockgerät-Zuweisungen werden für Instance-Speicher-AMIs nicht unterstützt.

Ursache: Die Blockgerät-Zuweisungen, die in der Startvorlage oder Startkonfiguration angegeben wurden, werden auf Ihrer Instance nicht unterstützt.

Solution (Lösung):

1. Erstellen Sie eine neue Startvorlage oder Startkonfiguration mit Blockgerät-Zuweisungen, die von Ihrem Instance-Typ unterstützt werden. Weitere Informationen finden Sie unter Gerätezuweisung blockieren im Amazon EC2 EC2-Benutzerhandbuch.
2. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit der neuen Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

Platzierungsgruppen dürfen nicht mit Instanzen des Typs '<Instance type>' verwendet werden. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Ihre Cluster Placement-Gruppe enthält einen ungültigen Instance-Typ.

Solution (Lösung):

1. Informationen zu gültigen Instance-Typen, die von den Placement-Gruppen unterstützt werden, finden Sie unter Placement-Gruppen im Amazon EC2 EC2-Benutzerhandbuch.
2. Folgen Sie der Anleitung unter Platzierungsgruppen, um eine neue Platzierungsgruppe zu erstellen.
3. Alternativ können Sie eine neue Startvorlage oder Startkonfiguration mit einem unterstützten Instance-Typ erstellen.
4. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit einer neuen Platzierungsgruppe, Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

Kunde. InternalError: Client-Fehler beim Start.

Problem: Amazon EC2 Auto Scaling versucht, eine Instance zu starten, die über ein verschlüsseltes EBS-Volume verfügt, aber die serviceverknüpfte Rolle hat keinen Zugriff auf den vom AWS KMS Kunden verwalteten Schlüssel, mit dem sie verschlüsselt wurde. Weitere Informationen finden Sie unter Erforderliche AWS KMS Schlüsselrichtlinie für die Verwendung mit verschlüsselten Volumes.

Ursache 1: Sie benötigen eine Schlüsselrichtlinie, welche die Berechtigung erteilt, den kundenverwalteten Schlüssel für die richtige servicebezogene Rolle zu verwenden.

Lösung 1: Erlauben Sie der serviceverknüpften Rolle, den kundenverwalteten Schlüssel wie folgt zu verwenden:

1. Ermitteln Sie, welche serviceverknüpfte Rolle für diese Auto Scaling-Gruppe verwendet werden soll.
2. Aktualisieren Sie die Schlüsselrichtlinie für den kundenverwalteten Schlüssel und erlauben Sie der serviceverknüpften Rolle, den vom Kunden verwalteten Schlüssel zu verwenden.
3. Aktualisieren Sie die Auto Scaling-Gruppe, damit sie die serviceverknüpfte Rolle verwenden kann.

Ein Beispiel für eine Schlüsselrichtlinie, mit der die serviceverknüpfte Rolle den kundenverwalteten Schlüssel verwenden kann, finden Sie unterBeispiel 1: Schlüsselrichtlinienabschnitte, welche Zugriff auf den kundenverwalteten Schlüssel erlauben.

Ursache 2: Wenn sich der vom Kunden verwaltete Schlüssel und die Auto Scaling Scaling-Gruppe in unterschiedlichen AWS Konten befinden, müssen Sie den kontoübergreifenden Zugriff auf den vom Kunden verwalteten Schlüssel konfigurieren, um der richtigen serviceverknüpften Rolle die Erlaubnis zur Verwendung des vom Kunden verwalteten Schlüssels zu erteilen.

Lösung 2: Erlauben Sie der servicebezogenen Rolle im externen Konto, den kundenverwalteten Schlüssel im lokalen Konto wie folgt zu verwenden:

1. Aktualisieren Sie die Schlüsselrichtlinie für den kundenverwalteten Schlüssel, um dem Konto der Auto-Scaling-Gruppe Zugriff auf den kundenverwalteten Schlüssel zu gewähren.
2. Definieren Sie in der Auto-Scaling-Gruppe einen IAM-Benutzer oder eine IAM-Rolle zur Erstellung einer Zuwendung.
3. Ermitteln Sie, welche serviceverknüpfte Rolle für diese Auto Scaling-Gruppe verwendet werden soll.
4. Erstellen Sie eine Zuwendung für den kundenverwalteten Schlüssel, wobei die serviceverknüpfte Rolle als berechtigtes Prinzipal dient.
5. Aktualisieren Sie die Auto Scaling-Gruppe, damit sie die serviceverknüpfte Rolle verwenden kann.

Weitere Informationen finden Sie unter Beispiel 2: Schlüsselrichtlinienabschnitte, welche Zugriff auf den kundenverwalteten Schlüssel über mehrere Konten erlauben.

Lösung 3: Verwenden Sie einen kundenverwalteten Schlüssel im selben AWS -Konto wie der Auto-Scaling-Gruppe

1. Kopieren Sie den Snapshot und verschlüsseln Sie ihn erneut mit einem anderen kundenverwalteten Schlüssel im Konto der Auto-Scaling-Gruppe.
2. Erlauben Sie der serviceverknüpften Rolle, den neuen kundenverwalteten Schlüssel zu verwenden. Lesen Sie die Schritte für Lösung 1.

Wir haben derzeit nicht genügend <instance type>-Kapazität in der Availability Zone, die Sie angefragt haben. Die EC2-Instance konnte nicht gestartet werden.

Error Message: Wir haben in der von Ihnen angeforderten Availability Zone (<requested Availability Zone>), derzeit nicht genügend <instance type>-Kapazität. Unser System arbeitet an der Bereitstellung zusätzlicher Kapazität. Sie können im Augenblick <instance type> Kapazität erhalten, indem Sie in Ihrer Anfrage keine Availability Zone auswählen oder <llist of Availability Zones that currently supports the instance type> auswählen. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Derzeit wird die Kombination aus angefordertem Instance-Typ und Availability Zone nicht unterstützt.

Lösung: Versuchen Sie Folgendes, um das Problem zu beheben:

- Warten Sie ein paar Minuten, bis Amazon EC2 Auto Scaling Kapazitäten für diesen Instance-Typ in anderen aktivierten Availability Zones gefunden hat.
- Erweitern Sie Ihre Auto-Scaling-Gruppe auf zusätzliche Availability Zones. Weitere Informationen finden Sie unter Hinzufügen oder Entfernen von Availability Zones.
- Befolgen Sie die bewährte Praxis, eine Vielzahl von Instance-Typen zu verwenden, damit Sie nicht von einem bestimmten Instance-Typ abhängig sind. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.

Die angefragte Reservierung ist nicht ausreichend kompatibel und hat nicht genügend freie Kapazität für diese Anfrage. Die EC2-Instance konnte nicht gestartet werden.

Ursache 1: Sie haben das Limit für die Anzahl der Instances erreicht, die Sie mit einer `targeted` On-Demand-Kapazitätsreservierung starten können.

Lösung 1: Erhöhen Sie entweder die Anzahl der Instances, die Sie mit der `targeted` On-Demand-Kapazitätsreservierung starten können, oder verwenden Sie eine Kapazitätsreservierungs-Gruppe, sodass alles, was über die reservierte Kapazität hinausgeht, als reguläre On-Demand-Kapazität gestartet wird. Weitere Informationen finden Sie unter Verwenden Sie On-Demand-Kapazitätsreservierungen, um Kapazitäten in bestimmten Availability Zones zu reservieren..

Ursache 2: Sie haben das Limit für die Anzahl der Instances erreicht, die Sie mit einem Kapazitätsblock starten können.

Bei Kapazitätsblöcken sind Sie durch die Menge der ursprünglich gekauften Kapazität eingeschränkt. Wenn die Zahl der Starts höher ist als erwartet und die gesamte verfügbare Kapazität aufgebraucht wird, führt dies dazu, dass Starts fehlschlagen. Beendete Instances durchlaufen einen langwierigen Bereinigungsprozess, bevor sie vollständig beendet werden. Während dieser Zeit können sie nicht wiederverwendet werden. Dies kann auch dazu führen, dass Starts fehlschlagen. Weitere Informationen finden Sie unter Capacity BlocksFür Machine-Learning-Workloads verwenden.

Lösung 2: Versuchen Sie Folgendes, um das Problem zu beheben:

- Behalten Sie die Anfrage unverändert bei. Wenn eine Capacity Block-Instance beendet wird, müssen Sie einige Minuten warten, bis die Instance beendet ist und die Kapazität wieder verfügbar ist. Amazon EC2 Auto Scaling löst automatisch die Startanforderung aus, bis die Kapazität verfügbar ist.
- Stellen Sie sicher, dass Sie genügend Kapazität erwerben, um Ihre Spitzenauslastung bewältigen zu können, damit dieser Fehler nicht häufig auftritt.

Ihre Kapazitätsblock-Reservierung <reservation id> ist noch nicht aktiv. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Der angegebene Kapazitätsblock ist noch nicht aktiv.

Lösung: Folgen Sie dem empfohlenen Ansatz für Kapazitätsblöcke und verwenden Sie die geplante Skalierung. Auf diese Weise können Sie sicherstellen, dass Sie die gewünschte Kapazität Ihrer Auto-Scaling-Gruppe nur dann erhöhen, wenn die Reservierung aktiv ist, und sie verringern, bevor die Reservierung beendet ist.

Es ist keine Spot-Kapazität verfügbar, die Ihrer Anforderung entspricht. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Derzeit ist nicht genügend freie Kapazität vorhanden, um Ihre Anforderung nach Spot-Instances zu erfüllen.

Lösung: Versuchen Sie Folgendes, um das Problem zu beheben:

- Warten Sie einige Minuten; die Kapazität kann häufig wechseln. Amazon EC2 Auto Scaling löst automatisch die Startanforderung aus, bis die Kapazität verfügbar ist.
- Erweitern Sie Ihre Auto-Scaling-Gruppe auf zusätzliche Availability Zones. Weitere Informationen finden Sie unter Hinzufügen oder Entfernen von Availability Zones.
- Befolgen Sie die bewährte Praxis, eine Vielzahl von Instance-Typen zu verwenden, damit Sie nicht von einem bestimmten Instance-Typ abhängig sind. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen.

<number of instances> Instance wird/Instances werden bereits ausgeführt. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Sie haben das Limit der Anzahl der Instances, die Sie in einer Region starten können, erreicht. Wenn Sie Ihr AWS Konto erstellen, legen wir Standardlimits für die Anzahl der Instances fest, die Sie pro Region ausführen können.

Lösung: Versuchen Sie Folgendes, um das Problem zu beheben:

- Wenn Ihre aktuellen Limits Ihren Bedürfnissen nicht entsprechen, können Sie eine Kontingenterhöhung auf regionaler Basis anfordern. Weitere Informationen finden Sie unter Amazon EC2-Servicekontingente im Amazon EC2 EC2-Benutzerhandbuch.
- Senden Sie eine neue Anforderung mit einer geringeren Anzahl von Instances (die Sie später erhöhen können).

Fehlersuche bei Amazon EC2 Auto Scaling: AMI-Probleme

Auf dieser Seite finden Sie Informationen zu AMI-Problemen, mögliche Ursachen und Maßnahmen, die Sie zum Lösen der Probleme ergreifen können.

Wie Sie eine Fehlermeldung abrufen, erfahren Sie unter Abrufen einer Fehlermeldung aus Skalierungen.

Wenn Ihre EC2-Instances aufgrund von AMI-Problemen nicht gestartet werden können, erscheinen eine oder mehrere der folgenden Fehlermeldungen.

AMI-Probleme

- Die AMI-ID <ID of your AMI> existiert nicht. Die EC2-Instance konnte nicht gestartet werden.
- Das AMI <AMI-ID> hat den Status "Schwebend" und kann nicht ausgeführt werden. Die EC2-Instance konnte nicht gestartet werden.
- Ungültiger Gerätename <device name>. Die EC2-Instance konnte nicht gestartet werden.
- Die Architektur 'arm64' des angegebenen Instance-Typs entspricht nicht der Architektur 'x86_64' des angegebenen AMI... Das Starten der EC2-Instance ist fehlgeschlagen.
- AMI '<AMI ID>' ist deaktiviert und kann nicht ausgeführt werden. Die EC2-Instance konnte nicht gestartet werden.

> **Important**
>
> AWS unterstützt die private gemeinsame Nutzung eines AMI mit einem anderen AWS Konto, indem die AMI-Berechtigungen geändert werden. Wenn ein AMI privat geschaltet wird, ohne gemeinsam genutzt zu werden, kann dies zu einem Autorisierungsfehler beim Starten neuer Instances führen. Weitere Informationen zum Teilen von privaten AMIs finden Sie unter Teilen eines AMI mit bestimmten AWS Konten im Amazon EC2 EC2-Benutzerhandbuch.

Die AMI-ID <ID of your AMI> existiert nicht. Die EC2-Instance konnte nicht gestartet werden.

- Ursache: Das AMI wurde nach dem Erstellen der Startkonfiguration möglicherweise gelöscht.
- Solution (Lösung):
  1. Erstellen Sie eine neue Startvorlage oder Startkonfiguration mit einem gültigen AMI.

2. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit der neuen Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

Das AMI <AMI-ID> hat den Status "Schwebend" und kann nicht ausgeführt werden. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Wenn Sie Ihr AMI eben erst erstellt haben (indem Sie einen Snapshot einer laufenden Instance aufgenommen oder einen anderen Weg gewählt haben), ist es möglicherweise noch nicht verfügbar.

Lösung: Sie müssen mit dem Erstellen Ihrer Startvorlage oder Startkonfiguration warten, bis Ihr AMI verfügbar ist.

Ungültiger Gerätename <device name>. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Wenn Sie ein EBS-Volume an eine EC2-Instance anhängen, müssen Sie einen gültigen Gerätenamen für das Volume angeben. Das ausgewählte AMI muss diesen Gerätenamen unterstützen.

Solution (Lösung):

1. Erstellen Sie eine neue Startvorlage oder Startkonfiguration und geben Sie den richtigen Gerätenamen für Ihr AMI an. Die empfohlene Namenskonvention variiert je nach Virtualisierungstyp des AMI. Weitere Informationen finden Sie unter Gerätenamen im Amazon EC2 EC2-Benutzerhandbuch.
2. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit der neuen Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

Die Architektur 'arm64' des angegebenen Instance-Typs entspricht nicht der Architektur 'x86_64' des angegebenen AMI... Das Starten der EC2-Instance ist fehlgeschlagen.

Ursache 1: Wenn die Architektur des AMI und der in Ihrer Startvorlage oder Startkonfiguration verwendete Instance-Typ nicht übereinstimmen, erhalten Sie einen Fehler, wenn Amazon EC2 Auto Scaling versucht, eine Instance mit der inkompatiblen Instance-Konfiguration zu starten.

Lösung 1:

1. Überprüfen Sie die Architektur Ihres AMI mit dem Befehl describe-images oder von der Amazon EC2-Konsole aus, indem Sie den Wert Architecture im Detailbereich der Seite Amazon Machine Images (AMI) überprüfen.
2. Suchen Sie mit dem Befehl describe-instance-types oder von der Amazon EC2-Konsole aus nach einem Instance-Typ, der dieselbe Architektur wie Ihr AMI hat, indem Sie die Spalte Architektur auf dem Bildschirm Instance-Typen überprüfen. Weitere Informationen zur Auswahl eines kompatiblen Instance-Typs finden Sie unter Kompatibilität bei der Änderung des Instance-Typs im Amazon EC2 EC2-Benutzerhandbuch.
3. Erstellen Sie eine neue Startvorlage oder Startkonfiguration mit einem Instance-Typ, der die gleiche Architektur wie Ihr AMI hat.
4. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit der neuen Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

Ursache 2: Amazon EC2 Auto Scaling versucht, einen Instance-Typ zu starten, der in der Richtlinie für gemischte Instances für Ihre Auto-Scaling-Gruppe angegeben ist, aber der Instance-Typ hat nicht die gleiche Architektur wie das in Ihrer Startvorlage angegebene AMI.

Lösung 1: Nehmen Sie keine Instance-Typen mit unterschiedlichen Architekturen in Ihre Richtlinie für gemischte Instances auf.

1. Überprüfen Sie die Architektur Ihres AMI mit dem Befehl describe-images oder von der Amazon EC2-Konsole aus, indem Sie den Wert Architecture im Detailbereich der Seite Amazon Machine Images (AMI) überprüfen.
2. Überprüfen Sie die Architektur jedes Instance-Typs, den Sie in Ihre Richtlinie für gemischte Instances aufnehmen möchten, mithilfe des Befehls describe-instance-types oder von der Amazon EC2-Konsole aus, indem Sie die Spalte Architektur auf dem Bildschirm Instance-Typen überprüfen. Weitere Informationen zur Auswahl kompatibler Instance-Typen finden Sie unter Kompatibilität bei der Änderung des Instance-Typs im Amazon EC2 EC2-Benutzerhandbuch.
3. Aktualisieren oder entfernen Sie die inkompatiblen Instance-Typen aus Ihrer Auto-Scaling-Gruppe mithilfe des Befehls update-auto-scaling-group.

Lösung 2: Um sowohl Arm- (Graviton2) als auch x86_64- (Intel) Instances in derselben Auto-Scaling-Gruppe zu starten, müssen Sie Startvorlagen verwenden, die von einem Arm-kompatiblen AMI bzw.

einem Intel-x86-kompatiblen AMI unterstützt werden, um den Instance-Typen in Ihrer Richtlinie für gemischte Instances zu entsprechen.

1. Überprüfen Sie die Architektur des AMI in Ihrer vorhandenen Startvorlage mit dem Befehl describe-images oder von der Amazon EC2-Konsole aus, indem Sie den Wert Architecture im Detailbereich der Seite Amazon Machine Images (AMI) überprüfen.
2. Erstellen Sie eine neue Startvorlage mit einem AMI, das der anderen Architektur entspricht, die Sie verwenden möchten.
3. Aktualisieren Sie Ihre Auto-Scaling-Gruppe, um die vorhandene Startvorlage zu überschreiben, und geben Sie die neue Startvorlage für jeden kompatiblen Instance-Typ an, indem Sie den update-auto-scaling-group-Befehl verwenden. Weitere Informationen finden Sie unter Verwenden Sie eine andere Startvorlage für einen Instance-Typ.

AMI '<AMI ID>' ist deaktiviert und kann nicht ausgeführt werden. Die EC2-Instance konnte nicht gestartet werden.

Ursache: Sie versuchen, Instances von einem AMI aus zu starten, das deaktiviert wurde. Weitere Informationen finden Sie unter Deaktivieren eines AMI im Amazon EC2 EC2-Benutzerhandbuch.

Solution (Lösung):

1. Erstellen Sie eine neue Startvorlage oder Startkonfiguration und geben Sie ein AMI an, das nicht deaktiviert ist.
2. Aktualisieren Sie Ihre Auto-Scaling-Gruppe mit der neuen Startvorlage oder Startkonfiguration mithilfe des Befehls update-auto-scaling-group.

Fehlersuche bei Amazon EC2 Auto Scaling: Load Balancer-Probleme

Auf dieser Seite finden Sie Informationen zu Problemen, die vom Load Balancer Ihrer Auto-Scaling-Gruppe verursacht wurden, mögliche Ursachen und Maßnahmen, die Sie zum Lösen der Probleme ergreifen können.

Wie Sie eine Fehlermeldung abrufen, erfahren Sie unter Abrufen einer Fehlermeldung aus Skalierungen.

Wenn Ihre EC2-Instances aufgrund von Problemen mit dem Load Balancer Ihrer Auto-Scaling-Gruppe nicht gestartet werden können, erscheinen eine oder mehrere der folgenden Fehlermeldungen.

Load Balancer-Probleme

- Eine oder mehrere Zielgruppen. Das Validieren der Load Balancer-Konfiguration ist fehlgeschlagen.
- Load Balancer kann nicht gefunden <your load balancer>werden. Das Validieren der Load Balancer-Konfiguration ist fehlgeschlagen.
- Es ist kein AKTIVER Load Balancer namens <Load Balancer-Name> vorhanden. Das Aktualisieren der Load Balancer-Konfiguration ist fehlgeschlagen.
- Die EC2-Instance <instance ID> ist nicht in der VPC. Das Aktualisieren der Load Balancer-Konfiguration ist fehlgeschlagen.

> **Note**
>
> Sie können Reachability Analyzer verwenden, um Verbindungsprobleme zu beheben, indem Sie überprüfen, ob Instances in Ihrer Auto-Scaling-Gruppe über den Load Balancer erreichbar sind. Weitere Informationen zu den verschiedenen Netzwerk-Fehlkonfigurationsproblemen, die von Reachability Analyzer automatisch erkannt werden, finden Sie unter Reachability Analyzer – Erläuterungscodes im Benutzerhandbuch zu Reachability Analyzer.

Eine oder mehrere Zielgruppen. Das Validieren der Load Balancer-Konfiguration ist fehlgeschlagen.

Problem: Wenn Ihre Auto Scaling-Gruppe Instances startet, versucht Amazon EC2 Auto Scaling zu überprüfen, ob die Elastic Load Balancing-Ressourcen, die der Auto Scaling-Gruppe zugeordnet sind, vorhanden sind. Wenn eine Zielgruppe nicht gefunden werden kann, schlägt die Skalierungsaktivität fehl und Sie erhalten den Fehler `One or more target groups not found. Validating load balancer configuration failed.`.

Ursache 1: Eine an Ihre Auto Scaling-Gruppe angehängte Zielgruppe wurde gelöscht.

Lösung 1: Sie können entweder eine neue Auto-Scaling-Gruppe ohne die Zielgruppe erstellen oder die ungenutzte Zielgruppe aus der Auto Scaling-Gruppe mithilfe der Amazon EC2 Auto Scaling-Konsole oder dem Befehl detach-load-Balancer-Zielgruppen entfernen.

Ursache 2: Die Zielgruppe existiert, aber es gab ein Problem beim Versuch, den Zielgruppe n-ARN beim Erstellen der Auto Scaling-Gruppe anzugeben. Ressourcen werden nicht in der richtigen Reihenfolge erstellt.

Lösung 2: Erstellen Sie eine neue Auto-Scaling-Gruppe und geben Sie den Namen des Load Balancers am Ende ein.

Load Balancer kann nicht gefunden <your load balancer>werden. Das Validieren der Load Balancer-Konfiguration ist fehlgeschlagen.

Problem: Wenn Ihre Auto -Gruppe Instances startet, versucht Amazon EC2 Auto Scaling zu überprüfen, ob die Elastic Load Balancing-Ressourcen, die der Auto Scaling-Gruppe zugeordnet sind, vorhanden sind. Wenn ein Classic Load Balancer nicht gefunden werden kann, schlägt die Skalierungsaktivität fehl und Sie erhalten den Fehler `Cannot find Load Balancer <your load balancer>. Validating load balancer configuration failed.`.

Ursache 1: Der Classic Load Balancer wurde gelöscht.

Lösung 1: Sie können entweder eine neue Auto Scaling-Gruppe ohne den Load Balancer erstellen oder den nicht verwendeten Load Balancer aus der Auto Scaling-Gruppe entfernen, indem Sie die Amazon EC2 Auto Scaling-Konsole oder dem Befehl detach-load-Balancer-Zielgruppen entfernen.

Ursache 2: Der Classic Load Balancer existiert, aber es gab ein Problem bei der Angabe des Load Balancer-Namens bei der Erstellung der Auto Scaling-Gruppe. Ressourcen werden nicht in der richtigen Reihenfolge erstellt.

Lösung 2: Erstellen Sie eine neue Auto-Scaling-Gruppe und geben Sie den Namen des Load Balancers am Ende ein.

Es ist kein AKTIVER Load Balancer namens <Load Balancer-Name> vorhanden. Das Aktualisieren der Load Balancer-Konfiguration ist fehlgeschlagen.

Ursache: Der angegebene Load Balancer wurde möglicherweise gelöscht.

Lösung: Sie können entweder einen neuen Load Balancer und dann eine neue Auto-Scaling-Gruppe erstellen oder eine neue Auto-Scaling-Gruppe ohne den Load Balancer einrichten.

Die EC2-Instance <instance ID> ist nicht in der VPC. Das Aktualisieren der Load Balancer-Konfiguration ist fehlgeschlagen.

Ursache: Die angegebene Instance befindet sich nicht in der VPC.

Lösung: Sie können entweder den Load Balancer der Instance löschen oder eine neue Auto-Scaling-Gruppe erstellen.

Fehlersuche bei Amazon EC2 Auto Scaling: Startvorlagen

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die beim Erstellversuch einer Startvorlage für Ihre Auto-Scaling-Gruppe auftreten könnten.

Instances können nicht gestartet werden

Wenn Sie keine Instances mit einer bereits angegebenen Startvorlage starten können, überprüfen Sie die folgenden Hinweise zur allgemeinen Problembehandlung: Fehlersuche bei Amazon EC2 Auto Scaling: Startfehler von EC2-Instance.

Sie müssen eine gültige, vollständig formatierte Startvorlage verwenden (ungültiger Wert)

Problem: Wenn Sie versuchen, eine Startvorlage für eine Auto-Scaling-Gruppe anzugeben, erhalten Sie den `You must use a valid fully-formed launch template`-Fehler. Möglicherweise tritt dieser Fehler auf, da die Werte in der Startvorlage nur überprüft werden, wenn eine Auto-Scaling-Gruppe erstellt oder aktualisiert wird, welche die Startvorlage verwendet.

Ursache 1: Wenn der `You must use a valid fully-formed launch template`-Fehler auftritt, dann gibt es Probleme, die Amazon EC2 Auto Scaling veranlassen, etwas über die Startvorlage als ungültig zu betrachten. Das ist ein generischer Fehler, der verschiedene Ursachen haben kann.

Lösung 1: Versuchen Sie die folgenden Schritte zur Fehlerbehebung:

1. Beachten Sie den zweiten Teil der Fehlermeldung, um weitere Informationen zu erhalten. Im Anschluss an den `You must use a valid fully-formed launch template`-Fehler finden Sie eine spezifischere Fehlermeldung, die das Problem identifiziert, das Sie beheben müssen.

2. Wenn Sie die Ursache nicht finden können, testen Sie Ihre Startvorlage mit dem run-instances-Befehl. Nutzen Sie die Option `--dry-run` wie im folgenden Beispiel So können Sie das Problem reproduzieren und Einblicke in seine Ursache erhalten.

```
aws ec2 run-instances --launch-template LaunchTemplateName=my-template,Version='1' --
dry-run
```

3. Wenn ein Wert nicht gültig ist, stellen Sie sicher, dass die angegebene Ressource vorhanden ist und dass sie korrekt ist. Wenn Sie beispielsweise ein Amazon EC2-Schlüsselpaar bestimmen, muss die Ressource im Konto und in der Region vorhanden sein, in der Sie die Auto-Scaling-Gruppe erstellen oder aktualisieren.
4. Wenn erwartete Informationen fehlen, überprüfen Sie Ihre Einstellungen und passen Sie die Startvorlage nach Bedarf an.
5. Nachdem Sie Ihre Änderungen vorgenommen haben, führen Sie den run-instances-Befehl mit der `--dry-run`-Option aus, um zu überprüfen, ob Ihre Startvorlage gültige Werte verwendet.

Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe.

Sie sind nicht berechtigt, die Startvorlage zu verwenden (unzureichende Berechtigungen)

Problem: Wenn Sie versuchen, eine Startvorlage für eine Auto-Scaling-Gruppe anzugeben, erhalten Sie den `You are not authorized to use launch template`-Fehler.

Ursache 1: Wenn Sie versuchen, eine Startvorlage zu verwenden und die von Ihnen verwendeten IAM-Anmeldeinformationen nicht über ausreichende Berechtigungen verfügen, wird in einer Fehlermeldung darauf hingewiesen, dass Sie nicht zur Verwendung der Startvorlage berechtigt sind.

Lösung 1: Um das Problem zu lösen, versuchen Sie Folgendes:

- Vergewissern Sie sich, dass die IAM-Anmeldeinformationen, die Sie für die Anfrage verwenden, über die Berechtigung zum Aufrufen der benötigten EC2-API-Aktionen verfügen, einschließlich der Aktion `ec2:RunInstances`. Wenn Sie Tags in Ihrer Startvorlage angegeben haben, müssen Sie auch über die Berechtigung verfügen, die `ec2:CreateTags`-Aktion zu verwenden.
- Alternativ können Sie auch überprüfen, ob den IAM-Anmeldeinformationen, die Sie für die Anfrage verwenden, die `AmazonEC2FullAccess`-Richtlinie zugewiesen ist. Diese AWS verwaltete Richtlinie gewährt vollen Zugriff auf alle Amazon EC2-Ressourcen und zugehörigen Services, einschließlich Amazon EC2 Auto Scaling und Elastic Load Balancing. CloudWatch

Weitere Informationen zu den für die Verwendung von Startvorlagen erforderlichen Berechtigungen, einschließlich beispielhafter IAM-Richtlinien, finden Sie unter Steuern des Zugriffs auf Startvorlagen mit IAM-Berechtigungen im Amazon EC2 EC2-Benutzerhandbuch. Weitere Beispiele für IAM-Richtlinien finden Sie unter Support für Startvorlagen.

Ursache 2: Wenn Sie versuchen, eine Startvorlage zu verwenden, die ein Instance-Profil angibt, müssen Sie über die IAM-Berechtigung verfügen, die dem Instance-Profil zugeordnete IAM-Rolle zu übergeben.

Lösung 2: Stellen Sie sicher, dass die IAM-Anmeldeinformationen, die Sie zum Erstellen der Anforderung verwenden, über die korrekte `iam:PassRole`-Berechtigung verfügt, um die angegebene Rolle an den Amazon-EC2-Auto-Scaling-Service zu übergeben. Weitere Informationen und eine IAM-Beispielrichtlinie finden Sie unter IAM-Rollen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden. Weitere Themen zur Problembehandlung im Zusammenhang mit Instance-Profilen finden Sie unter Problembehandlung bei Amazon EC2 und IAM im IAM-Benutzerhandbuch.

Ursache 3: Wenn Sie versuchen, eine Startvorlage zu verwenden, die ein AMI in einem anderen angibt AWS-Konto, und das AMI privat ist und nicht mit dem von AWS-Konto Ihnen verwendeten geteilt wird, erhalten Sie die Fehlermeldung, dass Sie nicht berechtigt sind, die Startvorlage zu verwenden.

Lösung 3: Stellen Sie sicher, dass die Berechtigungen für das AMI das von Ihnen verwendete Konto beinhalten. Weitere Informationen finden Sie unter Ein AMI mit bestimmten Personen teilen AWS-Konten im Amazon EC2 EC2-Benutzerhandbuch.

Ähnliche Informationen

Die folgenden verwandten Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit diesem Service.

Ressource	Beschreibung
Amazon EC2 Auto Scaling API-Referenz	Die Dokumentation für jeden API-Vorgang zeigt die Anforderungsparameter und die XML-Antwort und enthält Links zu sprachspezifischen SDK-Referenzthemen.
Auto Scaling in der AWS CLI -Befehlsreferenz	Beschreibungen der AWS CLI Befehle, die Sie für die Arbeit mit Auto Scaling Scaling-Gruppen verwenden können.
AWS Tools for PowerShell Cmdlet-Referenz	Mit den AWS Tools für PowerShell können Sie über die PowerShell Befehlszeile Skripts für Operationen auf Ihren AWS Ressourcen erstellen.
Erstellen von Auto-Scaling-Gruppen mit AWS CloudFormation	Mit der AWS::AutoScaling::AutoScaling können Sie Ihre Auto Scaling Scaling-Gruppen ohne manuelle Aktionen erstellen, modellieren und verwalten.
Endpunkte und Kontingente für Amazon EC2 Auto Scaling in Allgemeine AWS-Referenz	Informationen zu Amazon EC2 Auto Scaling-Regionen und -Endpunkten
Produktseite	Die Hauptwebsite für Informationen zu Amazon EC2 Auto Scaling.
AWS Re:POST	AWS verwalteter Frage-und-Antwort-Service (Q & A), der von Experten geprüfte Antworten auf Ihre technischen Fragen per Crowdsourcing bietet.

Ressource	Beschreibung
Erstellen Sie ein AMI im Amazon EC2 EC2-Benutzerhandbuch	Erfahren Sie, wie sie ein Amazon Machine Image (AMI) aus Ihrer benutzerdefinierten Instance erstellen.
So stellen Sie im Amazon EC2 EC2-Benutzerhandbuch eine Verbindung zu Ihrer Linux-Instance her	Erfahren Sie, wie Sie eine Verbindung zu den Linux-Instances herstellen, die Sie starten.
So stellen Sie im Amazon EC2 EC2-Benutzerhandbuch eine Verbindung zu Ihrer Windows-Instance her	Erfahren Sie, wie Sie eine Verbindung zu den Windows-Instances herstellen, die Sie starten.
Einen Abrechnungsalarm zur Überwachung Ihrer geschätzten AWS Gebühren im CloudWatch Amazon-Benutzerhandbuch erstellen	Erfahren Sie, wie Sie Ihre geschätzten Gebühren mithilfe von überwachen können CloudWatch.
Benutzerhandbuch zum Application Auto Scaling	Erfahren Sie, wie Sie Auto Scaling für skalierbare Ressourcen für Amazon Web Services über Amazon EC2 hinaus konfigurieren.

Die folgenden allgemeinen Ressourcen helfen Ihnen dabei, mehr darüber AWS zu erfahren.

- Kurse und Workshops — Links zu rollen- und Spezialkursen sowie zu Übungen zum Selbststudium, mit denen Sie Ihre AWS Fähigkeiten verbessern und praktische Erfahrungen sammeln können.
- AWS Developer Center — Erkunden Sie Tutorials, laden Sie Tools herunter und erfahren Sie mehr über Veranstaltungen für Entwickler. AWS
- AWS Entwicklertools — Links zu Entwicklertools, SDKs, IDE-Toolkits und Befehlszeilentools für die Entwicklung und Verwaltung von AWS Anwendungen.
- Ressourcencenter für die ersten Schritte — Erfahren Sie, wie Sie Ihre AWS-Konto Anwendung einrichten, der AWS Community beitreten und Ihre erste Anwendung starten.
- Praktische Tutorials — Folgen Sie den step-by-step Tutorials, um Ihre erste Anwendung zu starten. AWS

- AWS Whitepapers — Links zu einer umfassenden Liste von technischen AWS Whitepapers zu Themen wie Architektur, Sicherheit und Wirtschaft, die von Solutions Architects oder anderen technischen Experten verfasst wurden. AWS
- AWS Support Center — Die zentrale Anlaufstelle für die Erstellung und Verwaltung Ihrer Fälle. AWS Support Enthält auch Links zu anderen hilfreichen Ressourcen wie Foren, häufig gestellten technischen Fragen, dem Status des Dienstes und AWS Trusted Advisor.
- AWS Support— Die wichtigste Webseite mit Informationen über AWS Support einen Support-Kanal mit schnellen Reaktionszeiten one-on-one, der Sie bei der Entwicklung und Ausführung von Anwendungen in der Cloud unterstützt.
- Kontakt – Zentraler Kontaktpunkt für Fragen zu AWS -Abrechnung, Konten, Ereignissen Missbrauch und anderen Problemen.
- AWS Nutzungsbedingungen der Website — Detaillierte Informationen zu unseren Urheberrechten und Marken, zu Ihrem Konto, Ihrer Lizenz und Ihrem Zugriff auf die Website sowie zu anderen Themen.

Dokumentverlauf

In der folgenden Tabelle sind wichtige Ergänzungen zur Amazon EC2 Auto Scaling-Dokumentation ab Juli 2018 enthalten. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie den RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Sicherheits-IAM-Update	Die AutoScalingService RolePolicyverwaltete Richtlinie gewährt Amazon EC2 (`ec2:GetSecurityGroupsForVpc` und`ec2:GetInstanceTypesFromInstanceRequirements`) jetzt zusätzliche Berechtigungen.	29. Februar 2024
Der Ruhezustand im warmen Pool wird zusätzlich unterstützt AWS-Regionen	Sie können jetzt Instances in einem warmen Pool in zwei weiteren Regionen in den Ruhezustand versetzen: AWS GovCloud (US-Ost) und (US-West). AWS GovCloud Weitere Informationen zu Warm-Pools finden Sie im Amazon EC2 Auto Scaling-Benutzerhandbuch unter Warm-Pools für Amazon EC2 Auto Scaling.	26. Februar 2024
Der Winterschlaf im warmen Pool wird zusätzlich unterstützt AWS-Regionen	Sie können jetzt Instances in einem warmen Pool in zwei weiteren Regionen in den Ruhezustand versetzen: Europa (Zürich) und Naher Osten (VAE). Weitere	21. Februar 2024

	Informationen zu Warm-Pools finden Sie im Amazon EC2 Auto Scaling-Benutzerhandbuch unter Warm-Pools für Amazon EC2 Auto Scaling.	
Support für die kontoüber greifende Verwendung von Parametern	Sie können jetzt einen von einem anderen gemeinsam genutzten AWS Systems Manager Parameter AWS-Konto mit Amazon EC2 Auto Scaling verwenden. Weitere Informationen finden Sie unter Verwenden von AWS Systems Manager Parametern anstelle von AMI-IDs in Startvorlagen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.	21. Februar 2024
Neue Option zum Schutz vor Spot-Preisen	Sie können jetzt Ihren Schwellenwert für den Preisschutz für Spot-Instances als Prozentsatz eines On-Demand-Preises definieren, wenn Sie die attributbasierte Instance-Typauswahl verwenden. Weitere Informationen finden Sie unter Preisschutz im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.	29. Januar 2024

Wartungsrichtlinien für Instances	Sie können jetzt eine Wartungsrichtlinie für Instances verwenden, um festzulegen, ob Instances vor oder nach der Beendigung vorhandener Instances bei Ereignissen gestartet werden, die eine Ersetzung Ihrer Instances erfordern, einschließlich einer Instance-Aktualisierung. Weitere Informationen finden Sie unter Wartungsrichtlinie für Instances im Benutzerhandbuch für Amazon EC2 Auto Scaling.	15. November 2023
Kapazitätsblöcke für ML	Sie können jetzt Instances in einem Kapazitätsblock starten, indem Sie die Kapazitätsblockreservierungs-ID angeben, wenn Sie eine Startvorlage erstellen. Mit Kapazitätsblöcken können Sie GPU-Instances für ein späteres Datum reservieren, um kurzfristige Workloads für Machine Learning (ML) zu unterstützen. Weitere Informationen finden Sie unter Verwenden von Kapazitätsblöcken für Machine-Learning-Workloads im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.	31. Oktober 2023

Neue Funktionen zur Instance-Aktualisierung	Sie können jetzt eine Instance-Aktualisierung so konfigurieren, dass ihr Status auf „Fehlgeschlagen“ gesetzt wird und optional ein Rollback durchgeführt wird, wenn festgestellt wird, dass ein bestimmter CloudWatch Alarm in den `ALARM` Status übergegangen ist. Weitere Informationen finden Sie unter Änderungen mit einem Rollback rückgängig machen im Benutzerhandbuch für Amazon EC2 Auto Scaling.	31. Juli 2023
Änderungen im Handbuch	Ein neues Thema über den Start von On-Demand-Instances in Kapazitätsreservierungen wurde dem Handbuch hinzugefügt. Weitere Informationen finden Sie unter Verwenden von On-Demand-Kapazitätsreservierungen, um Kapazitäten in spezifischen Availability Zones zu reservieren im Amazon EC2 Auto Scaling-Benutzerhandbuch.	28. Juli 2023

Änderungen im Handbuch	Der Anleitung wurde ein neues Thema zur Migration Ihrer AWS CloudFormation Stacks von Startkonfigurationen zu Startvorlagen hinzugefügt. Weitere Informationen erhalten Sie unter Migration von AWS CloudFormation -Stacks von Startkonfigurationen zu Startvorlagen im Benutzerhandbuch zum Amazon EC2 Auto Scaling.	18. April 2023

Support für neue API-Betriebe	Diese Version fügt mit `AttachTrafficSources` , `DetachTrafficSources` und `DescribeTrafficSources` drei neue API-Operationen hinzu. Auch ein neues Feld, `TrafficSources` , wurde den Ergebnissen der `DescribeAutoScalingGroups` -Operationen hinzugefügt. Ein neues Aktivitätsstatus, `WaitingForConnectionDraining` , wurde den Ergebnissen der `DescribeScalingActivities` -Operationen hinzugefügt. Amazon EC2 Auto Scaling unterstützt auch `VPC_LATTICE` , einen neuen Wert für das `HealthCheckType` -Feld in `CreateAutoScalingGroup` -, `UpdateAutoScalingGroup` - und `DescribeAutoScalingGroups` -Operationen. Weitere Informationen finden Sie in der Amazon EC2 Auto Scaling-API-Referenz.	31. März 2023

Support für Amazon VPC Lattice	Dies ist die allgemein verfügbare Version von VPC Lattice für Amazon EC2 Auto Scaling. Weitere Informationen finden Sie unter Weiterleitung des Datenverkehrs zu Ihrer Auto-Scaling-Gruppe mit einer VPC-Lattice-Zielgruppe im Benutzerhandbuch zu Amazon EC2 Auto Scaling.	31. März 2023
Änderungen im Handbuch	Der Abschnitt mit AWS CLI Beispielen für die Arbeit mit Elastic Load Balancing enthält jetzt neue und aktualisierte Beispiele. Weitere Informationen finden Sie unter Beispiele für die Arbeit mit Elastic Load Balancing with the AWS Command Line Interface (AWS CLI) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.	31. März 2023
Zusätzliche Support für vorausschauende Skalierung AWS-Regionen	Sie können jetzt Richtlinien für vorausschauende Skalierung in den Regionen Naher Osten (VAE) und AWS GovCloud (USA-Ost) erstellen. Weitere Informationen finden Sie unter Prädiktive Skalierung von Cooldowns für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	16. März 2023

Neue Funktionen zur Instance-Aktualisierung	Sie können jetzt Instances im Standby-Modus beenden oder ignorieren und Instances ersetzen oder ignorieren, die vor dem Abskalieren geschützt sind, anstatt darauf zu warten, dass sie austauschbar werden. Sie können auch Änderungen nach einer fehlgeschlagenen Instance-Aktualisierung zurücksetzen. Im Rahmen dieser Aktualisierung wurde die Dokumentation um Themen zum Zurücksetzen einer Instance-Aktualisierung, zum Abbrechen einer Instance-Aktualisierung und zum Verständnis der Standardwerte für die konfigurierbaren Parameter einer Instance-Aktualisierung erweitert. Weitere Informationen finden Sie unter Ersetzen von Auto Scaling-Instances basierend auf einer Instance-Aktualisierung im Benutzerhandbuch zum Amazon EC2 Auto Scaling.	10. Februar 2023

Support für die Verwendung eines AWS Systems Manager Parameters für eine AMI-ID	Sie können jetzt einen Systems-Manager-Parameter anstelle einer AMI-ID in Ihrer Startvorlage verwenden. Weitere Informationen finden Sie unter Verwendung von AWS Systems Manager -Parametern anstelle von AMI-IDs in Startvorlagen im Benutzerhandbuch für Amazon EC2 Auto Scaling.	19. Januar 2023
Empfehlungen für prädiktive Skalierung	Sie können jetzt über die Konsole von Amazon EC2 Auto Scaling Empfehlungen für die Auswertung und Auswahl der richtigen Richtlinie für prädiktive Skalierung erhalten. Weitere Informationen finden Sie unter Auswerten Ihrer Richtlinie für prädiktive Skalierung im Benutzerhandbuch für Amazon EC2 Auto Scaling.	18. Januar 2023
Prädiktive Skalierung von Prognosen	Die durch prädiktive Skalierung generierten Prognosen werden jetzt alle sechs Stunden statt täglich aktualisiert. Weitere Informationen finden Sie unter Prädiktive Skalierung von Cooldowns für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	06. Januar 2023

Support für CloudWatch metrische Mathematik	Sie können jetzt Metrikberechnungen verwenden, wenn Sie Skalierungsrichtlinien für die Zielverfolgung erstellen. Mit metrischer Mathematik können Sie mehrere CloudWatch Metriken abfragen und mathematische Ausdrücke verwenden, um neue Zeitreihen auf der Grundlage dieser Metriken zu erstellen. Weitere Informationen finden Sie unter Erstellen einer Zielnachverfolgungs-Skalierungsrichtlinie für Amazon EC2 Auto Scaling mit Metrikberechnungen im Benutzerhandbuch für Amazon EC2 Auto Scaling.	08. Dezember 2022
Aktualisieren auf Berechtigungen für serviceverknüpfte IAM-Rollen	Die `AutoScalingServiceRolePolicy` -Richtlinie gewährt Amazon EC2 Auto Scaling jetzt zusätzliche Berechtigungen. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	6. Dezember 2022

Neue Spot-Zuweisungsstrategie	Sie können nun die preis- und kapazitätsoptimierte Zuweisungsstrategie verwenden, um Spot Instances aus den Spot-Pools anzufordern, bei denen die Wahrscheinlichkeit einer Unterbrechung am geringsten ist und die den niedrigsten Preis haben. Weitere Informationen finden Sie unter Zuweisungsstrategien im Amazon EC2 Auto Scaling-Benutzerhandbuch.	10. November 2022
Support für die prädiktive Skalierung in der Region Asien-Pazifik (Jakarta)	Richtlinien für die prädiktive Skalierung können jetzt in der Region Asien-Pazifik (Jakarta) erstellt werden. Weitere Informationen finden Sie unter Prädiktive Skalierung von Cooldowns für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	13. Oktober 2022

Support benutzerdefinierte r Metriken für prädiktive Skalierung in der Konsole	Sie können jetzt benutzerd efinierte Metrikdaten verwenden, wenn Sie prädiktiv e Scaling-Richtlinien über die Amazon EC2 Auto Scaling-Konsole erstellen. Weitere Informationen finden Sie unter Prädiktive Skalierung von Cooldowns für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerha ndbuch.	13. Oktober 2022
CloudWatch Überwachung für prädiktive Skalierungsmetriken	Sie können jetzt mithilfe von auf Überwachungsdaten für die prädiktive Skalierun g zugreifen. CloudWatch Dadurch können Sie Metrikber echnungen nutzen, um neue Zeitreihen zu erstellen , die die Genauigkeit von Prognosedaten anzeigen. Weitere Informationen finden Sie unter Überwachen von Predictive Scaling-Metriken mit CloudWatch im Amazon EC2 Auto Scaling Scaling-B enutzerhandbuch.	7. Juli 2022

Support für die prädiktive Skalierung in der Region Asien-Pazifik (Osaka)	Richtlinien für die prädiktive Skalierung können jetzt in der Region Asien-Pazifik (Osaka) erstellt werden. Weitere Informationen finden Sie unter Prädiktive Skalierung von Cooldowns für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	6. Juli 2022
Unterstützung des Warm-Pool-Ruhezustands in weiteren Regionen	Instances können jetzt in vier weiteren Regionen in den Warm-Pool-Ruhezustand versetzt werden: Afrika (Kapstadt), Asien-Pazifik (Jakarta), Asien-Pazifik (Osaka) und Europa (Mailand). Weitere Informationen zu Warm-Pools finden Sie im Amazon EC2 Auto Scaling-Benutzerhandbuch unter Warm-Pools für Amazon EC2 Auto Scaling.	5. Juli 2022

Update auf Zustandsprüfungen	Bei der Durchführung von Zustandsprüfungen hilft Ihnen Amazon EC2 Auto Scaling jetzt dabei, Ausfallzeiten zu minimieren, die aufgrund vorübergehender Probleme oder falsch konfigurierter Zustandsprüfungen auftreten können. Weitere Informationen finden Sie unter Wie Amazon EC2 Auto Scaling Ausfallzeiten minimiert im Benutzerhandbuch zu Amazon EC2 Auto Scaling.	21. Mai 2022
Standardmäßige Instance-Vorbereitung	Sie können jetzt alle Warmup- und Cooldown-Einstellungen für eine Auto Scaling-Gruppe vereinheitlichen und die Leistung von Skalierungsrichtlinien optimieren, die kontinuierlich skalieren, indem Sie das standardmäßige Aufwärmen der Instanz aktivieren. Weitere Informationen finden Sie unter Festlegen der standardmäßigen Instance-Vorbereitung für eine Auto-Scaling-Gruppe im Amazon EC2 Auto Scaling-Benutzerhandbuch.	19. April 2022

Änderungen im Handbuch	Dem Leitfaden wurde ein neues Kapitel über die Integration mit anderen AWS Diensten hinzugefügt. Weitere Informationen finden Sie unter In Amazon EC2 Auto Scaling integrierte AWS -Dienste im Benutzerhandbuch zum Amazon EC2 Auto Scaling.	29. März 2022
Aktualisieren auf Berechtigungen für serviceverknüpfte IAM-Rollen	Die `AutoScalingServiceRolePolicy` -Richtlinie gewährt Amazon EC2 Auto Scaling jetzt zusätzliche Leseberechtigungen. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien für Amazon EC2 Auto Scaling im Benutzerhandbuch zu Amazon EC2 Auto Scaling.	28. März 2022
Instance-Metadaten liefern den Ziellebenszyklusstatus	Sie können den Zielleben szyklusstatus einer Auto-Scaling-Instance aus den Instance-Metadaten abrufen. Weitere Informationen finden Sie unter Abrufen des Zielleben szyklusstatus durch Instance-Metadaten im Benutzerhandbuch für Amazon EC2 Auto Scaling.	24. März 2022

Support für neue Warmpool-Funktionalität	Sie können Instances jetzt in einem Warm Pool in den Ruhezustand versetzen, um Instances zu stoppen, ohne ihren Speicherinhalt (RAM) zu löschen. Sie können Instances jetzt auch beim Abskalieren in den Warm Pool zurückgeben, anstatt immer Instance-Kapazität zu beenden, die Sie später benötigen werden. Weitere Informationen finden Sie unter Warm-Pools für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	24. Februar 2022
Änderungen im Handbuch	Die Konsole von Amazon EC2 Auto Scaling wurde mit zusätzlichen Optionen aktualisiert, die Sie beim Starten einer Instance-Aktualisierung unterstützen, bei der Überspringen aktiviert und eine gewünschte Konfiguration angegeben ist. Weitere Informationen finden Sie unter Starten oder Abbrechen einer Instance-Aktualisierung (Konsole) im Benutzerhandbuch für Amazon EC2 Auto Scaling.	3. Februar 2022

Benutzerdefinierte Metriken für prädiktive Skalierungsrichtlinien	Sie können jetzt wählen, ob Sie benutzerdefinierte Metriken verwenden möchten, wenn Sie prädiktive Skalierungsrichtlinien erstellen. Sie können auch die Metrikmathematik verwenden, um die Metriken, die Sie in Ihre Richtlinie aufnehmen, weiter anzupassen. Weitere Informationen finden Sie unter Erweiterte prädiktive Skalierungsrichtlinienkonfigurationen mit benutzerdefinierten Metriken.	24. November 2021
Neue On-Demand-Zuweisungsstrategie	Sie können jetzt wählen, ob On-Demand-Instances auf der Grundlage des Preises gestartet werden sollen (der günstigste Instance-Typ zuerst), wenn Sie eine Auto-Scaling-Gruppe erstellen, die eine Richtlinie für gemischte Instances verwendet. Weitere Informationen finden Sie unter Zuweisungsstrategien im Amazon EC2 Auto Scaling-Benutzerhandbuch.	27. Oktober 2021

Attributbasierte Auswahl des Instance-Typs	Amazon EC2 Auto Scaling bietet Unterstützung für die attributbasierte Auswahl des Instance-Typs. Anstatt die Instance-Typen manuell auszuwählen, können Sie Ihre Instance-Anforderungen als eine Reihe von Attributen ausdrücken, wie z.B. vCPU, Arbeitsspeicher und Speicher. Weitere Informationen finden Sie unter Erstellen einer Auto-Scaling-Gruppe mit attributbasierter Auswahl des Instance-Typs im Amazon EC2 Auto Scaling-Benutzerhandbuch.	27. Oktober 2021
Unterstützung für das Filtern von Gruppen nach Tags	Sie können Ihre Auto-Scaling-Gruppen jetzt mit Hilfe von Tag-Filtern filtern, wenn Sie mit dem Befehl `describe-auto-scaling-groups` Informationen über Ihre Auto-Scaling-Gruppen abrufen. Weitere Informationen finden Sie unter Verwenden von Tags zum Filtern von Auto-Scaling-Gruppen im Amazon EC2 Auto Scaling-Benutzerhandbuch.	14. Oktober 2021

Änderungen im Handbuch	Die Amazon EC2 Auto Scaling Scaling-Konsole wurde aktualisiert, damit Sie benutzerdefinierte Kündigungsrichtlinien mit AWS Lambda erstellen können. Die Dokumentation der Konsole wurde entsprechend überarbeitet. Weitere Informationen finden Sie unter Verwendung verschiedener Beendigungsrichtlinien (Konsole).	14. Oktober 2021
Support für das Kopieren von Startkonfigurationen zu Startvorlagen	Sie können jetzt alle Startkonfigurationen in einer AWS Region von der Amazon EC2 Auto Scaling Scaling-Konsole aus in neue Startvorlagen kopieren. Weitere Informationen erhalten Sie unter Startkonfigurationen zu Startvorlagen kopieren im Benutzerhandbuch zum Amazon EC2 Auto Scaling.	9. August 2021

Erweitert die Funktionalität der Instance-Aktualisierung	Beim Ersetzen von Instances können Sie jetzt Aktualisierungen, z. B. eine neue Version einer Startvorlage, einschließen, indem Sie die gewünschte Konfiguration zum `start-instance-refresh` -Befehl hinzufügen. Sie können auch das Ersetzen von Instances überspringen, die bereits über die gewünschte Konfiguration verfügen, indem Sie das Überspringen des Abgleichs aktivieren. Weitere Informationen finden Sie unter Ersetzen von Auto Scaling-Instances basierend auf einer Instance-Aktualisierung im Benutzerhandbuch zum Amazon EC2 Auto Scaling.	05. August 2021
Support für benutzerdefinierte Beendigungsrichtlinien	Sie können jetzt benutzerdefinierte Kündigungsrichtlinien mit AWS Lambda erstellen. Weitere Informationen finden Sie unter Erstellen einer benutzerdefinierten Beendigungsrichtlinie mit Lambda. Die Dokumentation zum Angeben von Beendigungsrichtlinien wurde entsprechend aktualisiert.	29. Juli 2021

Änderungen im Handbuch	Die Amazon EC2 Auto Scaling-Konsole wurde aktualisiert und um zusätzliche Funktionen erweitert, die Sie bei der Erstellung von geplanten Aktionen mit einer angegebenen Zeitzone unterstützen. Die Dokumentation für Geplante Skalierung wurde entsprechend überarbeitet.	3. Juni 2021
gp3-Volumes in Startkonfigurationen	Sie können jetzt gp3-Volumes in den Blockgerätezuordnungen für Startkonfigurationen angeben.	2. Juni 2021
Support für prädiktive Skalierung	Sie können jetzt mit prädiktiver Skalierung Ihre Amazon-EC2-Auto-Scaling-Gruppen mithilfe einer Skalierungsrichtlinie proaktiv skalieren. Weitere Informationen finden Sie unter Prädiktive Skalierung von Cooldowns für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch. Mit diesem Update beinhaltet die AutoScalingServiceRolePolicyverwaltete Richtlinie jetzt die Erlaubnis, die `cloudwatch:GetMetricData` API-Aktion aufzurufen.	19. Mai 2021

Änderungen im Handbuch	Sie können jetzt von auf Beispielvorlagen für Lifecycle -Hooks zugreifen GitHub. Weitere Informationen finden Sie unter Lebenszyklus-Hooks für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	9. April 2021
Support für Warm-Pools	Sie können jetzt die Leistung (Kaltstart minimieren) und die Kosten (Überbereitstellung der Instance-Kapazität stoppen) für Anwendungen mit langen ersten Startzeiten ausgleich en, indem Sie Warm Pools zu Auto-Scaling-Gruppen hinzufügen. Weitere Informati onen finden Sie unter Warm-Pools für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	8. April 2021
Support für Checkpoints	Sie können nun Checkpoin ts zu einer Instance-Aktualisi erung hinzufügen, um Instances in Phasen zu ersetzen und Überprüfu ngen für Ihre Instances an bestimmten Punkten durchzuführen. Weitere Informationen finden Sie unter Checkpoints zu einer Instance-Aktualisierung hinzufügen im Amazon EC2 Auto Scaling-B enutzerhandbuch.	18. März 2021

Änderungen im Handbuch	Verbesserte Dokumentation für die Verwendung EventBridge mit Amazon EC2 Auto Scaling Scaling-Ereignissen und Lifecycle-Hooks. Weitere Informationen finden Sie unter Verwenden von Amazon EC2 Auto Scaling mit EventBridge und Tutorial: Einen Lifecycle-Hook konfigurieren, der eine Lambda-Funktion aufruft im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.	18. März 2021
Unterstützung für lokale Zeitzonen	Sie können jetzt wiederkehrende geplante Aktionen in der lokalen Zeitzone erstellen, indem Sie die `--time-zone` -Option dem put-scheduled-update-group-action-Befehl hinzufügen. Wenn Ihre Zeitzone Sommerzeit befolgt, wird die wiederkehrende Aktion automatisch für Sommerzeit angepasst. Weitere Informationen finden Sie unter Geplante Skalierung im Amazon EC2 Auto Scaling-Benutzerhandbuch.	9. März 2021

Erweitert die Funktionalität für Richtlinien für gemischte Instances	Sie können jetzt Instance-Typen für Ihre Spot-Kapazität priorisieren, wenn Sie eine Richtlinie für gemischte Instances verwenden. Amazon EC2 Auto Scaling versucht, die Prioritäten des Instance-Typen auf Best Effort-Basis zu erfüllen, optimiert jedoch zuerst die Kapazität. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen im Amazon EC2 Auto Scaling-Benutzerhandbuch.	8. März 2021
Skalieren von Aktivitäten für gelöschte Gruppen	Sie können jetzt Skalierungsaktivitäten für gelöschte Auto-Scaling-Gruppen anzeigen, indem Sie dem Befehl describe-scaling-activities die `--include-deleted-groups` Option hinzufügen. Weitere Informationen finden Sie unter Fehlerbehebung bei Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	23. Februar 2021

Verbesserungen an der Konsole	Sie können jetzt einen Application Load Balancer oder Network Load Balancer über die Amazon EC2 Auto Scaling-Konsole erstellen und anhängen. Weitere Informationen finden Sie unter Einen neuen Application Load Balancer oder Network Load Balancer (Konsole) erstellen und anhängen im Amazon EC2 Auto Scaling-Benutzerhandbuch.	24. November 2020
Mehrere Netzwerkschnittstellen	Sie können jetzt eine Startvorlage für eine Auto-Scaling-Gruppe konfigurieren, die mehrere Netzwerkschnittstellen angibt. Weitere Informationen finden Sie unter Netzwerkschnittstellen in einer VPC.	23. November 2020
Mehrere Startvorlagen	Mehrere Startvorlagen können jetzt mit Auto-Scaling-Gruppen verwendet werden. Weitere Informationen finden Sie unter Angeben einer anderen Einführungsvorlage für einen Instance-Typ im Amazon EC2 Auto Scaling-Benutzerhandbuch.	19. November 2020

Gateway Load Balancer	Aktualisierter Leitfaden, der zeigt, wie Sie einen Gateway Load Balancer an eine Auto-Scaling-Gruppe anfügen, um sicherzustellen, dass Appliance-Instances, die von Amazon EC2 Auto Scaling gestartet wurden, automatis ch registriert und vom Load Balancer abgemeldet werden. Weitere Informationen finden Sie unter Elastic Load Balancing-Typen und Anfügen eines Load Balancer an Ihre Auto-Scaling-Gruppe im Amazon EC2 Auto Scaling-B enutzerhandbuch.	10. November 2020
Maximale Lebensdauer von Instances	Sie können jetzt die maximale Instance-Lebensdauer auf einen Tag (86.400 Sekunden) reduzieren. Weitere Informati onen finden Sie unter Ersetzen von Auto Scaling-Instances basierend auf der maximalen Instance-Lebensdauer im Amazon EC2 Auto Scaling-B enutzerhandbuch.	9. November 2020

Kapazitätsausgleich	Sie können Ihre Auto-Scaling-Gruppe so konfigurieren, dass sie um eine Ersatz-Spot-Instance startet, wenn Amazon EC2 eine Empfehlung zum Neuausgleich ausgibt. Weitere Informationen dazu finden Sie unter Amazon EC2 Auto Scaling Capacity Rebalancing im Amazon EC2 Auto Scaling User Guide.	4. November 2020
Instance Metadata Service Version 2	Sie können die Verwendung des Instance-Metadatenservices Version 2 verlangen. Es handelt sich um eine sitzungsorientierte Methode zum Anfordern von Instance-Metadaten, wenn Startkonfigurationen verwendet werden. Weitere Informationen finden Sie unter Konfigurieren der Instance-Metadatenoptionen im Amazon EC2 Auto Scaling-Benutzerhandbuch.	28. Juli 2020

Änderungen im Handbuch	Verschiedene Verbesserungen und neue Konsolenverfahren in den Abschnitten Steuern, welche Auto-Scaling-Instances während der Abskalierung beendet werden, Überwachen der Auto-Scaling-Instances und Auto-Scaling-Gruppen, Startvorlagen und Startkonfigurationen im Benutzerhandbuch für Amazon EC2 Auto Scaling.	28. Juli 2020
Instance-Aktualisierung	Starten Sie eine Instance-Aktualisierung, um alle Instances in Ihrer Auto-Scaling-Gruppe zu aktualisieren, wenn Sie eine Konfigurationsänderung vornehmen. Weitere Informationen finden Sie unter Ersetzen von Auto Scaling-Instances basierend auf einer Instance-Aktualisierung im Benutzerhandbuch zum Amazon EC2 Auto Scaling.	16. Juni 2020

Änderungen im Handbuch	Verschiedene Verbesserungen in den Abschnitten Ersetzen von -Instances basierend auf der maximalen Instance-Lebensdauer, Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen, Skalierung auf Grundlage von Amazon SQS und Markieren von Auto-Scaling-Gruppen und -Instances im Benutzerhandbuch zu Amazon EC2 Auto Scaling.	6. Mai 2020
Änderungen im Handbuch	Verschiedene Verbesserungen an der IAM-Dokumentation. Weitere Informationen finden Sie unter Startvorlagen-Support und Beispiele für identitätsbasierte Amazon EC2 Auto Scaling-Richtlinien im Benutzerhandbuch für Amazon EC2 Auto Scaling.	4. März 2020

Deaktivieren von Skalierungsrichtlinie	Sie können jetzt Skalierungsrichtlinien deaktivieren und wieder aktivieren. Mit dieser Funktion können Sie eine Skalierungsrichtlinie vorübergehend deaktivieren, während die Konfigurationsdetails beibehalten werden, so dass Sie die Richtlinie später erneut aktivieren können. Weitere Informationen finden Sie unter Deaktivieren einer Skalierungsrichtlinie für eine Auto-Scaling-Gruppe im Amazon EC2 Auto Scaling-Benutzerhandbuch.	18. Februar 2020
Hinzufügen einer Benachrichtigungsfunktion	Amazon EC2 Auto Scaling sendet jetzt Ereignisse an Sie, AWS Health Dashboard wenn Ihre Auto Scaling Scaling-Gruppen aufgrund einer fehlenden Sicherheitsgruppe oder Startvorlage nicht skalieren können. Weitere Informationen finden Sie unter AWS Health Dashboard-Benachrichtigungen für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	12. Februar 2020

Änderungen im Handbuch	Verschiedene Verbesserungen und Korrekturen in den Abschnitten Funktionsweise von Amazon EC2 Auto Scaling mit IAM, Identitätsbasierte Amazon EC2 Auto Scaling-Richtlinienbeispiele, Erforderliche CMK-Schlüsselrichtlinie für Verwendung mit verschlüsselten Volumes und Überwachung Ihrer Auto-Scaling-instances und -Gruppen des Amazon EC2 Auto Scaling-Benutzerhandbuchs.	10. Februar 2020
Änderungen im Handbuch	Verbesserte Dokumentation für Auto-Scaling-Gruppen, die Instance-Gewichtung verwenden. Erfahren Sie, wie Sie Skalierungsrichtlinien verwenden, wenn Sie „Kapazitätseinheiten“ verwenden, um die gewünschte Kapazität zu messen. Weitere Informationen finden Sie unter Funktionsweise von Skalierungsrichtlinien und Skalierungsanpassungstypen im Benutzerhandbuch für Amazon EC2 Auto Scaling.	6. Februar 2020

Neues Sicherheitskapitel	Ein neues Sicherheits-Kapitel im Benutzerhandbuch für Amazon EC2 Auto Scaling hilft Ihnen zu verstehen, wie Sie das Modell der geteilten Verantwortung anwenden, wenn Sie Amazon EC2 Auto Scaling verwenden. Als Teil dieses Updates wurde das Kapitel „Zugriffskontrolle für Ihre Amazon EC2 Auto Scaling-Ressourcen“ im Benutzerhandbuch durch den neuen, nützlicheren Abschnitt Identity and Access Management für Amazon EC2 Auto Scaling ersetzt.	4. Februar 2020
Empfehlungen für Instance-Typen	AWS Compute Optimizer bietet Amazon EC2 EC2-Instance-Empfehlungen, um Ihnen zu helfen, die Leistung zu verbessern, Geld zu sparen oder beides. Weitere Informationen finden Sie unter Erhalten von Empfehlungen für einen Instance-Typ im Benutzerhandbuch für Amazon EC2 Auto Scaling.	3. Dezember 2019

Dedicated Hosts und Hostressourcengruppen	Aktualisierte Anleitung, um zu zeigen, wie eine Startvorlage erstellt wird, die eine Host-Ressourcengruppen angibt. Auf diese Weise können Sie eine Auto-Scaling-Gruppe mit einer Startvorlage erstellen, die ein BYOL-AMI angibt, die auf Dedicated Hosts verwendet wird. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Group im Amazon EC2 Auto Scaling-Benutzerhandbuch.	3. Dezember 2019
Unterstützung für Amazon VPC-Endpunkte	Sie können jetzt eine private Verbindung zwischen Ihrer VPC und Amazon EC2 Auto Scaling herstellen. Weitere Informationen finden Sie unter Amazon EC2 Auto Scaling und VPC-Schnittstellenendpunkte im Amazon EC2 Auto Scaling-Benutzerhandbuch.	22. November 2019

Maximale Lebensdauer von Instances	Sie können Instances jetzt automatisch ersetzen, indem Sie die Maximaldauer für den Betrieb einer Instance angeben. Wenn sich Instances diesem Grenzwert nähern, werden sie von Amazon EC2 Auto Scaling schrittweise ersetzt. Weitere Informationen finden Sie unter Ersetzen von Auto Scaling-Instances basierend auf der maximalen Instance-Lebensdauer im Amazon EC2 Auto Scaling-Benutzerhandbuch.	19. November 2019
Instance-Gewichtung	Für Auto-Scaling-Gruppen mit mehreren Instance-Typen können Sie nun optional die Anzahl der Kapazitätseinheiten angeben, die jeder Instance-Typ zur Kapazität der Gruppe beiträgt. Weitere Informationen finden Sie unter Instance-Gewichtung für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.	19. November 2019

Mindestanzahl der Instance-Typen	Sie müssen keine zusätzlichen Instance-Typen mehr für Gruppen von Spot, On-Demand und Reserved Instances angeben. Für alle Auto-Scaling-Gruppen beträgt der Mindestwert jetzt ein Instance-Typ. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen im Amazon EC2 Auto Scaling-Benutzerhandbuch.	16. September 2019
Unterstützung für eine neue Spot-Zuweisungsstrategie	Amazon EC2 Auto Scaling unterstützt jetzt eine neue „kapazitätsoptimierte“ Spot-Zuweisungsstrategie, die Ihre Anfrage mit Spot-Instance-Pools erfüllt, die basierend auf der verfügbaren Spot-Kapazität optimal ausgewählt werden. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen im Amazon EC2 Auto Scaling-Benutzerhandbuch.	12. August 2019

Änderungen im Handbuch	Verbesserte Amazon EC2 Auto Scaling-Dokumentation in den Themen Serviceverknüpfte Rollen und Erforderliche CMK-Schlüsselrichtlinie für die Verwendung mit verschlüsselten Volumes.	1. August 2019
Unterstützung für Tagging-Erweiterung	Amazon EC2 Auto Scaling fügt Tags jetzt im Rahmen desselben API-Aufrufs, der die Instances startet, zu Amazon EC2 Instances hinzu. Weitere Informationen finden Sie unter Markieren von Auto-Scaling-Gruppen und -Instances.	26. Juli 2019
Änderungen im Handbuch	Verbesserte Amazon EC2 Auto Scaling-Dokumentation im Thema Aus- und Fortsetzen von Skalierungsprozessen. Das Thema Beispiele für vom Kunden verwaltete Richtlinien wurde aktualisiert und enthält jetzt ein Beispiel für eine Richtlinie, die es Benutzern ermöglicht, nur bestimmte serviceverknüpfte Rollen mit benutzerdefiniertem Suffix an Amazon EC2 Auto Scaling zu übergeben.	13. Juni 2019

Unterstützung für neue Amazon EBS-Funktion	Unterstützung für neue Amazon EBS-Funktion im Startvorlagen-Thema hinzugefügt. Ändern Sie den Verschlüsselungsstatus eines EBS-Volumes während der Wiederherstellung von einem Snapshot. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Group im Amazon EC2 Auto Scaling-Benutzerhandbuch.	13. Mai 2019
Änderungen im Handbuch	Verbesserte Amazon EC2 Auto Scaling-Dokumentation in den folgenden Abschnitten: Steuern der Auswahl der bei der horizontalen Skalierung nach unten zu beendenden Auto-Scaling-Instances, Auto-Scaling-Gruppen, Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen und Dynamische Skalierung für Amazon EC2 Auto Scaling.	12. März 2019

Unterstützung zum Kombinieren von Instance-Typen und Kaufoptionen	Bereitstellung und automatische Skalierung der Instances in den Kaufoptionen (Spot-, On-Demand- und Reserved Instances) und Instance-Typen innerhalb einer einzelnen Auto-Scaling-Gruppe. Weitere Informationen finden Sie unter Auto-Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen im Amazon EC2 Auto Scaling-Benutzerhandbuch.	13. November 2018
Thema für die Skalierung auf Basis von Amazon SQS aktualisiert	Aktualisierte Anleitung mit der Beschreibung, wie Sie benutzerdefinierte Metriken verwenden, um eine Auto-Scaling-Gruppe als Reaktion auf geänderten Bedarf von einer Amazon SQS-Warteschlange zu skalieren. Weitere Informationen finden Sie unter Skalierung basierend auf Amazon SQS im Benutzerhandbuch für Amazon EC2 Auto Scaling.	26. Juli 2018

In der folgenden Tabelle werden die wichtigen Änderungen an der Amazon EC2 Auto Scaling-Dokumentation vor Juli 2018 beschrieben.

Funktion	Beschreibung	Datum der Veröffentlichung
Unterstützung für Skalierungsrichtli	Richten Sie in nur wenigen Schritten eine dynamische Skalierung für Ihre Anwendung ein. Weitere Informati	12. Juli 2017

Funktion	Beschreibung	Datum der Veröffentlichung
nien für die Ziel-Nachverfolgung	onen finden Sie unter Skalierungsrichtlinien für die Zielverfolgung für Amazon EC2 Auto Scaling.	
Unterstützung für Berechtigungen auf Ressource nebene	Erstellen von IAM-Richtlinien zur Kontrolle des Zugriffs auf Ressourcenebene. Weitere Informationen finden Sie unter Steuern des Zugriffs auf Ihre Amazon EC2 Auto Scaling-Ressourcen.	15. Mai 2017
Überwachen von Verbesserungen	Auto-Scaling-Gruppenmetriken erfordern nicht mehr, dass Sie die detaillierte Überwachung aktivieren. Sie können die Erfassung von Gruppenmetriken jetzt auf der Registerkarte Monitoring der Konsole aktivieren und dort Metrikdiagramme anzeigen. Weitere Informationen finden Sie unter Überwachen Ihrer Auto Scaling Scaling-Gruppen und -Instances mithilfe von Amazon CloudWatch.	18. August 2016
Support für Application Load Balancer	Hinzufügen von Zielgruppen zu einer neuen oder bestehenden Auto-Scaling-Gruppe. Weitere Informationen finden Sie unter Anfügen eines Load Balancers an Ihre Auto-Scaling-Gruppe.	11. August 2016
Ereignisse für Lebenszyklus-Hooks	Amazon EC2 Auto Scaling sendet Ereignisse an, EventBridge wenn es Lifecycle-Hooks aufruft. Weitere Informationen finden Sie unter Ermitteln, EventBridge wann Ihre Auto Scaling-Gruppe skaliert.	24. Februar 2016
Instance-Schutz	Verhindern der Beendigung bestimmter Instances durch Amazon EC2 Auto Scaling bei der horizontalen Skalierung nach unten. Weitere Informationen finden Sie unter Instance-Schutz.	07. Dezember 2015

Funktion	Beschreibung	Datum der Veröffentlichung
Richtlinien zur schrittweisen Skalierung	Erstellen einer Skalierungsrichtlinie, die Ihnen eine Skalierung auf Grundlage des Ausmaßes der Alarmüberschreitung ermöglicht. Weitere Informationen finden Sie unter Skalierungsrichtlinientypen.	06. Juli 2015
Aktualisieren des Load Balancers	Hinzufügen eines Load Balancers zu und Trennen eines Load Balancers von einer vorhandenen Auto-Scaling-Gruppe. Weitere Informationen finden Sie unter Anfügen eines Load Balancers an Ihre Auto-Scaling-Gruppe.	11. Juni 2015
Support für ClassicLink	Verknüpfen von EC2-Classic-Instances der Auto-Scaling-Gruppe mit einer VPC, was die Kommunikation zwischen diesen verknüpften EC2-Classic-Instances und den Instances in der VPC mit privaten IP-Adressen ermöglicht. Weitere Informationen finden Sie unter Verknüpfen von EC2-Classic-Instances mit einer VPC.	19. Januar 2015
Lebenszyklus-Hooks	Belassen von neu gestarteten oder in der Beendigung begriffenen Instances in einem schwebenden Status, während an ihnen Aktionen durchgeführt werden. Weitere Informationen hierzu finden Sie unter Amazon EC2 Auto Scaling-Lebenszyklus-Hooks.	30. Juli 2014
Trennen von Instances	Trennen Sie die Instances von der Auto-Scaling-Gruppe. Weitere Informationen finden Sie unter Detach EC2 Instances From Your Auto Scaling Group.	30. Juli 2014
Versetzen von Instances in einen Standby-Status	Versetzen von Instances in einem `InService` -Status in einen `Standby`-Status. Weitere Informationen finden Sie unter Vorübergehendes Entfernen von Instances aus einer Auto-Scaling-Gruppe.	30. Juli 2014

Funktion	Beschreibung	Datum der Veröffentlichung
Verwalten von Tags	Verwalten Sie Ihre Auto-Scaling-Gruppen mit der AWS Management Console. Weitere Informationen finden Sie unter Markieren von Auto-Scaling-Gruppen und -Instances.	01. Mai 2014
Unterstützung für Dedicated Instances	Starten von Dedicated Instances durch Angabe eines Placement-Tenancy-Attributs beim Erstellen einer Startkonfiguration. Weitere Informationen finden Sie unter Tenancy zur Instance-Platzierung.	23. April 2014
Erstellen einer Gruppe oder Startkonfiguration aus einer EC2-Instance	Erstellen einer Auto-Scaling-Gruppe oder einer Startkonfiguration mithilfe einer EC2-Instance. Weitere Informationen zum Erstellen einer Startkonfiguration mithilfe einer EC2-Instance finden Sie unter Erstellen einer Startkonfiguration mithilfe einer EC2-Instance. Weitere Informationen zum Erstellen einer Auto-Scaling-Gruppe mithilfe einer EC2-Instance finden Sie unter Erstellen einer Auto-Scaling-Gruppe mit einer EC2-Instance.	02. Januar 2014
Hinzufügen von Instances	Aktivieren der automatischen Skalierung für eine EC2-Instance durch Hinzufügen der Instance zu einer bestehenden Auto-Scaling-Gruppe. Weitere Informationen finden Sie unter Hinzufügen von EC2-Instances zu Ihrer Auto-Scaling-Gruppe.	02. Januar 2014
Anzeigen von Kontolimits	Anzeigen der Limits von Auto-Scaling-Ressourcen für Ihr Konto. Weitere Informationen finden Sie unter Auto-Scaling-Begrenzungen.	02. Januar 2014
Konsolensupport für Amazon EC2 Auto Scaling	Greifen Sie mit dem auf Amazon EC2 Auto Scaling zu. AWS Management Console Weitere Informationen finden Sie unter Erste Schritte mit Amazon EC2 Auto Scaling.	10. Dezember 2013

Funktion	Beschreibung	Datum der Veröffentlichung
Zuweisen einer öffentlichen IP-Adresse	Zuweisen einer öffentlichen IP-Adresse an eine Instance einer VPC. Weitere Informationen finden Sie unter Starten von Auto-Scaling-Instances in einer VPC.	19. September 2013
Instance-Beendigungsrichtlinie	Angeben einer Instance-Beendigungsrichtlinie für Amazon EC2 Auto Scaling zur Verwendung beim Beenden von EC2-Instances. Weitere Informationen finden Sie unter Steuern, welche Auto-Scaling-Instances während der horizontalen Skalierung nach unten beendet werden.	17. September 2012
Unterstützung für IAM-Rollen	Starten von EC2-Instances mit einem IAM-Instance-Profil. Sie können diese Funktion verwenden, um Instances IAM-Rollen zuzuweisen, was Anwendungen sicheren Zugriff auf andere Amazon Web Services ermöglicht. Weitere Informationen finden Sie unter Starten von Auto-Scaling-Instances mit einer IAM-Rolle.	11. Juni 2012
Unterstützung für Spot-Instances	Starten Sie Spot-Instances mit einer Startkonfiguration. Weitere Informationen finden Sie unter Anfordern von Spot-Instanzen für fehlertolerante und flexible Anwendungen.	7. Juni 2012
Markieren von Gruppen und Instances	Markieren von Auto-Scaling-Gruppen und Festlegen der Einstellung, dass das Tag auch nach seiner Erstellung gestarteten EC2-Instances hinzugefügt wird. Weitere Informationen finden Sie unter Markieren von Auto-Scaling-Gruppen und -Instances.	26. Januar 2012

Funktion	Beschreibung	Datum der Veröffentlichung
Support für Amazon SNS	Verwenden von Amazon SNS für den Empfang von Benachrichtigungen, wenn Amazon EC2 Auto Scaling EC2-Instances startet oder beendet. Weitere Informationen finden Sie unter Erhalten von SNS-Benachrichtigungen über Skalierungen Ihrer Auto-Scaling-Gruppe. Amazon EC2 Auto Scaling umfasst auch die folgenden neuen Funktionen: • Die Möglichkeit, wiederkehrende Skalierungsaktivitäten mithilfe von Cron-Syntax einzurichten. Weitere Informationen finden Sie unter `PutScheduledUpdateGroupAction` -API-Operation. • Eine neue Konfigurationseinstellung, mit der Sie horizontal skalieren können, ohne die gestartete Instance dem Load Balancer () LoadBalancer hinzuzufügen. Weitere Informationen finden Sie unter `ProcessType` -API-Datentyp. • Das Flag `ForceDelete` in der Operation `DeleteAutoScalingGroup` , das Amazon EC2 Auto Scaling zum Löschen der Auto-Scaling-Gruppe einschließlich der ihr zugeordneten Instances anweist, ohne auf die Beendigung der Instances zu warten. Weitere Informationen finden Sie unter `DeleteAutoScalingGroup` -API-Operation.	20. Juli 2011
Geplante Skalierungsaktionen	Unterstützung für geplante Skalierungsaktionen hinzugefügt. Weitere Informationen finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling.	2. Dezember 2010
Support für Amazon VPC	Support für Amazon VPC hinzugefügt. Weitere Informationen finden Sie unter Starten von Auto-Scaling-Instances in einer VPC.	2. Dezember 2010

Funktion	Beschreibung	Datum der Veröffentlichung
Unterstützung für HPC-Cluster	HPC-Cluster (High Performance Computing (HPC)) werden nun unterstützt.	2. Dezember 2010
Unterstützung für Zustandsprüfungen	Die Verwendung von Elastic Load Balancing-Zustandsprüfungen mit Amazon EC2 Auto Scaling-verwalteten EC2-Instances wurde hinzugefügt. Weitere Informationen finden Sie unter Integritätsprüfungen für Instances in einer Auto Scaling Scaling-Gruppe.	2. Dezember 2010
Support für CloudWatch Alarme	Der ältere Auslösemechanismus wurde entfernt und Amazon EC2 Auto Scaling neu gestaltet, um die CloudWatch Alarmfunktion zu verwenden. Weitere Informationen finden Sie unter Dynamische Skalierung für Amazon EC2 Auto Scaling.	2. Dezember 2010
Anhalten und Fortsetzen von Skalierungen	Unterstützung für das Anhalten und Fortsetzen von Skalierungsprozessen hinzugefügt.	2. Dezember 2010
Unterstützung für IAM	IAM wird nun unterstützt. Weitere Informationen finden Sie unter Steuern des Zugriffs auf Ihre Amazon EC2 Auto Scaling-Ressourcen.	2. Dezember 2010

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.