



Leitfaden zur Implementierung

Instance Scheduler auf AWS



Instance Scheduler auf AWS: Leitfaden zur Implementierung

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

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

Table of Contents

Übersicht über die Lösung	1
Features und Vorteile	2
Anwendungsfälle	3
Konzepte und Definitionen	3
Cost (Kosten)	4
Faktoren der Kostenskalierung	4
Berechnung der Planungsziele	4
Strategien zur Kostenoptimierung	5
Beispiele für Referenzpreise (monatlich)	5
Kostenschätzung für Ihren Einsatz	8
Kontingente	8
Einschränkungen bei der Skalierung	8
Weitere Überlegungen	9
AWS-Servicekontingente	10
Unterstützte AWS Regionen	10
Kontenübergreifende Planung von Instances mithilfe eines Kontos IDs oder einer AWS-Organisations-ID	11
Aktivieren Sie die kontenübergreifende Planung mithilfe von Account IDs	11
Aktivierung der kontenübergreifenden Terminplanung mithilfe der AWS-Organisations-ID	12
Konto IDs mit dem AWS Systems Manager Parameter Store verwalten	12
Dienste, die für die Planung unterstützt werden	13
Verhalten beim Herunterfahren der Instanz	13
Amazon EC2	13
Amazon RDS, Amazon Neptune und Amazon DocumentDB	13
Amazon-RDS-Wartungsfenster	14
Amazon EC2 Auto-Scaling-Gruppen	14
Architektur	16
Architekturdiagramm	16
Überlegungen zum AWS-Well-Architected-Design	19
Operative Exzellenz	19
Sicherheit	20
Zuverlässigkeit	20
Leistungseffizienz	20
Kostenoptimierung	21

Nachhaltigkeit	21
Scheduler-Konfigurationstabelle	21
Planer-CLI	21
In dieser Lösung verwendete AWS-Services	22
Sicherheit	24
AWS KMS	24
Amazon IAM	24
Verschlüsselte EC2-EBS-Volumes	25
EC2-Lizenzmanager	26
Erste Schritte	28
Überblick über den Bereitstellungsprozess	28
CloudFormation AWS-Vorlagen	29
Schritt 1: Starten Sie den Instance Scheduler Hub-Stack	29
Schritt 2 (optional): Starten Sie den Remote-Stack in sekundären Konten	37
Konfigurieren Sie die Lösung	41
Leitfaden für Bediener	42
Zeitpläne konfigurieren	42
Nutzung von Infrastructure as Code (empfohlen)	43
Verwenden der Amazon DynamoDB DynamoDB-Konsole und des Instance Schedulers auf der AWS-CLI	43
Kennzeichnen Sie Instanzen für die Planung	44
Den Tag-Wert festlegen	44
EC2-Instances mit verschlüsselten EBS-Volumes	45
Im License Manager verwaltete EC2-Instances	45
Referenz zum Zeitplan	45
Zeiträume	45
Zeitzone	46
Feld „Neue Instanzen beenden“	46
Feld „Ruhezustand“	46
Feld erzwungen	46
Feld „Laufend“ beibehalten	47
Feld für das Wartungsfenster von Systems Manager (gilt nur für EC2-Instances)	47
Instance-Typ	48
Definitionen planen	48
Referenz für den Zeitraum	51
Start- und Stopzeiten	51

Wochentage	52
Tage des Monats	53
Monate	53
Definitionen von Perioden	53
Beispiele für Zeitpläne	56
Standardarbeitszeiten von 9 bis 5 Stunden	57
Stoppen Sie Instances nach 17 Uhr	59
Stoppen Sie Instances über das Wochenende	61
Ressourcen zur Lösung	64
Planer-CLI	65
Voraussetzungen	65
Anmeldeinformationen	65
Installieren Sie die Scheduler-CLI	66
Befehlsstruktur	67
Allgemeine Argumente	67
Verfügbare Befehle	68
Zeitraum erstellen	69
Zeitplan erstellen	71
Zeitraum löschen	73
Zeitplan löschen	74
Zeiträume beschreiben	74
Zeitpläne beschreiben	76
describe-schedule-usage	77
Aktualisierungszeitraum	79
Zeitplan für die Aktualisierung	79
help	79
Aktualisieren Sie die globalen Konfigurationseinstellungen	81
Verwalten Sie Zeitpläne mithilfe von Infrastructure as Code (IaC)	81
Behandlung von EC2-Fehlern mit unzureichender Kapazität	84
Konfiguration	84
Funktionsweise	84
Anforderungen und Einschränkungen	85
Beispiel	85
EC2 Auto Scaling Scaling-Gruppenplanung	85
Überblick über die ASG-Planung	85
Definition von für Running/Stopped ASGs	85

ASG-Verhalten Start/Stop	86
Überwachen Sie die Lösung	87
Protokollierung und Benachrichtigungen	87
Informations-Tags	87
CloudWatch Protokolliert Insights-Abfragen	91
Dashboard mit operativen Erkenntnissen	92
EventBridge Ereignisse überwachen	94
Fehlerbehebung	97
Lösung eines bekannten Problems	97
Problem: Instanzen werden nicht in einem Remote-Konto geplant (v1.4-v3.0)	97
Auflösung	97
Problem: Instanzen werden nicht geplant (v3.1+)	98
Auflösung	98
Problem: Verschlüsselte EC2-Instances werden nicht gestartet	98
Auflösung	98
Problem: Unerwartete API-Kosten aufgrund von Informations-Tagging	99
Auflösung	99
Problem: RDS-Instances werden nicht gestoppt, wenn „RDS-Snapshots erstellen“ aktiviert ist	99
Auflösung	99
Kontaktieren Sie AWS Support.	99
Fall erstellen	99
Wie können wir helfen?	100
Zusätzliche Informationen	100
Helfen Sie uns, Ihren Fall schneller zu lösen	100
Löse es jetzt oder kontaktiere uns	100
Aktualisieren Sie die Lösung	101
Wichtigste Änderungen in bestimmten Versionen	102
v1.5.0	102
v3.0.0	103
v3.1.0	104
Deinstallieren Sie die Lösung	106
Verwendung der AWS-Managementkonsole	106
Verwenden der AWS-Befehlszeilenschnittstelle	106
Entwicklerhandbuch	108
Quellcode	108

Referenz	109
Datenerfassung	109
Zugehörige Ressourcen	109
Mitwirkende	110
Überarbeitungen	112
Hinweise	113
.....	cxiv

Automatisieren Sie das Starten und Stoppen von AWS-Instanzen

Die Instance Scheduler on AWS-Lösung automatisiert das Starten und Stoppen verschiedener AWS-Services, darunter [Amazon Elastic Compute Cloud](#) (Amazon EC2) und Amazon [Relational Database Service \(Amazon RDS\)](#) -Instances.

Diese Lösung trägt zur Senkung der Betriebskosten bei, indem Ressourcen, die nicht genutzt werden, gestoppt und Ressourcen gestartet werden, wenn ihre Kapazität benötigt wird.

Beispielsweise kann ein Unternehmen Instance Scheduler auf AWS verwenden, um Instances jeden Tag außerhalb der Geschäftszeiten automatisch zu stoppen. Wenn Sie alle Ihre Instances mit voller Auslastung laufen lassen, kann diese Lösung zu Kosteneinsparungen von bis zu 70% für die Instances führen, die nur während der regulären Geschäftszeiten erforderlich sind (die wöchentliche Nutzung wurde von 168 Stunden auf 50 Stunden reduziert).

Instance Scheduler auf AWS nutzt Amazon Web Services (AWS) Resource Tags und [AWS Lambda](#), um Instances in mehreren AWS-Regionen und Konten nach einem vom Kunden definierten Zeitplan automatisch zu stoppen und neu zu starten. Mit dieser Lösung können Sie auch den Ruhezustand für gestoppte EC2-Instances verwenden.

Dieser Implementierungsleitfaden bietet einen Überblick über die Instance Scheduler on AWS-Lösung, ihre Referenzarchitektur und ihre Komponenten, Überlegungen zur Planung der Bereitstellung und Konfigurationsschritte für die Bereitstellung der Lösung in der AWS-Cloud.

Dieses Handbuch richtet sich an IT-Infrastrukturarchitekten, Administratoren und DevOps Fachleute, die Instance Scheduler auf AWS in ihrer Umgebung implementieren möchten.

Verwenden Sie diese Navigationstabelle, um schnell Antworten auf diese Fragen zu finden:

Wenn du willst.	Lesen.
Informieren Sie sich über die Kosten für den Betrieb dieser Lösung. Die geschätzten Kosten für den Betrieb dieser Lösung in der Region USA Ost (Nord-Virginia) belaufen sich auf 13,15 USD pro Monat.	Kosten
Machen Sie sich mit den Sicherheitsüberlegungen für diese Lösung vertraut.	AWS Well-Architected Security, Sicherheit

Wenn du willst.	Lesen.
Zeitpläne konfigurieren.	Scheduler-Konfigurationstabelle
Erfahren Sie, welche AWS-Regionen für diese Lösung unterstützt werden.	Unterstützte AWS-Regionen
Sehen Sie sich die in dieser Lösung enthaltene CloudFormation AWS-Vorlage an oder laden Sie sie herunter, um die Infrastrukturressourcen (den „Stack“) für diese Lösung automatisch bereitzustellen.	CloudFormation AWS-Vorlagen
Greifen Sie auf den Quellcode zu und verwenden Sie optional das AWS Cloud Development Kit (AWS CDK), um die Lösung bereitzustellen.	GitHub Repository

Features und Vorteile

Die Instance Schedule on AWS-Lösung bietet die folgenden Funktionen:

Kontenübergreifende Planung von Instances

Diese Lösung umfasst eine Vorlage, die die [AWS Identity and Access Management](#) (IAM) -Rollen erstellt, die zum Starten und Stoppen von Instances in sekundären Konten erforderlich sind. Weitere Informationen finden Sie im Abschnitt [Kontoübergreifende Instanzplanung](#).

Automatisiertes Tagging

Instance Scheduler auf AWS kann automatisch Tags zu allen Instances hinzufügen, die er startet oder stoppt. Die Lösung umfasst auch Makros, mit denen Sie den Tags variable Informationen hinzufügen können.

Konfigurieren Sie Zeitpläne oder Perioden mit der Scheduler-CLI

Diese Lösung umfasst eine Befehlszeilenschnittstelle (CLI), die Befehle zur Konfiguration von Zeitplänen und Zeiträumen bereitstellt. Die CLI ermöglicht es Kunden, die Kosteneinsparungen für einen bestimmten Zeitplan abzuschätzen. Weitere Informationen finden Sie in der [Scheduler-CLI](#).

Verwalten Sie Zeitpläne mithilfe von Infrastructure as Code (IaC)

Diese Lösung bietet eine CloudFormation benutzerdefinierte AWS-Ressource, mit der Sie Zeitpläne mithilfe von Infrastructure as Code (IaC) verwalten können. Weitere Informationen finden Sie unter [Zeitpläne mithilfe von Infrastruktur als Code verwalten](#).

Integration mit Systems Manager Maintenance Windows

Für Amazon EC2 EC2-Instances kann Instance Scheduler auf AWS in die Wartungsfenster von [AWS Systems Manager](#) integriert werden, die in derselben Region wie diese Instances definiert sind, um sie entsprechend dem Wartungsfenster zu starten und zu beenden.

Anwendungsfälle

Instances werden nur während der Arbeitszeit ausgeführt

Wenn Sie alle Ihre Instances voll ausgelastet lassen, kann diese Lösung zu Kosteneinsparungen von bis zu 76% für die Instances führen, die nur während der regulären Geschäftszeiten benötigt werden (die wöchentliche Nutzung wurde von 168 Stunden auf 40 Stunden reduziert). Weitere Informationen finden Sie im [Beispielplan](#).

Stoppen von Instanzen nach der Geschäftszeit

Wenn Sie sicherstellen möchten, dass Entwicklungsinstanzen nach Geschäftsschluss und bis zur erneuten Verwendung ausgeschaltet sind, können Sie mit dieser Lösung einen Endzeitraum ohne Startzeitraum festlegen. Weitere Informationen finden Sie im [Beispielplan](#).

Konzepte und Definitionen

In diesem Abschnitt werden die wichtigsten Konzepte beschrieben und die für diese Lösung spezifische Terminologie definiert:

zeitplan

Gruppe von einer oder mehreren Perioden, an die eine Instanz gebunden ist.

Zeitraum

Laufzeit (en), definiert durch eine Start- und Stoppzeit.

sein

Eine unterstützte Ressource, die geplant werden kann. Zum Beispiel eine Amazon EC2-Instance oder ein Amazon RDS-Cluster Amazon EC2 und Amazon RDS.

reguläre Geschäftszeiten

An Wochentagen von 9.00 bis 17.00 Uhr (9.00 bis 17.00 Uhr) ET

Eine allgemeine Referenz zu AWS-Begriffen finden Sie im [AWS-Glossar](#).

Cost (Kosten)

Sie sind für die Kosten der AWS-Services verantwortlich, die Sie bei der Ausführung von Instance Scheduler verwenden. Wenn Sie wissen, wie die Kosten mit Ihrer Bereitstellungsgröße skalieren, können Sie Ihre Implementierung besser planen und optimieren.

Faktoren der Kostenskalierung

Die Kosten für Instance Scheduler hängen von mehreren Faktoren ab:

Anzahl der Scheduling-Ziele: Die Anzahl der einzelnen account-region-service Kombinationen, die verwaltet werden. Jedes Ziel erfordert einen separaten Lambda-Aufruf pro Planungsintervall.

Ressourcen pro Ziel: Die Anzahl der Ressourcen (EC2-Instances, RDS-Datenbanken usw.) innerhalb jedes Ziels beeinflusst die Lambda-Ausführungszeit und die Kosten für die Dauer.

Komplexität der operativen Metriken: Die Kosten für optionale CloudWatch Metriken hängen von der Anzahl der eindeutigen Instance-Typen und aktiven Zeitpläne ab, die in Ihrer gesamten Bereitstellung verfolgt werden.

Häufigkeit der Terminplanung: Die Lösung wird auf der Grundlage der von Ihnen konfigurierten Frequenz ausgeführt (Standard: 5 Minuten). Häufigere Überprüfungen erhöhen die Lambda Lambda-Aufrufe von 24-mal täglich (stündlich) auf 288-mal täglich (5-Minuten-Intervalle).

Berechnung der Planungsziele

Ein Planungsziel ist eine einzigartige Kombination account-region-service aus mindestens einer aktiv verwalteten Instanz. Mehrere Instanzen innerhalb derselben account-region-service Kombination gelten als ein einzelnes Scheduling-Ziel.

Berechnungsbeispiel:

- Konto A, us-east-1, 5 EC2-Instances = 1 Scheduling-Ziel
- Konto A, us-east-1, 3 RDS-Datenbanken = 1 Planungsziel
- Konto A, us-east-1, 2 Auto Scaling Scaling-Gruppen = 1 Planungsziel

- Konto A, us-west-2, 2 EC2-Instances = 1 Scheduling-Ziel
- Konto B, us-east-1, 10 EC2-Instances = 1 Scheduling-Ziel

Insgesamt: 5 Planungsziele

Das bedeutet, dass die Lösung pro Planungsintervall 5 separate Lambda-Funktionen aufruft, um alle Ressourcen in diesen account-region-service Kombinationen zu verwalten.

Note

Ziele können in den Geltungsbereich der Planung fallen, werden aber bei Kostenberechnungen erst dann als „aktiv“ betrachtet, wenn mindestens eine Ressource in diesem Ziel für die Planung markiert ist.

Zur Kostenoptimierung gruppiert Instance Scheduler alle Amazon RDS-bezogenen Dienste in einem einzigen Aufruf. Daher zählen Amazon RDS, [Amazon Aurora](#), [Amazon Neptune](#) und [Amazon DocDB Scheduling](#) alle als ein „RDS“-Service für Kostenberechnungen.

Strategien zur Kostenoptimierung

1. Bereitstellung in einer Region mit niedrigeren Lambda-Preisen
2. Verwenden Sie die standardmäßige Lambda-Speichereinstellung von 512 MB, sofern nicht erforderlich, um sie durch Skalierungsbeschränkungen für einzelne Ziele zu erhöhen
3. Minimiere die Anzahl der eindeutigen Zeitpläne und Instance-Typen, die aktiv genutzt werden
4. Passen Sie die Zeitplanungshäufigkeit Ihren Anforderungen entsprechend an
5. Deaktivieren Sie das Dashboard für Betriebskennzahlen, wenn Sie es nicht verwenden möchten

Weitere Informationen finden Sie auf der Preisseite für jeden [AWS-Service in dieser Lösung](#).

Wir empfehlen, über den AWS Cost Explorer ein [Budget](#) zu erstellen, um die Kosten besser verwalten zu können. Die Preise sind freibleibend.

Beispiele für Referenzpreise (monatlich)

Die folgenden Beispiele zeigen, wie die Kosten bei verschiedenen Bereitstellungsgrößen skalieren. Verwenden Sie diese als Anhaltspunkte, um die Kosten für Ihre spezifische Bereitstellung abzuschätzen.

Note

Bei allen Referenzpreisen handelt es sich um grobe Schätzungen der Kosten für die wichtigsten Dienste, die von der Lösung genutzt werden.

Kleine Bereitstellung (~9 \$ pro Monat)

Dieses Beispiel stellt eine typische Entwicklungs- oder kleine Produktionsbereitstellung dar:

- 5 aktive Ziele
- 20 verwaltete Ressourcen
- 3 aktive Zeitpläne
- 2 Instanztypen
- 5-minütiges Planungsintervall
- 512 MB Lambda-Funktion, durchschnittliche Laufzeit von 5 Sekunden

AWS Service	Monatliche Kosten [USD]
AWS Lambda	~2,00 \$
AWS KMS	~1,50 \$
CloudWatch Logs	~0,30 \$
CloudWatch Metriken	~5,30 \$
Amazon DynamoDB	~0,05 \$
Insgesamt:	~9,15 \$

Mittlerer Einsatz (~161 \$ monatlich)

Dieses Beispiel stellt eine Bereitstellung in mittelständischen Unternehmen dar:

- 250 aktive Ziele

- 1000 verwaltete Ressourcen
- 15 aktive Zeitpläne
- 15 Instanztypen
- 5-minütiges Planungsintervall
- 512 MB Lambda-Funktion, durchschnittliche Laufzeit von 5 Sekunden
- 5 EC2-Wartungsfenster

AWS Service	Monatliche Kosten [USD]
AWS Lambda	~\$95.00
Amazon DynamoDB	~1,00 \$
CloudWatch Logs	~10,00 \$
CloudWatch Metriken	~40,00 \$
AWS KMS	~15,00 \$
Insgesamt:	~161,00 \$

Umfangreicher Einsatz (~630 \$ monatlich)

Dieses Beispiel stellt eine große Unternehmensbereitstellung dar:

- 1000 aktive Ziele
- 5000 verwaltete Ressourcen
- 500 aktive Zeitpläne
- 50 Instance-Typen
- 5-minütiges Planungsintervall
- 512 MB Lambda-Funktion, durchschnittliche Laufzeit von 5 Sekunden
- 100 EC2-Wartungsfenster

AWS Service	Monatliche Kosten [USD]
AWS Lambda	~380,00 \$
Amazon DynamoDB	~5,00 \$
CloudWatch Logs	~50,00 \$
CloudWatch Metriken	~140,00 \$
AWS KMS	~55,00 \$
Insgesamt:	~630,00 \$

Kostenschätzung für Ihren Einsatz

So schätzen Sie die Kosten für Ihre spezifische Bereitstellung ab:

1. Zählen Sie Ihre gesamten verwalteten Ressourcen (EC2-Instances, RDS-Datenbanken usw.)
2. Ermitteln Sie die Anzahl der Konten und Regionen, die Sie verwalten werden
3. Berücksichtigen Sie Ihre erforderliche Terminplanungshäufigkeit
4. Entscheiden Sie, ob Sie Betriebskennzahlen benötigen
5. Verwenden Sie die obigen Referenzbeispiele, um Ihre erwarteten Kosten zu interpolieren

Kontingente

Einschränkungen bei der Skalierung

Instance Scheduler skaliert auf zwei Hauptachsen, um große Unternehmensbereitstellungen zu verwalten:

Vertikale Skalierung (Ressourcen pro Ziel)

Die vertikale Skalierung wird durch die Anzahl der Ressourcen begrenzt, die eine einzelne Scheduling Request Lambda-Funktion innerhalb eines einzigen Planungsziels (account/region/serviceKombination) effizient verarbeiten kann.

Der Instance Scheduler ist so konzipiert, dass er Tausende EC2s, Hunderte von und Hunderte von RDS dbs/clusters in einem einzigen [Scheduling-Ziel verarbeiten kann ASGs, kann jedoch durch die regionsübergreifende Latenz](#) begrenzt sein.

Um eine optimale Leistung zu gewährleisten, empfehlen wir, die Ausführungszeit des Scheduling Request Lambda zu überwachen (siehe [Operational Insights-Dashboard](#)). Wir empfehlen, die durchschnittliche Laufzeit unter 90 Sekunden zu halten und die maximale Spitzenzeit nicht über 4 Minuten zu halten.

Horizontale Skalierung (Anzahl der Ziele)

Die horizontale Skalierung ist durch die Anzahl der [aktiven Planungsziele](#) begrenzt, die verwaltet werden. Ein aktives Ziel ist eine account/region/service Kombination mit mindestens einer aktiv markierten Ressource. Instance Scheduler kann für viel mehr Konten und Regionen bereitgestellt werden, aber nur Ziele mit aktiv markierten Ressourcen wirken sich auf die Leistung aus.

Mit dem standardmäßigen Lambda-Parallelitätskontingent von 1000 können Sie 1000 aktive Ziele gleichzeitig ausführen. Lambda stellt zusätzliche Ausführungen automatisch in die Warteschlange, sodass Sie über dieses Limit hinaus skalieren können. Für eine optimale Leistung empfehlen wir, die kumulative Verzögerung unter 3 Minuten zu halten.

Bei einer durchschnittlichen Laufzeit von 15 Sekunden und einem Lambda-Parallelitätslimit von 1000 können Sie beispielsweise bis zu 12000 aktive Ziele verwalten und gleichzeitig die kumulierte Verzögerung unter 3 Minuten halten ($3 \text{ Minuten} \div 15 \text{ Sekunden} \times 1000 = 12000 \text{ Ziele}$).

Für größere Bereitstellungen können Sie beim AWS-Support eine Erhöhung des Lambda-Parallelitätskontingents beantragen.

Weitere Überlegungen

AWS-Ressourcen-Tags: AWS-Ressourcen haben in der Regel ein Limit von 50 Tags pro Ressource. Instance Scheduler benötigt 6 Informations- und Kontrolltags für den Betrieb der Lösung. Stellen Sie sicher, dass Ihre Ressourcen über eine ausreichende Tag-Kapazität verfügen, um sowohl die Instance Scheduler-Tags als auch Ihre bestehende Tagging-Strategie zu berücksichtigen.

Lambda-Ausführungslimits: Jede Lambda-Funktion des Scheduling Request Handlers hat ein Ausführungstimeout von 5 Minuten.

DynamoDB-Skalierung: Die Lösung verwendet On-Demand-Skalierung für ihre [Amazon DynamoDB-Tabellen](#) und passt die Kapazität automatisch an Ihre Arbeitslast an.

API-Ratenbegrenzungen: Bei sehr großen Bereitstellungen kann es zu einer Drosselung der AWS-Service-API kommen. Die Lösung umfasst eine Wiederholungslogik zur Behandlung vorübergehender Drosselungen. Eine übermäßige Drosselung kann jedoch die oberen Skalierungsgrenzen der Lösung verringern.

AWS-Servicekontingente

Servicekontingente für AWS-Services

Service Quotas, auch als Limits bezeichnet, sind die maximale Anzahl von Serviceressourcen oder -vorgängen für Ihr AWS-Konto. Stellen Sie sicher, dass Sie über ein ausreichendes Kontingent für jeden der in dieser Lösung implementierten Services verfügen. Weitere Informationen finden Sie unter [AWS-Servicekontingente](#).

CloudFormation AWS-Kontingente

Ihr AWS-Konto verfügt über CloudFormation AWS-Kontingente, die Sie beachten sollten, wenn Sie den Stack in dieser Lösung starten. Wenn Sie diese Kontingente verstehen, können Sie Limitationsfehler vermeiden, die Sie daran hindern würden, diese Lösung erfolgreich einzusetzen. Weitere Informationen finden Sie unter [CloudFormation AWS-Kontingente](#) im CloudFormation AWS-Benutzerhandbuch.

AWS Lambda Lambda-Kontingente

Ihr Konto hat ein standardmäßiges AWS Lambda Concurrent Execution-Kontingent von 1000. Für größere Bereitstellungen empfehlen wir, Instance Scheduler auf einem dedizierten Konto bereitzustellen, um zu vermeiden, dass mit anderen Workloads um Lambda-Parallelität konkurriert. Dieser Wert ist einstellbar. Weitere Informationen finden Sie im [AWS Lambda Getting Started Guide](#).

Unterstützte AWS Regionen

Sie können Instance Scheduler in jeder AWS-Region bereitstellen, einschließlich AWS-Regionen GovCloud (USA) und einiger [Opt-in-Regionen](#) (Regionen, die standardmäßig deaktiviert sind). Nachdem Sie die Lösung bereitgestellt haben, können Sie sie so konfigurieren, dass die entsprechenden Start- oder Stoppaktionen auf markierte EC2- und RDS-DB-Instances in allen Regionen Ihres Kontos angewendet werden. Wenn Sie die kontenübergreifende Instance-Planung verwenden, wendet die Lösung Aktionen auf Instances in allen konfigurierten Regionen in allen Konten an.

⚠ Important

Instance Scheduler on AWS-Aktionen wirken sich auf entsprechend markierte Instances in allen AWS-Regionen Ihres Kontos aus, obwohl die Lambda-Funktion in einer einzigen Region ausgeführt wird.

Sie können mehrere Bereitstellungen der Lösung verwenden, um eine große Anzahl von Instances oder Instances in vielen Konten und Regionen zu planen. Wenn Sie mehrere Scheduler bereitstellen, verwenden Sie für jeden Stack einen anderen Tag-Namen und konfigurieren Sie für jede Bereitstellung eine Reihe von Regionen, die sich nicht überschneiden.

Bei jeder Bereitstellung wird jede Instanz in jeder konfigurierten Region in einem Konto auf den Tag-Schlüssel überprüft, der die Ressourcen identifiziert, die geplant werden sollen. Wenn sich die Regionen für mehrere Bereitstellungen überschneiden, wird jede Instanz von mehreren Bereitstellungen geprüft.

ℹ Note

Instance Scheduler auf AWS kann für die Planung auf Instances in jeder Opt-in-Region abzielen, selbst wenn die Lösungstapel in AWS-Standardregionen bereitgestellt werden.

Kontenübergreifende Planung von Instances mithilfe eines Kontos IDs oder einer AWS-Organisations-ID

Diese Lösung umfasst eine Vorlage ([instance-scheduler-on-aws-remote.template](#)), die die [AWS Identity and Access Management \(IAM\) -Rollen und](#) andere notwendige Ressourcen erstellt, damit die Lösung mit der Planung in den sekundären Konten beginnen kann. Sie können die Berechtigungen in der Remote-Vorlage überprüfen und ändern, bevor Sie den Stack starten.

Aktivieren Sie die kontenübergreifende Planung mithilfe von Account IDs

So wenden Sie automatische Start-Stopp-Zeitpläne auf Ressourcen in sekundären Konten an:

1. Melden Sie sich bei der [AWS-Managementkonsole](#) an und klicken Sie auf die Schaltfläche, um die [instance-scheduler-on-aws](#) CloudFormation AWS-Vorlage im Hauptkonto zu starten.

2. Starten Sie die Remote-Vorlage ([instance-scheduler-on-aws-Remote](#)) in jedem entsprechenden sekundären Konto. Wenn jeder Remote-Stack gestartet wird, erstellt er eine kontenübergreifende Rolle — Amazon Resource Name (ARN).
3. Aktualisieren Sie den primären Lösungstapel mit der Konto-ID in den IDs Parametern Provide Organization ID oder List of Remote Account, damit die Lösung Start- und Stoppaktionen für Instances in den sekundären Konten ausführen kann.

Aktivierung der kontenübergreifenden Terminplanung mithilfe der AWS-Organisations-ID

So wenden Sie automatisierte Start-Stopp-Zeitpläne auf Ressourcen in sekundären Konten an:

1. Melden Sie sich bei der [AWS-Managementkonsole](#) an und klicken Sie auf die Schaltfläche, um die [instance-scheduler-on-aws](#) CloudFormation AWS-Vorlage im Hauptkonto zu starten.
2. Legen Sie den CloudFormation Parameter Using AWS Organizations fest? auf Ja und geben Sie die Organisations-ID in den IDs CloudFormation Parametern Organization-ID angeben ODER Liste der Remote-Konten ein.
3. Nachdem Sie den Stack im primären Konto bereitgestellt haben, starten Sie die Remote-Vorlage (`instance-scheduler-on-aws-remote`) in jedem entsprechenden sekundären Konto in derselben Region wie die Lösung im primären Konto. Wenn jeder Remote-Stack erfolgreich gestartet wurde, wird das primäre Lösungskonto mit der Konto-ID aktualisiert, ohne dass weitere Änderungen am primären Konto vorgenommen werden.

Konto IDs mit dem AWS Systems Manager Parameter Store verwalten

Verwenden Sie den AWS Systems Manager Parameter Store, um das Remote-Konto zu speichern IDs. Sie können das Remote-Konto IDs als Listenparameter speichern, wobei jedes Element eine Konto-ID ist, oder als Zeichenkettenparameter, der eine durch Kommas getrennte Liste von Remote-Konten enthält. IDs Der Parameter hat das Format `{param:_name_}`, wobei der Name dem Namen des Parameters im Parameter Store entspricht.

Um diese Funktion nutzen zu können, müssen Sie den Instance Scheduler auf dem AWS-Hub-Stack in demselben Konto wie Ihr Parameterspeicher starten.

Dienste, die für die Planung unterstützt werden

Instance Scheduler auf AWS unterstützt derzeit die Planung der folgenden Services:

- Amazon EC2
- Amazon EC2 Auto-Scaling-Gruppen
- Amazon RDS
- Amazon Aurora Aurora-Cluster
- Amazon DocumentDB
- Amazon Neptune

Verhalten beim Herunterfahren der Instanz

Amazon EC2

Diese Lösung ist so konzipiert, dass EC2-Instances automatisch gestoppt werden. Dabei wird davon ausgegangen, dass das Verhalten beim Herunterfahren von Instances auf Stopp und nicht auf Terminate gesetzt ist. Beachten Sie, dass Sie eine Amazon EC2 EC2-Instance nicht neu starten können, nachdem sie beendet wurde.

Standardmäßig sind EC2-Instances so konfiguriert, dass sie beim Herunterfahren gestoppt und nicht beendet werden. Sie können [dieses Verhalten jedoch ändern](#). Stellen Sie daher sicher, dass die Instances, die Sie mit dem Instance Scheduler auf AWS steuern, mit dem Verhalten „Stopp Shutdown“ konfiguriert sind. Andernfalls werden sie beendet.

Amazon RDS, Amazon Neptune und Amazon DocumentDB

Diese Lösung wurde entwickelt, um RDS-, Neptune- und DocDB-Instances automatisch zu stoppen, nicht zu löschen. Sie können den CloudFormation AWS-Vorlagenparameter Create RDS Instance Snapshot verwenden, um Snapshots von RDS-DB-Instances zu erstellen, bevor die Lösung die Instances stoppt. Snapshots werden aufbewahrt, bis die Instance das nächste Mal gestoppt und ein neuer Snapshot erstellt wird.

Note

Snapshots sind für Amazon Aurora Aurora-Cluster nicht verfügbar. Sie können den Vorlagenparameter Schedule Aurora Clusters verwenden, um RDS-DB-Instances zu starten

und zu stoppen, die Teil eines Aurora-Clusters sind oder Aurora-Datenbanken verwalten. Sie müssen den Cluster (nicht die einzelnen Instances) mit dem Tag-Schlüssel, den Sie bei der Erstkonfiguration definiert haben, und dem Namen des Zeitplans als Tag-Wert kennzeichnen, um diesen Cluster zu planen.

Weitere Informationen zu Einschränkungen beim Starten und Stoppen einer RDS-DB-Instance finden Sie unter [Vorübergehendes Stoppen einer Amazon RDS-DB-Instance](#) im Amazon RDS-Benutzerhandbuch.

Wenn eine RDS-DB-Instance gestoppt wird, wird der Cache geleert, was zu einer langsameren Leistung führen kann, wenn die Instance neu gestartet wird.

Amazon-RDS-Wartungsfenster

Jede RDS-DB-Instance hat ein wöchentliches [Wartungsfenster](#), in dem alle Systemänderungen vorgenommen werden. Während des Wartungsfensters startet Amazon RDS automatisch Instances, die länger als sieben Tage angehalten wurden, um Wartungsarbeiten durchzuführen. Amazon RDS stoppt die Instance nicht, sobald das Wartungsereignis abgeschlossen ist.

Mit der Lösung können Sie angeben, ob das bevorzugte Wartungsfenster einer RDS-DB-Instance als Laufzeit zu ihrem Zeitplan hinzugefügt werden soll. Die Lösung startet die Instance zu Beginn des Wartungsfensters und stoppt die Instance am Ende des Wartungsfensters, wenn keine andere Laufzeit vorgibt, dass die Instance ausgeführt werden soll, und wenn das Wartungsereignis abgeschlossen ist.

Wenn das Wartungsereignis bis zum Ende des Wartungsfensters nicht abgeschlossen ist, wird die Instanz bis zum Planungsintervall nach Abschluss des Wartungsereignisses ausgeführt. Weitere Informationen zum Amazon RDS-Wartungsfenster finden Sie unter [Wartung einer DB-Instance](#) im Amazon RDS-Benutzerhandbuch.

Amazon EC2 Auto-Scaling-Gruppen

Wir haben diese Lösung so konzipiert, dass Amazon EC2 Auto Scaling Scaling-Gruppen mithilfe von geplanten Skalierungsaktionen automatisch gestoppt werden. Sie können die Lösung verwenden, um geplante Skalierungsaktionen für die Auto Scaling Scaling-Gruppe (ASG) zu konfigurieren. Wenn eine ASG durch eine geplante Skalierungsaktion gestoppt wird, werden ihre Mindest-, Wunsch- und Maximalkapazität so 0 lange auf eingestellt, bis die ASG automatisch wieder gestartet wird. Dadurch

werden die minimalen, gewünschten und maximalen Kapazitäten auf ihre ursprünglichen Werte zurückgesetzt.

Architektur


Dieser Abschnitt enthält ein Referenzdiagramm zur Implementierungsarchitektur, [Überlegungen zum AWS Well-Architected-Design](#), [Sicherheitskomponenten](#), [Scheduler-Konfigurationen](#) und [AWS-Services](#), die in dieser Lösung verwendet werden.

Architekturdiagramm

Durch die Bereitstellung dieser Lösung werden die folgenden Komponenten in Ihrem AWS-Konto bereitgestellt.

Instance Scheduler in der AWS-Cloud

4. Eine DynamoDB-Registrierungstabelle verfolgt automatisch alle verwalteten Ressourcen. Wenn Ressourcen [für die Planung markiert](#) sind, werden sie als Reaktion auf AWS-Tagging-Ereignisse in dieser Tabelle registriert.
5. Jede Scheduling-Lambda-Funktion beschreibt markierte Ressourcen, wertet ihre Zeitpläne anhand der aktuellen Uhrzeit aus und führt die entsprechenden Start- oder Stoppaktionen aus.
 - a. Wenn bei EC2-Instances ein Startvorgang aufgrund unzureichender Kapazität fehlschlägt, kann die Lösung so konfiguriert werden, dass versucht wird, die Größe der Instance an [alternative Instance-Typen anzupassen, bevor der Startvorgang erneut](#) versucht wird.
6. Die Zeitplanverwaltung ist [über die DynamoDB-Konsole](#), das [Scheduler-CLI-Tool](#) oder benutzerdefinierte [CloudFormation AWS-Ressourcen](#) verfügbar. Die Lösung wird mit mehreren vorkonfigurierten Beispielzeitplänen bereitgestellt.
7. Kontoübergreifende Bereitstellungen verwenden eine Hub-Spoke-Architektur, bei der sich Spoke-Konten automatisch selbst beim Hub-Konto registrieren. Spoke-Stacks müssen in derselben Region wie der Hub-Stack bereitgestellt werden und müssen entweder vom Hub-Stack oder von Mitgliedern derselben [AWS-Organisation](#) vorab genehmigt werden.
8. Die Lösung veröffentlicht [Planungs- und Registrierungsereignisse](#) für EventBridge Busse im Hub-Konto (globale Ereignisse) und Spoke-Konten (lokale Ereignisse pro Region).

 Note

CloudFormation AWS-Ressourcen werden aus [\(AWS CDK-\)](#) Konstrukten erstellt.

Alle von dieser Lösung verwendeten Lambda-Funktionen nutzen AWS IAM für die Berechtigungsanforderungen für Ihre Ressourcen und AWS KMS für die Verschlüsselung von [Amazon Simple Notification Service \(Amazon SNS SNS-Thema\)](#) und DynamoDB-Tabellen.

Jedes Mal, wenn die Lösung ein Planungsintervall durchführt, vergleicht sie den aktuellen Status jeder entsprechend markierten Instance mit dem Zielstatus (definiert durch einen oder mehrere [Perioden](#) in einem Zeitplan im Instance-Tag) im zugehörigen Zeitplan. Das Zeitplanintervall wendet dann je nach Bedarf die entsprechende Start- oder Stoppaktion an.

Wenn die Lambda-Funktion beispielsweise an einem Freitag um 9 Uhr (ET) aufgerufen wird und sie eine gestoppte EC2- oder RDS-DB-Instance mit dem Tag `schedule=office-hours` identifiziert, sucht sie in Amazon DynamoDB nach den Konfigurationsdetails für den Bürozeitplan. Wenn der

Bürozeitplan einen Zeitraum enthält, der angibt, dass die Instance montags bis freitags von 9.00 Uhr ET bis 17.00 Uhr ET ausgeführt werden soll, startet die Lambda-Funktion diese Instanz.

Die Lambda-Funktion zeichnet auch Informationen über Ihre Ressourcen auf und zeigt sie in einem optionalen [Amazon CloudWatch Custom-Dashboard](#) an. Zu den aufgezeichneten Informationen gehören die Anzahl der für jeden Zeitplan markierten Instances, die Größe dieser Instances und ob sich diese Instances derzeit in einem laufenden oder gestoppten Zustand befinden oder nicht. Weitere Informationen zu diesem benutzerdefinierten Dashboard finden Sie unter [Operational Insights Dashboard](#).

Note

Das Stoppen einer Amazon EC2 Instance unterscheidet sich vom Beenden einer Amazon EC2 Instance. Standardmäßig sind Amazon EC2 Instances so konfiguriert, dass sie beim Herunterfahren gestoppt und nicht beendet werden. Sie können dieses Verhalten jedoch ändern. Bevor Sie diese Lösung verwenden, stellen Sie sicher, dass die Instances so eingestellt sind, dass sie je nach Bedarf gestoppt oder beendet werden.

Überlegungen zum AWS-Well-Architected-Design

Wir haben diese Lösung mit Best Practices aus dem [AWS Well-Architected Framework](#) entwickelt, das Kunden dabei unterstützt, zuverlässige, sichere, effiziente und kostengünstige Workloads in der Cloud zu entwerfen und zu betreiben.

In diesem Abschnitt wird beschrieben, wie die Entwurfsprinzipien und Best Practices des Well-Architected Framework bei der Erstellung dieser Lösung angewendet wurden.

Operative Exzellenz

In diesem Abschnitt wird beschrieben, wie wir diese Lösung unter Verwendung der Prinzipien und bewährten Verfahren des Pfeilers [Operational Excellence](#) konzipiert haben.

- Die Lösung überträgt Metriken an Amazon CloudWatch, um die Beobachtbarkeit der Komponenten (wie Infrastruktur und Lambda-Funktionen) zu gewährleisten.
- AWS X-Ray verfolgt Lambda-Funktionen.
- Verwendet Amazon SNS für die Fehlerberichterstattung.

Sicherheit

In diesem Abschnitt wird beschrieben, wie wir diese Lösung unter Verwendung der Prinzipien und bewährten Methoden der [Sicherheitssäule](#) konzipiert haben.

- Für die gesamte Kommunikation zwischen den Diensten werden IAM-Rollen verwendet.
- Für die gesamte Kommunikation mit mehreren Konten werden IAM-Rollen verwendet.
- Für alle Rollen, die von der Lösung verwendet werden, gilt der Zugriff mit den geringsten Rechten. Mit anderen Worten, sie enthalten nur die Mindestberechtigungen, die erforderlich sind, damit der Dienst ordnungsgemäß funktionieren kann.
- Alle Datenspeicher, einschließlich DynamoDB-Tabellen, verfügen über eine Verschlüsselung im Ruhezustand.

Zuverlässigkeit

In diesem Abschnitt wird beschrieben, wie wir diese Lösung unter Verwendung der Prinzipien und bewährten Verfahren der Zuverlässigkeitskomponente konzipiert haben.

- Die Lösung verwendet, wo immer möglich, serverlose AWS-Services (wie Lambda und DynamoDB), um eine hohe Verfügbarkeit und Wiederherstellung nach einem Serviceausfall sicherzustellen.
- Die Datenverarbeitung verwendet Lambda-Funktionen. Die Lösung speichert Daten in DynamoDB, sodass sie standardmäßig in mehreren Availability Zones gespeichert werden.

Leistungseffizienz

In diesem Abschnitt wird beschrieben, wie wir diese Lösung unter Verwendung der Prinzipien und bewährten Verfahren des Pfeilers Leistungseffizienz konzipiert haben.

- Die Lösung verwendet eine serverlose Architektur.
- Sie können die Lösung in jeder AWS-Region starten, die die in dieser Lösung verwendeten AWS-Services unterstützt (wie Lambda und DynamoDB). Einzelheiten finden Sie unter [Unterstützte AWS-Regionen](#).
- Die Lösung wird täglich automatisch getestet und bereitgestellt. Unsere Lösungsarchitekten und Fachexperten überprüfen die Lösung auf Bereiche, in denen experimentiert und verbessert werden muss.

Kostenoptimierung

In diesem Abschnitt wird beschrieben, wie wir diese Lösung unter Verwendung der Prinzipien und bewährten Verfahren des [Pfeilers Kostenoptimierung](#) konzipiert haben.

- Die Lösung verwendet eine serverlose Architektur, und Kunden zahlen nur für das, was sie tatsächlich nutzen.
- Die Rechenschicht ist standardmäßig auf Lambda eingestellt, das ein pay-per-use Modell verwendet.

Nachhaltigkeit

In diesem Abschnitt wird beschrieben, wie wir diese Lösung unter Verwendung der Prinzipien und bewährten Verfahren der Säule Nachhaltigkeit konzipiert haben.

- Die Lösung verwendet verwaltete und serverlose Dienste, um die Umweltbelastung durch die Back-End-Dienste zu minimieren.
- Das serverlose Design der Lösung zielt darauf ab, den CO2-Fußabdruck im Vergleich zu dem Fußabdruck kontinuierlich betriebener Server vor Ort zu reduzieren.

Scheduler-Konfigurationstabelle

Bei der Bereitstellung erstellt Instance Scheduler auf AWS eine Amazon DynamoDB-Tabelle, die globale Konfigurationseinstellungen enthält.

Globale Konfigurationselemente enthalten ein Typattribut mit dem Wert config in der Konfigurationstabelle. Zeitpläne und Perioden enthalten Typattribute mit Werten für Zeitplan bzw. Zeitraum. Mithilfe der DynamoDB-Konsole oder der [Befehlszeilenschnittstelle](#) der Lösung können Sie Zeitpläne und Zeiträume zur Konfigurationstabelle hinzufügen, aktualisieren oder daraus entfernen. Sie bearbeiten jedoch keine Elemente mit einem bestimmten Konfigurationstyp, da diese Elemente von der Lösung verwaltet werden.

Planer-CLI

Die Lösung umfasst eine CLI, die Befehle zur Konfiguration von Zeitplänen und Zeiträumen bereitstellt. Mit der CLI können Sie die Kosteneinsparungen für einen bestimmten Zeitplan abschätzen. Die von der Schedule CLI bereitgestellten Kostenschätzungen dienen nur ungefähren

Zwecken. Weitere Informationen zur Konfiguration und Verwendung der Scheduler-CLI finden Sie unter [Scheduler CLI](#).

In dieser Lösung verwendete AWS-Services

AWS Service	Description
AWS Lambda	Kern. Solution stellt eine Lambda-Funktion bereit, die die gesamte Logik zum Planen der Instanzen und zum Verwalten von Updates für den CloudFormation Stack mithilfe einer benutzerdefinierten Ressourcenfunktion enthält.
Amazon-DynamoDB	Kern. Die Lösung erstellt DynamoDB-Tabellen zum Speichern der Zeitplankonfiguration, der Statusinformationen und der zuletzt ausgeführten Aktionen der Instanzen sowie eine Tabelle zum Speichern des Systems Manager Manager-Wartungsfensters für Planungszwecke.
Amazon CloudWatch	Kern. Die Lösung speichert Debugging- und Informationsprotokolle.
AWS IAM	Kern. Die Lösung verwendet IAM, um Berechtigungen für die Planung von Instanzen zu erhalten.
Amazon SNS	Kern. Solution erstellt ein SNS-Thema, um den Benutzern Fehlermeldungen zu senden, die sie abonnieren und im Falle von Fehlern beheben können.
AWS KMS	Kern. Die Lösung erstellt einen AWS-KMS-Schlüssel zur Verschlüsselung des SNS-Themas.

AWS Service	Description
Amazon EventBridge	Kern. Die Lösung erstellt eine EventBridge Lösung erstellt EventBridge geplante Regeln, die AWS Lambda in einem konsistenten Intervall aufrufen.“
AWS Systems Manager	Unterstützend. Ermöglicht die Überwachung von Ressourcen auf Anwendungsebene und die Visualisierung von Ressourcenoperationen und Kostendaten.
Amazon EC2	Geplant. Die Lösung wird zum Starten und Stoppen von EC2-Instances verwendet. Die Instanzen werden durch spezifische Tags identifiziert key/values , die in der Lösung konfiguriert sind.
Amazon RDS	Geplant. Die Lösung wird verwendet, um den Status von RDS-DB-Instances auf Verfügbar oder Gestoppt zu ändern. Die Instances werden durch spezifische Tags identifiziert key/values , die in der Lösung konfiguriert sind.
Amazon Aurora	Geplant. Die Lösung wird verwendet, um den Status von Aurora-Clustern auf Verfügbar oder Gestoppt zu ändern. Die Cluster werden durch spezifische Tags identifiziert key/values , die in der Lösung konfiguriert sind.
Amazon Neptune	Geplant. Die Lösung wird verwendet, um den Status der Neptune-Instanzen auf Verfügbar oder Gestoppt zu ändern. Die Instanzen werden durch spezifische Tags identifiziert key/values , die in der Lösung konfiguriert sind.

AWS Service	Description
Amazon DocumentDB	Geplant. Die Lösung wird verwendet, um den Status der DocumentDB-Instanzen auf Verfügbar oder Gestoppt zu ändern. Die Instanzen werden durch spezifische Tags identifiziert key/values , die in der Lösung konfiguriert sind.
Amazon EC2 Auto Scaling Scaling-Gruppen	Geplant. Die Lösung wird verwendet, um geplante Skalierungsregeln für EC2 Auto Scaling Scaling-Gruppen zu verwalten. Diese Regeln werden start/stop Auto Scaling Scaling-Gruppen gemäß einem zugehörigen Zeitplan gruppieren. Gruppen werden durch spezifische Tags identifiziert key/values , die in der Lösung konfiguriert sind.

Sicherheit

Wenn Sie Systeme auf der AWS-Infrastruktur aufbauen, werden Sie und AWS gemeinsam für die Sicherheit verantwortlich sein. Dieses [Modell der geteilten Verantwortung](#) reduziert Ihren betrieblichen Aufwand, da AWS die Komponenten wie das Host-Betriebssystem, die Virtualisierungsebene und die physische Sicherheit der Einrichtungen, in denen die Services betrieben werden, betreibt, verwaltet und kontrolliert. Weitere Informationen zur AWS-Sicherheit finden Sie unter [AWS Cloud Security](#).

AWS KMS

Die Lösung erstellt einen vom Kunden verwalteten AWS-Schlüssel, der zur Konfiguration der serverseitigen Verschlüsselung für das SNS-Thema und die DynamoDB-Tabellen verwendet wird.

Amazon IAM

Die Lambda-Funktionen der Lösung erfordern Berechtigungen für den Zugriff auf Hub-Kontoressourcen und Zugriff auf get/put Systems Manager Manager-Parameter, Zugriff auf CloudWatch Protokollgruppen, den AWS KMS KMS-Schlüssel encryption/decryption, and publish

messages to SNS. In addition, Instance Scheduler will also create Scheduling Roles in all managed accounts that will provide access to start/stop EC2, RDS, Autoscaling-Ressourcen, DB-Instances, das Ändern von Instance-Attributen und das Aktualisieren von Tags für diese Ressourcen. Alle erforderlichen Berechtigungen werden von der Servicerolle Solution to Lambda bereitgestellt, die als Teil der Lösungsvorlage erstellt wurde.

Bei der Bereitstellung stellt Instance Scheduler IAM-Rollen mit eingeschränktem Geltungsbereich für jede seiner Lambda-Funktionen zusammen mit Scheduler-Rollen bereit, die nur von bestimmten Scheduling-Lambdas in der bereitgestellten Hub-Vorlage übernommen werden können. Diese Schedule-Rollen werden Namen haben, die dem Muster folgen, und. `{namespace}-Scheduler-Role {namespace}-ASG-Scheduling-Role`

Ausführliche Informationen zu den Berechtigungen, die den einzelnen Servicerollen gewährt werden, finden Sie in den [CloudFormation Vorlagen](#).

Verschlüsselte EC2-EBS-Volumes

Wenn Sie EC2-Instances planen, die an mit AWS KMS verschlüsselte EBS-Volumes angehängt sind, müssen Sie Instance Scheduler die Erlaubnis erteilen, die zugehörigen AWS-KMS-Schlüssel zu verwenden. Dadurch kann Amazon EC2 die angehängten EBS-Volumes während der gestarteten Funktion entschlüsseln. Diese Berechtigung muss der Scheduling-Rolle in demselben Konto wie die EC2-Instance (s) erteilt werden, die den Schlüssel verwenden.

Um die Erlaubnis zur Verwendung eines AWS-KMS-Schlüssels mit Instance Scheduler zu erteilen, fügen Sie den ARN des AWS-KMS-Schlüssels zum Instance Scheduler-Stack (Hub oder Spoke) in demselben Konto wie die EC2-Instance (en) hinzu, indem Sie die Schlüssel (n) verwenden:

KMS-Schlüssel-ARNS für EC2

Kms Key Arns for EC2

comma-separated list of kms arns to grant Instance Scheduler kms:CreateGrant permissions to provide the EC2 service with Decrypt permissions for encrypted EBS volumes. This allows the scheduler to start EC2 instances with attached encrypted EBS volumes. provide just (*) to give limited access to all kms keys, leave blank to disable. For details on the exact policy created, refer to security section of the implementation guide (<https://aws.amazon.com/solutions/implementations/instance-scheduler-on-aws/>)

Enter CommaDelimitedList

Dadurch wird automatisch die folgende Richtlinie generiert und der Scheduling-Rolle für dieses Konto hinzugefügt:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringLike": {
          "kms:ViaService": "ec2.*.amazonaws.com"
        },
      },
      "Null": {
        "kms:EncryptionContextKeys": "false",
        "kms:GrantOperations": "false"
      },
      "ForAllValues:StringEquals": {
        "kms:EncryptionContextKeys": [
          "aws:ebs:id"
        ],
        "kms:GrantOperations": [
          "Decrypt"
        ]
      },
      "Bool": {
        "kms:GrantIsForAWSResource": "true"
      },
      "Action": "kms:CreateGrant",
      "Resource": [
        "Your-KMS-ARNs-Here"
      ],
      "Effect": "Allow"
    }
  ]
}
```

EC2-Lizenzmanager

Bei der Planung von EC2-Instances, die in AWS License Manager verwaltet werden, müssen Sie Instance Scheduler die Erlaubnis erteilen, die zugehörigen Lizenzkonfigurationen zu verwenden. Auf diese Weise kann die Lösung Instances ordnungsgemäß starten und beenden und gleichzeitig die Lizenzbestimmungen einhalten. Diese Berechtigung muss der Scheduling-Rolle in demselben Konto erteilt werden wie die EC2-Instance (en), die License Manager verwenden.

Um die Erlaubnis zur Verwendung von AWS License Manager mit Instance Scheduler zu erteilen, fügen Sie die License Manager Manager-Konfigurations-ARNs zum Instance Scheduler-Stack (Hub oder Spoke) in demselben Konto hinzu wie die EC2-Instance (s), die License Manager verwenden:

License Manager Manager-Konfiguration ARNs für EC2

License Manager Arns for EC2

comma-separated list of license manager arns to grant Instance Scheduler ec2:StartInstance permissions to provide the EC2 service with license manager permissions to start the instances. This allows the scheduler to start EC2 instances with license manager configuration enabled. Leave blank to disable. For details on the exact policy created, refer to security section of the implementation guide (<https://aws.amazon.com/solutions/implementations/instance-scheduler-on-aws/>)

Enter CommaDelimitedList

Dadurch wird automatisch die folgende Richtlinie generiert und der Scheduling-Rolle für dieses Konto hinzugefügt:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ec2:StartInstances",
      "Resource": [
        "Your-License-Manager-ARNs-Here"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Weitere Informationen zu License Manager-Berechtigungen finden Sie unter [Identitäts- und Zugriffsmanagement für AWS License Manager](#) im AWS License Manager Manager-Benutzerhandbuch.

Erste Schritte

Dieses Handbuch enthält einen kurzen Überblick und Anweisungen zur schnellen Implementierung der Lösung. Diese Lösung verwendet [CloudFormation AWS-Vorlagen und -Stacks](#), um ihre Bereitstellung zu automatisieren. Die CloudFormation Vorlagen spezifizieren die in dieser Lösung enthaltenen AWS-Ressourcen und ihre Eigenschaften. Der CloudFormation Stack stellt die Ressourcen bereit, die in den Vorlagen beschrieben sind.

Überblick über den Bereitstellungsprozess

Important

Diese Lösung beinhaltet eine Option zum Senden anonymisierter Betriebsmetriken an AWS. Wir verwenden diese Daten, um besser zu verstehen, wie Kunden diese Lösung und die damit verbundenen Services und Produkte nutzen. AWS ist Eigentümer der im Rahmen dieser Umfrage gesammelten Daten. Die Datenerfassung unterliegt der [Datenschutzerklärung](#).

Um diese Funktion zu deaktivieren, laden Sie die Vorlage herunter, ändern Sie den Abschnitt CloudFormation AWS-Zuordnung und verwenden Sie dann die CloudFormation AWS-Konsole, um Ihre aktualisierte Vorlage hochzuladen und die Lösung bereitzustellen.

Folgen Sie den step-by-step Anweisungen in diesem Abschnitt, um die Lösung zu konfigurieren und in Ihrem Konto bereitzustellen.

Bereitstellungszeit: Ungefähr 5-10 Minuten (ohne Konfiguration).

[Schritt 1: Starten Sie den Instance-Scheduler-Stack](#)

1. Starten Sie die CloudFormation AWS-Vorlage in Ihrem AWS-Konto.
2. Geben Sie Werte für die erforderlichen Parameter ein.
3. Überprüfen Sie die anderen Vorlagenparameter und passen Sie ihre Werte bei Bedarf an.

[Schritt 2 \(optional\): Starten Sie den Remote-Stack in sekundären Konten](#)

1. Starten Sie die CloudFormation AWS-Vorlage in Ihrem AWS-Konto.

2. Geben Sie Werte für die erforderlichen Parameter ein.

CloudFormation AWS-Vorlagen

Diese Lösung verwendet [CloudFormation AWS-Vorlagen und -Stacks](#), um ihre Bereitstellung zu automatisieren. Die CloudFormation Vorlagen spezifizieren die in dieser Lösung enthaltenen AWS-Ressourcen und ihre Eigenschaften. Der CloudFormation Stack stellt die Ressourcen bereit, die in den Vorlagen beschrieben sind.

Sie können die CloudFormation Vorlagen für diese Lösung herunterladen, bevor Sie sie bereitstellen.

View template

[scheduler-on-aws](#).template — Verwenden Sie diese Vorlage, um die Lösung und alle zugehörigen Komponenten zu starten. Die Standardkonfiguration stellt eine AWS-Lambda-Funktion, eine Amazon DynamoDB-Tabelle, ein CloudWatch Amazon-Ereignis und CloudWatch benutzerdefinierte Metriken bereit. Sie können die Vorlage jedoch auch an Ihre spezifischen Bedürfnisse anpassen.

View template

instance-scheduler-on-aws-remote.template — Verwenden Sie diese Vorlage, um die kontenübergreifende Rolle zu starten, die von der Lösung verwendet wird, um Instances in Spoke-Konten zu planen. Bei Bereitstellungen mit AWS Organizations wird bei der Bereitstellung der Vorlage auch das Spoke-Konto beim Hub registriert, sodass keine manuelle Konfiguration erforderlich ist.

Note

Wenn Sie diese Lösung bereits bereitgestellt haben, finden Sie Anweisungen [zum Update unter Lösung](#) aktualisieren.

Schritt 1: Starten Sie den Instance Scheduler Hub-Stack

Folgen Sie den step-by-step Anweisungen in diesem Abschnitt, um die Lösung in Ihrem Konto bereitzustellen.

Bereitstellungszeit: ungefähr fünf Minuten

Launch solution

1. Melden Sie sich bei der [AWS-Managementkonsole](#) an und klicken Sie auf die Schaltfläche, um die AWS-Vorlage* instance-scheduler-on-aws .template* zu starten. CloudFormation
2. Die Vorlage wird standardmäßig in der Region USA Ost (Nord-Virginia) gestartet. Um die Lösung in einer anderen AWS-Region zu starten, verwenden Sie die Regionsauswahl in der Navigationsleiste der Konsole.
3. Vergewissern Sie sich auf der Seite Stack erstellen, dass sich die richtige Vorlagen-URL im Textfeld Amazon S3 S3-URL befindet, und wählen Sie Weiter.
4. Weisen Sie Ihrem Lösungsstapel auf der Seite „Stack-Details angeben“ einen Namen zu. Informationen zu Einschränkungen bei der Benennung von Zeichen finden Sie unter [IAM- und AWS STS STS-Kontingente](#) im AWS Identity and Access Management-Benutzerhandbuch.
5. Überprüfen Sie unter Parameter die Parameter für diese Lösungsvorlage und ändern Sie sie nach Bedarf. Diese Lösung verwendet die folgenden Standardwerte.

Parameter	Standard	Description
Infrastruktur		
Namespace	default	Geben Sie eine eindeutige Kennung an, um zwischen mehreren Lösungsbereitstellungen zu unterscheiden (keine Leerzeichen). Beispiel: Dev.
Verwenden Sie AWS Organizations	No	Verwenden Sie AWS Organizations, um die Registrierung von Spoke-Konten zu automatisieren.
ID/Remote Unternehmenskonto IDs	<Optional Input>	Wenn Sie AWS Organizations verwenden, ist dieses Feld erforderlich. Geben Sie die Organisations-ID an, zum Beispielo- xxxxyyy

Parameter	Standard	Description
		<p>. Andernfalls geben Sie eine durch Kommas getrennte Liste vertrauenswürdiger Spoke-Konten an IDs , die sich selbst für die Terminplanung registrieren können (maximal 40), z. B. 1111111111, 2222222222</p>
Schlüssel für das Zeitplan-Tag	Schedule	<p>Der Tag-Schlüssel, den die Lösung liest, um den Zeitplan für eine Ressource zu bestimmen. Der Wert einer Ressource gibt den Namen des Zeitplans an. Wenn Sie den Standardwert ändern möchten, weisen Sie ihm einen Namen zu, der auf einfache Weise einheitlich und korrekt auf alle erforderlichen Instanzen angewendet werden kann. Hinweis: Beim Tag-Schlüssel wird zwischen Groß- und Kleinschreibung unterschieden.</p>

Parameter	Standard	Description
Bewahren Sie Daten und Protokolle auf	Enabled	Aktivieren Sie den Löschschutz für DynamoDB-Tabellen, die von der Lösung verwendet werden. Dies führt dazu, dass die Tabellen beim Löschen dieses Stacks beibehalten werden. Um die Tabellen beim Löschen dieses Stacks zu löschen, deaktivieren Sie zunächst diesen Parameter.
Globale Einstellungen		
Scheduling aktivieren	Yes	Stellen Sie diese Option No ein, um alle Planungsvorgänge auszusetzen.
Standardzeitzone	UTC	Standard-Zeitzoneerkennung der IANA (International Assigned Numbers Authority) für Zeitpläne, die keine Zeitzone angeben. Eine Liste der gültigen Zeitzone-Identifikatoren finden Sie in der Spalte TZ-ID der Liste der Zeitzone der TZ-Datenbank .

Parameter	Standard	Description
Planungsintervall (Minuten)	5	Intervall in Minuten zwischen den Läufen des Scheduler s. Kürzere Intervalle erhöhen die Genauigkeit und Reaktionsfähigkeit, erhöhen aber auch die Kosten. Produktionsbereitstellungen benötigen für einen stabilen Betrieb mindestens 5 Minuten. Kürzere Werte gelten nur für Tests in kleinem Maßstab.
Aktivieren Sie die EC2 SSM-Wartungsfenster	No	Erlaubt Zeitplänen die Angabe eines oder mehrerer Namen für das Systems Manager-Wartungsfenster. Der Instance Scheduler auf AWS stellt dann sicher, dass Instances, die mit diesem Zeitplan gekennzeichnet sind, mindestens zehn Minuten vor den zugehörigen Wartungsfenstern gestartet werden.
Erstellen Sie beim Stopp RDS-Instance-Snapshots	No	Wählen Sie aus, ob vor dem Stoppen von RDS-DB-Instances ein Snapshot erstellt werden soll. Hinweis: Snapshots sind für Amazon Aurora Aurora-Cluster nicht verfügbar.

Parameter	Standard	Description
Präfix für den ASG-Aktionsnamen	IS-	Das Präfix, das die Lösung bei der Benennung von Scheduled Scaling-Aktionen für Auto Scaling Scaling-Gruppen verwendet. Aktionen mit diesem Präfix werden von der Lösung nach Bedarf hinzugefügt und entfernt.
Schlüssel für das geplante ASG-Tag	scheduled	Als veraltet gekennzeichnet. Dieser Parameter dient nur zu Migrationszwecken und sollte nicht bearbeitet werden.
Planung von Hub-Konten		
Region (en)	<Optional Input>	Liste der Regionen, in denen Instances geplant werden. Zum Beispiel <code>us-east-1</code> , <code>us-west-1</code> . HINWEIS: Wenn Sie diesen Parameter leer lassen, verwendet die Lösung die aktuelle Region.

Parameter	Standard	Description
KMS-Schlüssel-ARNs für EC2	<Optional Input>	Durch Kommas getrennte Liste von KMS-ARNs, die Instance Scheduler auf AWS kms gewähren sollen: CreateGrant Berechtigungen, um dem EC2-Service Entschlüsselungsberechtigungen für verschlüsselte EBS-Volumes bereitzustellen. Dadurch kann der Scheduler EC2-Instances mit angehängten verschlüsselten EBS-Volumes starten. Geben Sie (*) ein, um eingeschränkten Zugriff auf alle KMS-Schlüssel zu gewähren; lassen Sie das Feld leer, um es zu deaktivieren. Einzelheiten zur erstellten Richtlinie finden Sie unter Encrypted EC2 EBS Volumes .
License Manager Manager-ARNs für EC2	<Optional Input>	Durch Kommas getrennte Liste von License Manager-Konfigurations-ARNs, um Instance Scheduler-Berechtigungen zum Starten von EC2-Instances zu gewähren, die von License Manager verwaltet werden. Lassen Sie das Feld leer, um es zu deaktivieren. Einzelheiten finden Sie unter EC2 License Manager .

Parameter	Standard	Description
Überwachung		
Aktivieren Sie das Tagging mit Informationen	Yes	Wenn diese Option aktiviert ist, schreibt Instance Scheduler Informations-Tags auf verwaltete Ressourcen, die die zuletzt durchgeführte Planungsaktion und alle aufgetretenen Fehler angeben. Weitere Informationen finden Sie unter Informations-Tags .
CloudWatch Debug-Protokolle aktivieren	No	Aktivieren Sie die Protokollierung auf Debug-Ebene. CloudWatch
Aufbewahrungszeitraum für Protokolle (Tage)	30	Die Aufbewahrungsfrist für CloudWatch Protokolle in Tagen.
Operative Überwachung	Enabled	Stellen Sie ein Dashboard mit operativen Erkenntnissen bereit, um CloudWatch benutzerdefinierte Messdaten zum Betrieb der Lösung zu sammeln. Das Dashboard kann auf Wunsch deaktiviert werden, um die damit verbundenen Kosten zu senken.
Sonstige		

Parameter	Standard	Description
SchedulingRequestHandler Speichergöße (MB)	512	Die Speichergöße der AWS Lambda Lambda-Funktion, die Ressourcen plant. Erhöhen Sie den Wert, wenn Sie eine hohe Speicherauslastung oder Timeouts feststellen.
Größe des Orchestrator-Speichers (MB)	512	Die Speichergöße der Orchestrator-Lambda-Funktion. Erhöhen Sie den Wert, wenn Sie eine hohe Speicherauslastung oder Timeouts feststellen.

6. Wählen Sie Weiter aus.
7. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
8. Überprüfen und bestätigen Sie auf der Seite Überprüfen und erstellen die Einstellungen. Markieren Sie das Kästchen, um zu bestätigen, dass die Vorlage IAM-Ressourcen erstellt.
9. Wählen Sie Submit, um den Stack bereitzustellen.

Sie können den Status des Stacks in der CloudFormation AWS-Konsole in der Spalte Status anzeigen. Sie sollten in etwa fünf Minuten den Status CREATE_COMPLETE erhalten.

Schritt 2 (optional): Starten Sie den Remote-Stack in sekundären Konten

Important

Der Remote-Stack muss in derselben Region wie der Hub-Stack bereitgestellt werden.

Diese automatisierte CloudFormation AWS-Vorlage konfiguriert sekundäre Kontoberechtigungen, die es dem Hub-Stack ermöglichen, Instances in anderen Konten zu planen. Installieren Sie die Remote-Vorlage erst, nachdem der primary/hub Stack erfolgreich im Hub-Konto installiert wurde.

Launch solution

1. Melden Sie sich bei der AWS-Managementkonsole des entsprechenden Sekundärkontos an und klicken Sie auf die Schaltfläche, um die CloudFormation AWS-Vorlage instance-scheduler-on-aws-remote zu starten.
2. Die Vorlage wird standardmäßig in der Region USA Ost (Nord-Virginia) gestartet. Um die Lösung in einer anderen AWS-Region zu starten, verwenden Sie die Regionsauswahl in der Navigationsleiste der Konsole. Wenn der Hub-Stack für die Verwendung von AWS Organizations konfiguriert ist, stellen Sie die Remote-Vorlage in derselben Region wie der Hub-Stack bereit.
3. Vergewissern Sie sich auf der Seite Stack erstellen, dass sich die richtige Vorlagen-URL im Textfeld Amazon S3 S3-URL befindet, und wählen Sie Weiter.
4. Weisen Sie Ihrem Remote-Stack auf der Seite „Details angeben“ einen Namen zu.
5. Überprüfen Sie unter Parameter den Parameter für die Vorlage und ändern Sie ihn.
6. Wenn die Option AWS Organizations aktiviert ist und der Hub-Stack ähnlich konfiguriert ist, sind keine weiteren Änderungen im Haupt-Stack erforderlich, um die Planung zu starten.
7. Wenn die AWS-Organisationsoption auf Nein gesetzt ist, sollte der Hub-Stack mit der neuen Konto-ID aktualisiert werden.

Parameter	Standard	Description
Infrastruktur		
Namespace	default	Eindeutiger Bezeichner, der verwendet wird, um zwischen mehreren Lösungsbereitstellungen zu unterscheiden. Muss auf denselben Wert wie der Hub-Stack gesetzt werden.
Verwenden Sie AWS Organizations	No	Verwenden Sie AWS Organizations, um die

Parameter	Standard	Description
		Registrierung von Spoke-Konten zu automatisieren. Muss auf denselben Wert wie der Hub-Stack gesetzt sein.
Hub-Konto-ID	<Requires Input>	Konto-ID des Instance Schedulers auf dem AWS-Hub-Stack, der Ressourcen in diesem Konto plant.
Schlüssel für das Zeitplan-Tag	Schedule	Der Tag-Schlüssel, den die Lösung liest, um den Zeitplan für eine Ressource zu bestimmen. Muss auf denselben Wert wie der Hub-Stack gesetzt werden.
Planung der Mitgliedskonten		
Region (en)	<Optional Input>	Liste der Regionen, in denen Instances geplant werden. Zum Beispiel <code>us-east-1</code> , <code>us-west-1</code> . (Dies muss nicht dieselbe Regionsliste wie der Hub sein). Wenn Sie diesen Parameter leer lassen, verwendet die Lösung die aktuelle Region.

Parameter	Standard	Description
KMS-Schlüssel-ARNs für EC2	<Optional Input>	Durch Kommas getrennte Liste von KMS-ARNs, um der Lösung kms: CreateGrant Berechtigungen zu gewähren, dem EC2-Dienst Entschlüsselungsberechtigungen für verschlüsselte EBS-Volumen bereitzustellen. Dadurch kann der Scheduler EC2-Instances mit angehängten verschlüsselten EBS-Volumen starten. Geben Sie (*) ein, um eingeschränkten Zugriff auf alle KMS-Schlüssel zu gewähren; lassen Sie das Feld leer, um es zu deaktivieren. Einzelheiten finden Sie unter Encrypted EC2 EBS Volumes .
License Manager Manager-ARNs für EC2	<Optional Input>	Durch Kommas getrennte Liste von License Manager-Konfigurations-ARNs, um Instance Scheduler-Berechtigungen zum Starten von EC2-Instances zu gewähren, die von License Manager verwaltet werden. Lassen Sie das Feld leer, um es zu deaktivieren. Einzelheiten finden Sie unter EC2 License Manager .

1. Wählen Sie Weiter aus.
2. Wählen Sie auf der Seite Optionen Weiter aus.

3. Überprüfen und bestätigen Sie auf der Seite Überprüfen und erstellen die Einstellungen. Achten Sie darauf, das Kästchen anzukreuzen, das bestätigt, dass die Vorlage IAM-Ressourcen erstellt.
4. Wählen Sie Submit, um den Stack bereitzustellen.

Sie können den Status des Stacks in der CloudFormation AWS-Konsole in der Spalte Status anzeigen. CREATE_COMPLETE In etwa fünf Minuten sollte ein Status von angezeigt werden.

Konfigurieren Sie die Lösung

Nachdem die Lösung bereitgestellt wurde, können Sie damit beginnen, Zeitpläne zu konfigurieren und Instanzen für den Scheduler zu taggen. Weitere Informationen zu diesen Aufgaben finden Sie unter [Zeitpläne konfigurieren](#) und [Instanzen für die Planung taggen](#).

Leitfaden für Bediener

[Dieses Handbuch richtet sich an Benutzer und Betreiber dieser Lösung und enthält Einzelheiten zur Konfiguration von Zeitplänen und zur Überwachung der Lösung.](#)

Zeitpläne konfigurieren

Important

Falsch konfigurierte Zeitpläne können dazu führen, dass Instanzen kontinuierlich ausgeführt werden und unerwartete Kosten anfallen. Bevor Sie Zeitpläne auf Ihre Ressourcen anwenden, überprüfen Sie Folgendes:

- Der Zeitplannamen im Ressourcen-Tag entspricht exakt einem in der Konfigurationstabelle definierten Zeitplan. Falsch geschriebene oder nicht existierende Zeitplannamen führen zu einem `UnknownSchedule` Fehler und die Instanz wird nicht vom Scheduler gestoppt. Suchen Sie nach dem `IS-Error` Tag auf Ihren Ressourcen, um diesen Zustand zu identifizieren.
- Wenn auf gesetzt `stop_new_instances` ist `false`, werden Instances, die außerhalb des geplanten Zeitraums laufen, zu dem sie das erste Mal markiert wurden, erst beim nächsten geplanten Stop-Übergang gestoppt. Dies kann dazu führen, dass Instances länger als erwartet laufen.
- Wenn auf gesetzt `retain_running` ist `true`, werden Instances, die vor Beginn einer Laufzeit manuell gestartet wurden, am Ende dieses Zeitraums nicht gestoppt. Dies ist beabsichtigt, kann aber dazu führen, dass Instances unbegrenzt laufen, wenn sie nicht überwacht werden.
- Bei Verwendung `enforced: false` (Standardeinstellung) startet der Scheduler Instanzen, die während einer Laufzeit manuell gestoppt wurden, nicht neu und beendet auch keine Instances, die außerhalb einer Laufzeit manuell gestartet wurden, nach dem ersten Stopp-Übergang.

Wir empfehlen, [informatives Tagging](#) zu aktivieren (standardmäßig aktiviert) und die `IS-LastAction` Tags `IS-Error` und auf Ihren Ressourcen regelmäßig zu überprüfen, um sicherzustellen, dass die Planung wie erwartet funktioniert.

Sobald die Lösung erfolgreich bereitgestellt wurde, können Sie mit der Konfiguration von Zeitplänen beginnen. Instance Scheduler auf AWS unterstützt zwei Methoden zur Verwaltung von Zeitplänen, wie unten beschrieben.

Note

Die Lösung kann eine beliebige Anzahl von Zeitplänen unterstützen, von denen jeder einen oder mehrere Zeiträume enthalten kann, die definieren, wann Instances, die durch diesen Zeitplan gesteuert werden, ausgeführt werden sollen. Weitere Informationen finden Sie unter [Zeitpläne](#) und [Perioden](#).

Nutzung von Infrastructure as Code (empfohlen)

Instance Scheduler auf AWS bietet ein AWS CloudFormation CustomResource , mit dem Sie Ihre Zeitpläne und Zeiträume mithilfe von Infrastructure as Code (IaC) verwalten können.

Informationen zur Verwaltung von Zeitplänen mithilfe von IaC finden Sie unter [Zeitpläne mithilfe von Infrastructure as Code \(IaC\) verwalten](#).

Verwenden der Amazon DynamoDB DynamoDB-Konsole und des Instance Schedulers auf der AWS-CLI

Important

Wenn Sie die benutzerdefinierte Ressource verwendet haben, um Zeitpläne mithilfe von IaC zu verwalten, dürfen Sie die DynamoDB-Konsole oder die Scheduler-CLI nicht verwenden, um diese Zeitpläne oder ihre Perioden zu löschen oder zu ändern. Wenn Sie dies tun, entsteht ein Konflikt zwischen den gespeicherten Parametern in CloudFormation und den Werten in der Tabelle. Verwenden Sie außerdem keine Zeiträume, die von verwaltet werden, CloudFormation in Zeitplänen, die mit der DynamoDB-Konsole oder der Scheduler-CLI erstellt wurden.

Bei der Bereitstellung des Instance Schedulers auf dem AWS-Hub-Stack erstellte die Lösung eine Amazon DynamoDB-Tabelle mit mehreren Beispielzeiträumen und Zeitplänen, die Sie als Referenz verwenden können, um Ihre eigenen benutzerdefinierten Zeiträume und Zeitpläne zu erstellen. Um

einen Zeitplan in DynamoDB zu erstellen, ändern Sie einen der Zeitpläne in der Konfigurationstabelle (ConfigTable) oder erstellen Sie einen neuen. Um einen Zeitplan mit der CLI zu erstellen, [installieren Sie zuerst die Scheduler-CLI](#) und verwenden Sie dann die [verfügbaren Befehle](#).

Note

[Beispiele für die Erstellung mehrerer Beispielpäne mit IaC, DynamoDB und der InstanceScheduler CLI finden Sie unter Beispielpäne.](#)

Dieser Abschnitt enthält Anleitungen und Referenzen zur Verwendung, Überwachung und Aktualisierung der Lösung sowie Informationen zur Problembehandlung und zum Support.

Kennzeichnen Sie Instanzen für die Planung

Bei der Bereitstellung der CloudFormation AWS-Vorlage haben Sie den Namen (Tag-Schlüssel) für das benutzerdefinierte Tag der Lösung definiert. Damit Instance Scheduler auf AWS eine Amazon EC2- oder Amazon RDS-Instance erkennt, muss der Tag-Schlüssel auf dieser Instance mit diesem benutzerdefinierten Tag-Schlüssel übereinstimmen. Daher ist es wichtig, dass Sie Tags konsistent und korrekt auf alle zutreffenden Instances anwenden. Sie können weiterhin bestehende [bewährte Methoden für das Tagging](#) für Ihre Instances verwenden, während Sie diese Lösung verwenden. Weitere Informationen finden Sie unter [Taggen Ihrer Amazon EC2 EC2-Ressourcen](#) und [Taggen von Amazon RDS-Ressourcen](#).

Verwenden Sie in der AWS-Managementkonsole den [Tag Editor](#), um Tags für mehrere Ressourcen gleichzeitig anzuwenden oder zu ändern. Sie können Tags auch manuell in der Konsole anwenden und ändern.

Kurz nach dem Taggen einer Ressource wird der Ressource vom Instance Scheduler ein ManagedBy IS-Tag zugewiesen, das darauf hinweist, dass die Ressource jetzt vom Scheduler verwaltet wird. Sie können nach diesem Tag suchen, um zu bestätigen, dass die Ressource korrekt für die Planung registriert wurde.

Den Tag-Wert festlegen

Wenn Sie ein Tag auf eine Instance anwenden, verwenden Sie den Tag-Schlüssel, den Sie bei der Erstkonfiguration definiert haben (standardmäßig ist der Tag-Schlüssel Schedule), und legen Sie

Zeitzone

Sie können auch eine Zeitzone für den Zeitplan angeben. Wenn Sie keine Zeitzone angeben, verwendet der Zeitplan die Standardzeitzone, die Sie beim Starten der Lösung angegeben haben. Eine Liste der akzeptablen Zeitzonewerte finden Sie in der Spalte TZ der [Liste der Zeitzonen der TZ-Datenbank](#).

Feld „Neue Instanzen beenden“

Das Feld `stop_new_instances` steuert, ob der Instance Scheduler eine Instance stoppen soll, wenn sie zum ersten Mal für die Planung markiert wird, wenn sie gerade außerhalb einer Laufzeit läuft. Standardmäßig ist dieses Feld auf `true` gesetzt.

Wenn der Wert auf `true` gesetzt ist und Sie eine laufende Instance kennzeichnen, die sich außerhalb der geplanten Laufzeit befindet, stoppt Instance Scheduler die Instanz sofort. Wenn der Wert auf `false` gesetzt ist, lässt Instance Scheduler die Instance bis zur nächsten geplanten Stoppzeit laufen.

Feld „Ruhezustand“

Das Feld `Hibernate` ermöglicht es Ihnen, den Ruhezustand für gestoppte Amazon EC2 EC2-Instances zu verwenden. Wenn dieses Feld auf `true` gesetzt ist, müssen Ihre EC2-Instances ein Amazon Machine Image (AMI) verwenden, das den Ruhezustand unterstützt. Weitere Informationen finden Sie unter [Unterstütztes Linux AMIs](#) im Amazon EC2 EC2-Benutzerhandbuch. Hierdurch wird der Inhalt des Instance-Arbeitsspeichers (RAM) auf dem Amazon-Elastic-Block-Store (Amazon EBS)-Stamm-Volume gespeichert. Wenn dieses Feld auf `true` gesetzt ist, werden Instances nicht gestoppt, sondern in den Ruhezustand versetzt, wenn die Lösung sie stoppt.

Wenn Sie für die Lösung den Ruhezustand festlegen, Ihre Instances jedoch nicht für den Ruhezustand [konfiguriert sind oder die Voraussetzungen für den Ruhezustand nicht erfüllen, protokolliert die Lösung eine Warnung und die Instances werden ohne Ruhezustand](#) gestoppt. Weitere Informationen finden Sie unter [Hibernate your On-Demand-Instance oder Spot-Instance](#) im Amazon EC2 EC2-Benutzerhandbuch.

Feld erzwungen

Zeitpläne enthalten ein erzwungenes Feld, mit dem Sie verhindern können, dass eine Instanz außerhalb einer Laufzeit manuell gestartet oder während einer Laufzeit manuell gestoppt wird. Wenn dieses Feld auf „`true`“ gesetzt ist und ein Benutzer eine Instanz außerhalb der Laufzeit manuell

startet, stoppt die Lösung die Instanz. Wenn dieses Feld auf true gesetzt ist, wird eine Instanz auch neu gestartet, wenn sie während einer Laufzeit manuell gestoppt wird.

Feld „Laufend“ beibehalten

Das Feld `retain_running` verhindert, dass die Lösung eine Instanz am Ende einer Laufzeit stoppt, wenn die Instanz vor Beginn der Laufzeit manuell gestartet wurde. Wenn beispielsweise eine Instance mit einem Zeitraum von 9.00 Uhr bis 17.00 Uhr manuell vor 9.00 Uhr gestartet wird, stoppt die Lösung die Instance nicht um 17.00 Uhr.

Feld für das Wartungsfenster von Systems Manager (gilt nur für EC2-Instances)

Das `ssm-maintenance-window` Feld ermöglicht es Ihnen, AWS Systems Manager Manager-Wartungsfenster automatisch als Laufzeiten zu einem Zeitplan hinzuzufügen. Wenn Sie den Namen eines Wartungsfensters angeben, das in demselben Konto und derselben AWS-Region wie Ihre Amazon EC2 EC2-Instances existiert, startet die Lösung die Instance mindestens 10 Minuten vor dem Start des Wartungsfensters und stoppt die Instance am Ende des Wartungsfensters, sofern keine andere Laufzeit vorgibt, dass die Instance ausgeführt werden soll.

Sobald das SSM-Wartungsfenster erstellt und der Zeitplan mit dem Namen des SSM-Wartungsfensters konfiguriert ist, werden die Änderungen beim nächsten geplanten Lauf des Lambda übernommen. Wenn Sie beispielsweise eine Frequenz von 5 Minuten ausgewählt haben, in der der Scheduler Lambda ausgeführt werden soll, werden die Änderungen des Wartungsfensters vom Lambda im nächsten 5-minütigen Intervall übernommen.

Der Instance Scheduler auf AWS stellt sicher, dass Ihre Instances mindestens 10 Minuten vor Beginn des Wartungsfensters gestartet werden. Abhängig von dem Wert, den Sie für den CloudFormation AWS-Parameter `Scheduling Interval` festgelegt haben, kann dies dazu führen, dass Ihre Instance mindestens 10 Minuten vor Beginn des Wartungsfensters gestartet wird, um sicherzustellen, dass die Instance mindestens 10 Minuten früher startet. Wenn Sie das Planungsintervall beispielsweise auf 30 Minuten festlegen, startet der Scheduler die Instance zwischen 10 und 40 Minuten vor Beginn des Wartungsfensters.

Note

Um diese Funktion verwenden zu können, muss der CloudFormation `Windows-Parameter EC2 SSM Maintenance` aktivieren im Solution Hub-Stack auf gesetzt sein. `yes`

Weitere Informationen finden Sie unter [AWS Systems Manager Maintenance Windows](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Instance-Typ

Nur für Amazon EC2 EC2-Instances ermöglicht Ihnen ein Zeitplan, einen optionalen gewünschten Instance-Typ für jeden Zeitraum in einem Zeitplan anzugeben. Wenn Sie in diesem Zeitraum einen Instance-Typ angeben, passt die Lösung die Größe der EC2-Instances automatisch an den angeforderten Instance-Typ an.

Verwenden Sie die Syntax @, um einen Instance-Typ anzugeben. <period-name><instance-type>
Beispiel, weekends@t2.nano. Beachten Sie, dass, wenn Sie einen Instance-Typ für einen Zeitraum angeben, der Amazon EC2 EC2-Instances und Amazon RDS-Instances plant, der Instance-Typ für Amazon RDS-Instances ignoriert wird.

Wenn sich der Instance-Typ einer laufenden Instance von dem für den Zeitraum angegebenen Instance-Typ unterscheidet, stoppt die Lösung die laufende Instance und startet die Instance mit dem angegebenen Instance-Typ neu. Weitere Informationen finden Sie unter [Ändern des Instance-Typs](#) im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.

Definitionen planen

Die Konfigurationstabelle von Instance Scheduler on AWS in Amazon DynamoDB enthält Zeitplandefinitionen. Eine Zeitplandefinition kann die folgenden Felder enthalten:

Feld	Description
description	Eine optionale Beschreibung des Zeitplans.
hibernate	Wählen Sie aus, ob Amazon EC2-Instances, auf denen Amazon Linux ausgeführt wird, in den Ruhezustand versetzt werden sollen. Wenn dieses Feld auf true gesetzt ist, versetzt der Scheduler Instances in den Ruhezustand, wenn er sie stoppt. Beachten Sie, dass Ihre Instances den Ruhezustand aktivieren müssen und die Voraussetzungen für den Ruhezustand erfüllen müssen.

Feld	Description
enforced	Wählen Sie aus, ob der Zeitplan durchgesetzt werden soll. Wenn dieses Feld auf true gesetzt ist, stoppt der Scheduler eine laufende Instance, wenn sie außerhalb der Laufzeit manuell gestartet wird, oder er startet eine Instance, wenn sie während der Laufzeit manuell gestoppt wird.
name	Der Name, der zur Identifizierung des Zeitplans verwendet wird. Dieser Name muss eindeutig sein und darf nur alphanumerische Zeichen, Bindestriche (-) und Unterstriche (_) enthalten.
periods	<p>Der Name der Perioden, die in diesem Zeitplan verwendet werden. Geben Sie die Namen genau so ein, wie sie im Feld für den Periodennamen angezeigt werden.</p> <p>Sie können mit der Syntax @ <period-name> auch einen Instanztyp für den Zeitraum angeben <instance-type>. Beispiel, weekdays@t2.large .</p>
retain_running	Wählen Sie aus, ob verhindert werden soll, dass die Lösung eine Instanz am Ende einer Laufzeit stoppt, wenn die Instanz vor Beginn des Zeitraums manuell gestartet wurde.
use_maintenance_window	Wählen Sie, ob Sie das Amazon RDS-Wartungsfenster als Laufzeit in einen Amazon RDS-Instance-Zeitplan oder ein AWS Systems Manager Manager-Wartungsfenster als Laufzeit in einen Amazon EC2 EC2-Instance-Zeitplan aufnehmen möchten. Dieses Feld ist standardmäßig aktiviert und kann deaktiviert werden, indem der Wert auf „false“ gesetzt wird

Feld	Description
ssm_maintenance_window	<p>Wählen Sie aus, ob Sie Wartungsfenster von AWS Systems Manager als zusätzliche Laufzeit für diesen Zeitplan hinzufügen möchten. Akzeptiert mehrere Namen StringSet von Wartungsfenstern, die mit den Namen von Fenstern in denselben account/region wie bei geplanten EC2-Instances abgeglichen werden.</p> <p>Hinweis: Diese Funktion gilt nur für EC2-Instances.</p>
stop_new_instances	<p>Wählen Sie aus, ob eine Instance gestoppt werden soll, wenn sie zum ersten Mal markiert wird, wenn sie außerhalb der Laufzeit läuft. Standardmäßig ist dieses Feld auf true gesetzt.</p>
timezone	<p>Die Zeitzone, die der Zeitplan verwenden wird. Wenn keine Zeitzone angegeben ist, wird die Standardzeitzone (UTC) verwendet. Eine Liste der akzeptablen Zeitzonewerte finden Sie in der Spalte TZ der Liste der Zeitzonen der TZ-Datenbank.</p>
use_metrics	<p>Wählen Sie aus, ob CloudWatch Metriken auf Zeitplanebene aktiviert werden sollen. Dieses Feld überschreibt die CloudWatch Metrikeinstellungen, die Sie bei der Bereitstellung angegeben haben.</p> <p>Hinweis: Wenn Sie diese Funktion aktivieren, fallen Gebühren in Höhe von 0,90 USD/Monat pro Zeitplan oder gebuchter Leistung an.</p>

Referenz für den Zeitraum

Perioden enthalten Bedingungen, anhand derer Sie die spezifischen Stunden, Tage und Monate festlegen können, in denen eine Instance ausgeführt werden soll. Ein Zeitraum kann mehrere Bedingungen enthalten, aber alle Bedingungen müssen erfüllt sein, damit der Instance Scheduler auf AWS die entsprechende Start- oder Stoppaktion anwendet.

Start- und Stoppzeiten

Die `endtime` Felder `begintime` und definieren, wann der Instance Scheduler auf AWS Instances startet und stoppt. Wenn Sie nur eine Startzeit angeben, muss die Instance manuell gestoppt werden. Beachten Sie, dass die Lösung anhand dieses Werts bestimmt, wann die Instance gestoppt werden soll, wenn Sie im Feld [Wochentage](#) einen Wert angeben. Wenn Sie beispielsweise einen Wert `begintime` von 9 Uhr ohne `endtime` und einen Wochentagswert von Montag bis Freitag angeben, wird die Instance am Freitag um 23:59 Uhr gestoppt, sofern Sie keinen angrenzenden Zeitraum geplant haben.

Ebenso muss die Instance manuell gestartet werden, wenn Sie nur eine Stoppzeit angeben. Wenn Sie keine Uhrzeit angeben, verwendet diese Lösung die Regeln für Wochentage, Wochentage oder Monate, um Instances je nach Bedarf am `beginning/end` jeweiligen Tag zu starten und zu beenden.

Die `endtime` Werte `begintime` und für Ihren Zeitraum müssen in der Zeitzone liegen, die im Zeitplan angegeben ist. Wenn Sie im Zeitplan keine Zeitzone angeben, verwendet die Lösung die Zeitzone, die Sie beim Starten der Lösung angegeben haben.

Wenn Ihr Zeitplan mehrere Zeiträume enthält, empfehlen wir, dass Sie `endtime` in Ihren Zeiträumen immer `begintime` sowohl ein als auch angeben.

Wenn Sie eine Instance vor der angegebenen Startzeit starten, wird die Instance bis zum Ende der Laufzeit ausgeführt. Ein Benutzer könnte beispielsweise einen Zeitraum definieren, in dem eine Instance täglich um 9 Uhr gestartet und diese Instanz um 17 Uhr beendet wird.

Geplantes Starten und Stoppen von 9:00 bis 17:00 Uhr



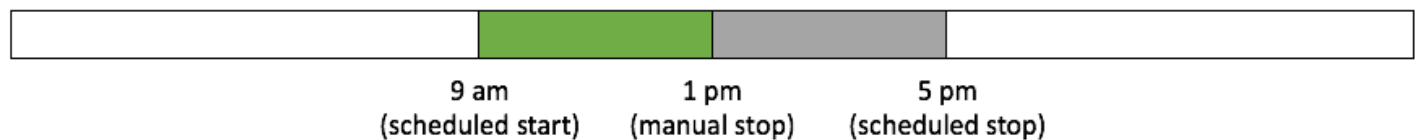
Wenn Sie diese Instanz um 5 Uhr manuell starten, stoppt die Lösung die Instanz um 17 Uhr. Wenn Sie das [Feld Retain Running](#) verwenden, stoppt die Lösung die Instance nicht um 17 Uhr.

Geplanter Stopp um 5 Uhr



Wenn Sie eine Instance vor der angegebenen Stoppzeit beenden, wird die Instance erst zu Beginn der nächsten Laufzeit ausgeführt. Fortsetzung des vorherigen Beispiels: Wenn der Benutzer die Instanz am Mittwoch um 13 Uhr stoppt, startet die Lösung die Instanz erst am Donnerstag um 9 Uhr.

Die Zeitleiste zeigt den geplanten Start um 9 Uhr, den manuellen Stopp um 13 Uhr und den geplanten Stopp um 17 Uhr.



Benachbarte Perioden

Die Lösung stoppt die Ausführung von Instances nicht, wenn der Zeitplan zwei aufeinanderfolgende Laufperioden enthält. Wenn Sie beispielsweise einen Zeitplan mit einem Zeitraum `endtime` von 23:59 Uhr und einen anderen Zeitraum mit einem Zeitraum `begintime` von Mitternacht am Folgetag haben, stoppt die Lösung die Ausführung von Instances nicht, wenn es `keineweekdays`, oder `months` Regeln gibt `monthdays`, die die Instances beenden.

Um einen Zeitplan zu implementieren, nach dem Instanzen von Montag bis Freitag von 9 Uhr bis 17 Uhr ausgeführt werden, benötigt die Lösung drei Zeiträume. In der ersten Phase werden die entsprechenden Instances am Montag von 9.00 Uhr bis 23.59 Uhr ausgeführt. In der zweiten Phase werden die Instances von Dienstag Mitternacht bis Donnerstag 23:59 Uhr ausgeführt. In der dritten Periode werden die Instances von Freitag Mitternacht bis Freitag 17:00 Uhr ausgeführt. Weitere Informationen finden Sie unter [Beispielpläne](#).

Wochentage

Das `weekdays` Feld definiert, an welchen Wochentagen eine Instanz ausgeführt wird. Sie können eine Liste von Tagen, einen Bereich von Tagen, das `n-te` Vorkommen dieses Tages in einem Monat oder das letzte Vorkommen dieses Tages in einem Monat angeben. Die Lösung unterstützt abgekürzte Tagesnamen (Mon) und Zahlen (0).

Tage des Monats

Das Feld `monthdays`, definiert, an welchen Tagen im Monat eine Instanz ausgeführt wird. Sie können eine Liste von Tagen, einen Bereich von Tagen, jeden n^{ten} Tag des Monats, den letzten Tag des Monats oder den Wochentag angeben, der einem bestimmten Datum am nächsten liegt.

Monate

Das `months` Feld definiert, in welchen Monaten eine Instanz ausgeführt wird. Sie können eine Liste von Monaten, einen Bereich von Monaten oder alle n^{ten} Monate angeben. Die Lösung unterstützt abgekürzte Monatsnamen (Jan) und Zahlen (1).

Definitionen von Perioden

Die Konfigurationstabelle von Instance Scheduler on AWS in Amazon DynamoDB enthält Periodendefinitionen. Eine Periodendefinition kann die folgenden Felder enthalten. Beachten Sie, dass einige Felder [nicht standardmäßige Cron-Zeichen](#) unterstützen.

Important

Sie müssen mindestens eines der folgenden Elemente angeben: Startzeit, Endzeit, Wochentage, Monate oder Monatstage.

Feld	Description
<code>begintime</code>	Die Uhrzeit im Format HH:MM, zu der die Instanz gestartet wird.
<code>description</code>	Eine optionale Beschreibung des Zeitraums.
<code>endtime</code>	Die Uhrzeit im Format HH:MM, zu der die Instanz beendet wird.
<code>months</code>	Geben Sie eine durch Kommas getrennte Liste von Monaten oder einen Bereich von Monaten mit Bindestrich ein, in denen die Instance ausgeführt werden soll. Geben Sie beispielsweise <code>jan, feb, mar</code> oder <code>ein, um</code>

Feld	Description
	<p>eine Instance 1, 2, 3 in diesen Monaten auszuführen. Oder Sie können jan-mar oder eingeben1-3.</p> <p>Sie können eine Instance auch so planen, dass sie jeden n^{ten} Monat oder jeden n^{ten} Monat in einem Bereich ausgeführt wird. Geben Sie beispielsweise Jan/3 oder ein, 1/3 um ab Januar alle drei Monate eine Instance auszuführen. Geben Jan-Jul/2 Sie ein, ob die Ausführung alle zwei Monate von Januar bis Juli erfolgen soll.</p>

Feld	Description
monthdays	<p>Geben Sie eine durch Kommas getrennte Liste von Tagen des Monats oder einen Bereich von Tagen mit Bindestrich ein, an denen die Instance ausgeführt werden soll. Geben Sie beispielsweise 1, 2, 3 oder ein, 1-3 um eine Instance an den ersten drei Tagen des Monats auszuführen. Sie können auch mehrere</p> <p style="text-align: right;"><small>Geben Sie beispielsweise ein1-3, 7-9 um eine Instanz vom 1. bis zum 3. und vom 7. bis zum 9. auszuführen.</small></p> <p>Bereiche eingeben.</p> <p>Sie können eine Instance auch so planen, dass sie an jedem n^{ten} Tag des Monats oder an jedem n^{ten} Tag des Monats in einem bestimmten Bereich ausgeführt wird. Geben Sie beispielsweise ein, dass ab 1/7 dem 1. jeden siebten Tag eine Instanz ausgeführt werden soll. <small>Geben Sie ein1-15/2, um vom 1. bis zum 15. Tag jeden zweiten Tag eine Instanz auszuführen</small></p> <p>Geben Sie einL, um eine Instanz am letzten Tag des Monats auszuführen. Geben Sie ein Datum und W ein, um eine Instance an dem Wochentag auszuführen, der dem angegebenen Datum am nächsten liegt. Geben Sie beispielsweise ein, 15W dass eine Instance an dem Wochentag ausgeführt werden soll, der dem 15. am nächsten liegt.</p>
name	<p>Der Name, der zur Identifizierung des Zeitraums verwendet wird. Dieser Name muss eindeutig sein und darf nur alphanumerische Zeichen, Bindestriche (-) und Unterstriche (_) enthalten.</p>

Feld	Description
weekdays	<p>Geben Sie eine durch Kommas getrennte Liste von Wochentagen oder einer Reihe von Wochentagen ein, an denen die Instance ausgeführt wird. Geben Sie beispielsweise 0, 1, 2 oder ein, um eine Instance 0-2 von Montag bis Mittwoch auszuführen. Sie können auch mehrere Bereiche eingeben. Geben Sie beispielsweise ein0-2, 4-6 um eine Instance jeden Tag außer Donnerstag auszuführen.</p> <p>Sie können eine Instance auch so planen, dass sie ^{an} jedem n-ten Tag eines Monats ausgeführt wird. Geben Sie beispielsweise Mon#1 oder ein, 0#1 um eine Instanz am ersten Montag des Monats auszuführen.</p> <p>Geben Sie einen Tag und L ein, um eine Instanz am letzten Tag dieses Wochentags im Monat auszuführen. Geben Sie beispielsweise friL oder ein, 4L um eine Instanz am letzten Freitag des Monats auszuführen.</p>

Wenn ein Zeitraum mehrere Bedingungen enthält, beachten Sie, dass alle Bedingungen erfüllt sein müssen, damit Instance Scheduler auf AWS die entsprechende Aktion anwenden kann. Beispiel: Ein Zeitraum, der ein weekdays Feld mit dem Wert von Mon#1 und ein Monatsfeld mit dem Wert von enthält, wendet Jan/3 die Aktion am ersten Montag des Quartals an.

Beispiele für Zeitpläne

Mit Instance Scheduler auf AWS können Sie Amazon Elastic Compute Cloud (Amazon EC2) und Amazon Relational Database Service (Amazon RDS) Instances automatisch starten und beenden. Der folgende Abschnitt enthält einige Beispielpläne, die an viele gängige Anwendungsfälle angepasst werden können.

Standardarbeitszeiten von 9 bis 5 Stunden

Dieser Zeitplan zeigt, wie Instances an Wochentagen von 9.00 Uhr bis 17.00 Uhr in London ausgeführt werden.

Zeiträume

In diesem Zeitraum werden Instances an Wochentagen (Mo-Fr) um 9 Uhr gestartet und Instances um 17 Uhr beendet.

Feld	Typ	Wert
begintime	String	09:00
endtime	String	16:59
name	String	weekdays-9-5
weekdays	StringSet	mon-fri

Plan

Der Name des Zeitplans gibt den Tag-Wert an, der auf Instances angewendet werden muss, und die Zeitzone, die verwendet werden soll.

Feld	Typ	Wert
name	String	london-working-hours
periods	StringSet	weekdays-9-5
timezone	String	Europe/London

Instanz-Tag

Um diesen Zeitplan auf Instances anzuwenden, müssen Sie den Instances das `Schedule=london-working-hours` Tag hinzufügen. Wenn Sie den Standard-Tagnamen im `Tagname`-Parameter von AWS CloudFormation Instance Scheduler ändern, ist Ihr Tag anders. Wenn Sie beispielsweise Sked

als Tag-Namen eingegeben haben, wird Ihr Tag wie folgt lauten `sked=london-working-hours`. Weitere Informationen finden Sie unter [Taggen Ihrer Ressourcen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Planer-CLI

Verwenden Sie die folgenden Befehle, um den obigen Zeitplan mit der [Instance Scheduler-CLI](#) zu konfigurieren:

```
scheduler-cli create-period --stack <stackname> --name weekdays-9-5 --weekdays mon-fri
--begintime 9:00 --endtime 16:59

scheduler-cli create-schedule --stack <stackname> --name london-working-hours --periods
weekdays-9-5 --timezone Europe/London

Europe/London
```

Benutzerdefinierte Ressource

Mit der folgenden CloudFormation Vorlage wird der obige Zeitplan mithilfe der [benutzerdefinierten Zeitplanressource](#) erstellt.

Um diese Vorlage bereitzustellen, müssen Sie den `ServiceInstanceScheduleServiceTokenARN` angeben, den Sie in der CloudFormation AWS-Konsole finden, indem Sie den [zuvor bereitgestellten Instance Scheduler Hub Stack](#) und dann Outputs auswählen.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  LondonWorkingWeek:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      NoStackPrefix: 'True'
      Name: london-working-hours
      Description: run instances from 9am to 5pm in London on weekdays
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN
      Timezone: Europe/London
```

Periods:

- Description: 9am to 5pm on weekdays
 - BeginTime: '09:00'
 - EndTime: '16:59'
 - WeekDays: mon-fri

Stoppen Sie Instances nach 17 Uhr

Instances können zu jeder Tageszeit nach Belieben gestartet werden. Dieser Zeitplan stellt sicher, dass ihnen täglich um 17 Uhr ET automatisch ein Stopp-Befehl gesendet wird.

Zeiträume

In diesem Zeitraum werden die Instances täglich um 17 Uhr beendet.

Feld	Typ	Wert
endtime	String	16:59
name	String	stop-at-5

Plan

Der Name des Zeitplans gibt den Tag-Wert an, der auf Instances angewendet werden muss, und die Zeitzone, die verwendet werden soll.

Feld		Value (Wert)
name	String	stop-at-5-new-york
periods	StringSet	stop-at-5
timezone	String	America/New York

Instanz-Tag

Um diesen Zeitplan auf Instances anzuwenden, müssen Sie den Instances das `Schedule=stop-at-5-new-york` Tag hinzufügen. Wenn Sie den Standard-Tag-Namen im `Tagname`-Parameter von AWS CloudFormation Instance Scheduler geändert haben, wird Ihr Tag anders sein. Wenn Sie

beispielsweise Sked als Tag-Namen eingegeben haben, wird Ihr Tag wie folgt lauten `sked=stop-at-5-new-york`. Weitere Informationen finden Sie unter [Taggen Ihrer Ressourcen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Planer-CLI

Verwenden Sie die folgenden Befehle, um den obigen Zeitplan mit der [Instance Scheduler-CLI](#) zu konfigurieren:

```
scheduler-cli create-period --stack <stackname> --name stop-at-5 --endtime 16:59

scheduler-cli create-schedule --stack <stackname> --name stop-at-5-new-york --periods
stop-at-5 --timezone America/New_York
```

Benutzerdefinierte Ressource

Mit der folgenden CloudFormation Vorlage wird der obige Zeitplan mithilfe der [benutzerdefinierten Zeitplanressource](#) erstellt.

Um diese Vorlage bereitzustellen, müssen Sie den `ServiceInstanceScheduleServiceTokenARN` angeben, den Sie in der CloudFormation AWS-Konsole finden, indem Sie auf den [zuvor bereitgestellten Instance Scheduler Hub Stack](#) klicken und Outputs auswählen.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  StopAfter5:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      NoStackPrefix: 'True'
      Name: stop-at-5-new-york
      Description: stop instances at 5pm ET every day
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN
      Timezone: America/New_York
      Periods:
        - Description: stop at 5pm
          EndTime: '16:59'
```

Stoppen Sie Instances über das Wochenende

Dieser Zeitplan zeigt, wie Instances von Montag 9 Uhr ET bis Freitag 17 Uhr ET ausgeführt werden. Da Montag und Freitag keine vollen Tage sind, umfasst dieser Zeitplan drei Zeiträume: Montag, Dienstag-Donnerstag und Freitag.

Zeiträume

Der erste Zeitraum beginnt mit Tagged Instances am Montag um 9.00 Uhr und endet um Mitternacht. Dieser Zeitraum umfasst die folgenden Felder und Werte.

Feld	Typ	Wert
begintime	String	09:00
endtime	String	23:59
name	String	mon-start-9am
weekdays	StringSet	mon

In der zweiten Phase werden markierte Instances den ganzen Tag von Dienstag bis Donnerstag ausgeführt. Dieser Zeitraum umfasst die folgenden Felder und Werte.

Feld		Value (Wert)
name	String	tue-thu-full-day
weekdays	StringSet	tue-thu

In der dritten Phase werden markierte Instances am Freitag um 17 Uhr beendet. Dieser Zeitraum umfasst die folgenden Felder und Werte.

Feld		Value (Wert)
begintime	String	00:00
endtime	String	16:59

Feld		Value (Wert)
name	String	fri-stop-5pm
weekdays	StringSet	fri

Plan

Der Zeitplan kombiniert die drei Zeiträume zu dem Zeitplan für markierte Instanzen. Der Zeitplan umfasst die folgenden Felder und Werte.

Feld		Value (Wert)
name	String	Mo-9-Fr-17 Uhr
periods	StringSet	Mo-Start-9 Uhr, Fr-Stop-17 Uhr tue-thu-full-day
timezone	String	Amerika/New_York

Instanz-Tag

Um diesen Zeitplan auf Instances anzuwenden, müssen Sie den Instances das `Schedule=mon-9am-fri-5pm` Tag hinzufügen. Beachten Sie, dass Ihr Tag anders sein wird, wenn Sie den Standard-Tagnamen im Tagname-Parameter von AWS CloudFormation Instance Scheduler geändert haben. Wenn Sie beispielsweise `Sked` als Tag-Namen eingegeben haben, wird Ihr Tag auch verwendet. `Sked=mon-9am-fri-5pm` Weitere Informationen finden Sie unter [Taggen Ihrer Ressourcen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Planer-CLI

Verwenden Sie die folgenden Befehle, um den obigen Zeitplan mit der [Instance Scheduler-CLI](#) zu konfigurieren:

```
scheduler-cli create-period --stack <stackname> --name
mon-start-9am --weekdays mon --begintime 9:00 --endtime 23:59
scheduler-cli create-period --stack <stackname> --name
```

```
tue-thu-full-day --weekdays tue-thu
scheduler-cli create-period --stack <stackname> --namefri-stop-5pm --weekdays fri --
begintime 0:00 --endtime 17:00

scheduler-cli create-schedule --stack <stackname> --name
mon-9am-fri-5pm --periods
mon-start-9am,tue-thu-full-day,fri-stop-5pm -timezone
America/New_York
```

Benutzerdefinierte Ressource

Mit der folgenden CloudFormation Vorlage wird der obige Zeitplan mithilfe der [benutzerdefinierten Zeitplanressource](#) erstellt.

Um diese Vorlage bereitzustellen, müssen Sie den ServiceInstanceScheduleServiceTokenARN angeben, den Sie in der CloudFormation AWS-Konsole finden, indem Sie den [zuvor bereitgestellten Instance Scheduler Hub Stack](#) und dann Outputs auswählen.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  StopOnWeekends:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      NoStackPrefix: 'True'
      Name: mon-9am-fri-5pm
      Description: start instances at 9am on monday and stop them at 5pm on friday
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN
      Timezone: America/New_York
      Periods:
        - Description: 9am monday start
          BeginTime: '09:00'
          EndTime: '23:59'
          WeekDays: mon
        - Description: all day tuesday-thursday
          WeekDays: tue-thu
        - Description: 5pm friday stop
          BeginTime: '00:00'
```

```
EndTime: '16:59'
WeekDays: fri
```

Ressourcen zur Lösung

Die folgenden Ressourcen werden als Teil des Instance Scheduler on AWS-Stacks erstellt.

Ressourcenname	Typ	Description
Wichtigste	AWS::Lambda::Function	Instance Scheduler auf der AWS Lambda Lambda-Funktion.
Scheduler Config Helper	Custom::ServiceSetup	Speichert globale Konfigurationseinstellungen in Amazon DynamoDB.
Scheduler: Berechtigung aufrufen	AWS::Lambda::Permission	Ermöglicht dem CloudWatch Amazon-Ereignis, die AWS-Lambda-Funktion des Instance Schedulers aufzurufen.
Scheduler-Protokolle	AWS::Logs::LogGroup	CloudWatch Protokollgruppe für Instance Scheduler.
Scheduler-Richtlinie	AWS::IAM::Policy	Richtlinie, die es dem Scheduler ermöglicht, Start- und Stoppaktionen durchzuführen, Amazon EC2 EC2-Instance-Attribute zu ändern, Tags festzulegen und auf Scheduler-Ressourcen zuzugreifen.
Scheduler-Regel	AWS::Events::Rule	EventBridge Amazon-Ereignisregel, die die Lambda-Funktion des Schedulers aufruft.

Ressourcenname	Typ	Description
Ereignisregel für Konfigurationsmetriken	<code>AWS::Events::Rule</code>	EventBridge Amazon-Ereignisregel, die regelmäßig die Funktion für anonymisierte Metriken zur Konfigurationsbeschreibung aufruft. Deaktiviert, wenn anonymisierte Metriken deaktiviert sind.
Tabelle „Status“	<code>AWS::DynamoDB::Table</code>	DynamoDB-Tabelle, die den letzten gewünschten Status von Instanzen speichert.
Tabelle Config“	<code>AWS::DynamoDB::Table</code>	DynamoDB-Tabelle, die globale Konfiguration, Zeitplan und Periodendaten speichert.
SNS-Thema zu Instance Scheduler	<code>AWS::SNS::Topic</code>	Sendet Warn- und Fehlermeldungen an abonnierte E-Mail-Adressen.

Planer-CLI

Mit dem Instance Scheduler auf der AWS-Befehlszeilenschnittstelle (CLI) können Sie Zeitpläne und Zeiträume konfigurieren und die Kosteneinsparungen für einen bestimmten Zeitplan abschätzen.

Voraussetzungen

Die CLI in dieser Lösung erfordert Python 3.8+ und die neueste Version von boto3.

Anmeldeinformationen

Um die Scheduler-CLI verwenden zu können, benötigen Sie Anmeldeinformationen für die AWS-CLI. Weitere Informationen finden Sie unter [Konfiguration und Einstellungen für Anmeldeinformationsdateien](#) im AWS-CLI-Benutzerhandbuch.

Ihre Anmeldeinformationen müssen über die folgenden Berechtigungen verfügen:

- `lambda:InvokeFunction`- Um die `InstanceSchedulerMain` Funktion im Scheduler-Stack aufzurufen und die Zeitplan- und Periodeninformationen in der Scheduler-Konfigurationsdatenbank von der Befehlszeile aus zu aktualisieren
- `cloudformation:DescribeStackResource`- Um die physische Ressourcen-ID der AWS-Lambda-Funktion aus dem Stack abzurufen, um die CLI-Anforderung zu bearbeiten

Von der Scheduler-CLI gestellte Anfragen und Antworten werden im `AdminCliRequestHandler-yyyymmdd` Protokollstream protokolliert.

Note

Wenn Sie ein Profil mit dem Argument `profile-name` angeben, muss das von Ihnen angegebene Profil über diese Berechtigungen verfügen. [Weitere Informationen zum Argument Profilname finden Sie unter Allgemeine Argumente.](#)

Installieren Sie die Scheduler-CLI

1. [Laden Sie](#) das Scheduler-CLI-Paket (`instance_scheduler_cli.zip`) herunter und platzieren Sie es in einem Verzeichnis auf Ihrem Computer.

Important

Die Installation schlägt fehl, wenn Sie die Dateien nicht in einem eigenen Verzeichnis platzieren und sie dann von diesem Verzeichnis aus installieren.

2. Entpacken Sie das ZIP-Archiv in ein eigenes Verzeichnis (`instance_scheduler_cli`).
3. Installieren Sie Scheduler-CLI aus demselben Verzeichnis, in dem Sie das entpackte CLI-Paket abgelegt haben, in Ihrer Umgebung:

Note

Scheduler-CLI benötigt Python 3.8 oder höher und die neuesten Versionen von `pip` und `boto3`. Wenn Sie nicht alle diese auf Ihrem lokalen Computer installiert haben, finden Sie in der [offiziellen Dokumentation von pip](#) Installationsanweisungen, bevor Sie versuchen, die Scheduler-CLI zu installieren.

```
pip install --no-index --find-links=instance_scheduler_cli instance_scheduler_cli
```

4. Überprüfen Sie, ob die Installation erfolgreich war mit:

```
scheduler-cli --help
```

Note

Falls gewünscht, ein [Sdist der CLI](#) und kann mit demselben Verfahren wie oben installiert werden.

Befehlsstruktur

Die Scheduler-CLI verwendet eine mehrteilige Struktur in der Befehlszeile. Der nächste Teil spezifiziert das Scheduler-CLI-Python-Skript. Die Scheduler-CLI verfügt über Befehle, die die Operationen angeben, die in Zeiträumen und Zeitplänen ausgeführt werden sollen. Die spezifischen Argumente für eine Operation können in der Befehlszeile in beliebiger Reihenfolge angegeben werden.

```
scheduler-cli <command> <arguments>
```

Allgemeine Argumente

Die Scheduler-CLI unterstützt die folgenden Argumente, die alle Befehle verwenden können:

Argument	Description
<code>--stack [replaceable]<stackname></code>	Der Name des Scheduler-Stacks. Wichtig: Dieses Argument ist für alle Befehle erforderlich.
<code>--region [replaceable]<regionname></code>	Der Name der Region, in der der Scheduler-Stack bereitgestellt wird.

Argument	Description
	Hinweis: Sie müssen dieses Argument verwenden, wenn die Standardkonfigurations- und Anmeldeinformationsdateien nicht in derselben Region wie der Lösungstapel installiert sind.
<code>--profile-name [replaceable] <profilename></code>	Der Name des Profils, das zur Ausführung von Befehlen verwendet werden soll. Wenn kein Profilname angegeben ist, wird das Standardprofil verwendet.
<code>--query</code>	Ein JMESPath Ausdruck, der die Befehlsausgabe steuert. Weitere Informationen zur Steuerung der Ausgabe finden Sie unter Steuern der Befehlsausgabe über die AWS-Befehlszeilenschnittstelle im AWS-CLI-Benutzerhandbuch.
<code>--help</code>	Zeigt gültige Befehle und Argumente für die Scheduler-CLI an. Bei Verwendung mit einem bestimmten Befehl werden gültige Unterbefehle und Argumente für diesen Befehl angezeigt.
<code>--version</code>	Zeigt die Versionsnummer der Scheduler-CLI an.

Verfügbare Befehle

- [Erstellungszeitraum](#)
- [Zeitplan erstellen](#)
- [Zeitraum löschen](#)
- [Zeitplan löschen](#)
- [Zeiträume beschreiben](#)
- [Zeitpläne beschreiben](#)

- [describe-schedule-usage](#)
- [Aktualisierungszeitraum](#)
- [Aktualisierungszeitplan](#)
- [help](#)

Zeitraum erstellen

Description

Erzeugt eine Periode. Eine Periode muss mindestens eines der folgenden Elemente enthalten: `begintime`, `endtimeweekdays`, `months`, oder `monthdays`.

Argumente

`--name`

- Der Name der Periode

Typ: Zeichenfolge

Erforderlich: Ja

`--description`

- Eine Beschreibung des Zeitraums

Typ: Zeichenfolge

Erforderlich: Nein

`--begintime`

- Der Zeitpunkt, zu dem die Laufzeit beginnt. Wenn `begintime` und nicht angegeben `endtime` sind, ist die Laufzeit von 00:00 bis 23:59.

Typ: Zeichenfolge

Einschränkungen: oder Format `H:MM HH:MM`

Erforderlich: Nein

--endtime

- Der Zeitpunkt, zu dem die Laufzeit endet. Wenn `begintime` und nicht angegeben `endtime` sind, ist die Laufzeit von 00:00 bis 23:59.

Typ: Zeichenfolge

Einschränkungen: oder Format H:MM HH:MM

Erforderlich: Nein

--weekdays

- Die Wochentage für den Zeitraum

Typ: Zeichenfolge

Einschränkungen: Durch Kommas getrennte Liste abgekürzter Tagesnamen (Mon) oder Zahlen (0). Verwenden Sie -, um einen Bereich anzugeben. ^{Verwenden Sie/, um jeden n-ten ^ Tag der Woche} anzugeben.

Erforderlich: Nein

--months

- Die Monate des Zeitraums

Typ: Zeichenfolge

Einschränkungen: Durch Kommas getrennte Liste abgekürzter Monatsnamen (Jan) oder Zahlen (1). Verwenden Sie -, um einen Bereich anzugeben. ^{Verwenden Sie/, um jeden n-ten ^ Monat} anzugeben.

Erforderlich: Nein

--monthdays

- Die Tage des Monats für den Zeitraum

Typ: Zeichenfolge

Einschränkungen: Durch Kommas getrennte Liste abgekürzter Monatsnamen (Jan) oder Zahlen (1). Verwenden Sie -, um einen Bereich anzugeben. ^{Verwenden Sie/, um jeden n-ten ^ Tag des Monats} anzugeben.

Erforderlich: Nein

Beispiel

```
$ scheduler-cli create-period --name "weekdays" --begintime 09:00 --endtime 18:00 --
weekdays mon-fri --stack Scheduler
{
  "Period": {
    "Name": "weekdays",
    "Endtime": "18:00",
    "Type": "period",
    "Begintime": "09:00",
    "Weekdays": [
      "mon-fri"
    ]
  }
}
```

Zeitplan erstellen

Description

Erstellt einen Zeitplan.

Argumente

--name

- Der Name des Zeitplans

Typ: Zeichenfolge

Erforderlich: Ja

--description

- Eine Beschreibung des Zeitplans

Typ: Zeichenfolge

Erforderlich: Nein

--enforced

- Erzwingt den geplanten Status für die Instanz

Erforderlich: Nein

--use-metrics

- Sammeln Sie CloudWatch Amazon-Metriken

Erforderlich: Nein

--periods

- Eine Liste der Laufzeiten für den Zeitplan. Wenn mehrere Perioden angegeben sind, startet die Lösung eine Instanz, wenn einer der Perioden als ausgewertet wird. `true`

Typ: Zeichenfolge

Einschränkungen: Durch Kommas getrennte Liste von Perioden. Wird verwendet `<period-name>@[replaceable]<instance type>`, um einen Instance-Typ für einen Zeitraum anzugeben. Beispiel, `weekdays@2.large`.

Erforderlich: Ja

--retain-running

- Verhindert, dass eine Instanz am Ende einer Laufzeit von der Lösung gestoppt wird, wenn die Instanz vor Beginn des Zeitraums manuell gestartet wurde.

Erforderlich: Nein

--ssm-maintenance-window

- Fügt einem Amazon EC2 EC2-Instance-Zeitplan ein AWS Systems Manager Manager-Wartungsfenster als Laufzeit hinzu.

Typ: Zeichenfolge

Erforderlich: Nein

--do-not-stop-new-instances

- Stoppen Sie eine Instance nicht, wenn sie zum ersten Mal markiert wird, wenn sie außerhalb einer Laufzeit läuft

Erforderlich: Nein

--timezone

- Die Zeitzone, die der Zeitplan verwenden wird

Typ: Zeichenfolgen-Array

Erforderlich: Nein (Wenn dieses Argument nicht verwendet wird, wird die Standardzeitzone aus dem Hauptlösungsstapel verwendet.)

`--use-maintenance-window`

- Fügt einem Amazon RDS-Instance-Zeitplan ein Amazon RDS-Wartungsfenster als Laufzeit oder ein AWS Systems Manager Manager-Wartungsfenster als Laufzeit zu einem Amazon EC2 EC2-Instance-Zeitplan hinzu

Typ: wahr/falsch

Erforderlich: Nein (Standardeinstellung ist wahr)

Beispiel

```
$ scheduler-cli create-schedule --name LondonOfficeHours --periods weekdays,weekends --
timezone Europe/London --stack Scheduler
{
  "Schedule": {
    "Enforced": false,
    "Name": "LondonOfficeHours",
    "StopNewInstances": true,
    "Periods": [
      "weekends",
      "weekdays"
    ],
    "Timezone": "Europe/London",
    "Type": "schedule"
  }
}
```

Zeitraum löschen

`--name`

- Der Name des entsprechenden Zeitraums

Typ: Zeichenfolge

Erforderlich: Ja

⚠ Important

Wenn der Zeitraum in vorhandenen Zeitplänen verwendet wird, müssen Sie ihn aus diesen Zeitplänen entfernen, bevor Sie ihn löschen.

Beispiel

```
$ scheduler-cli delete-period --name weekdays --stack Scheduler
{
  "Period": "weekdays"
}
```

Zeitplan löschen

Description

Löscht einen vorhandenen Zeitplan

Argumente

`--name`

- Der Name des entsprechenden Zeitplans

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

```
$ scheduler-cli delete-schedule --name LondonOfficeHours --stack Scheduler
{
  "Schedule": "LondonOfficeHours"
}
```

Zeiträume beschreiben

Description

Listet die konfigurierten Zeiträume für den Instance Scheduler-Stack auf

Argumente

--name

- Der Name eines bestimmten Zeitraums, den Sie beschreiben möchten

Typ: Zeichenfolge

Erforderlich: Nein

Beispiel

```
$ scheduler-cli describe-periods --stack Scheduler
{
  "Periods": [
    {
      "Name": "first-monday-in-quarter",
      "Months": [
        "jan/3"
      ],
      "Type": "period",
      "Weekdays": [
        "mon#1"
      ],
      "Description": "Every first Monday of each quarter"
    },
    {
      "Description": "Office hours",
      "Weekdays": [
        "mon-fri"
      ],
      "Begintime": "09:00",
      "Endtime": "17:00",
      "Type": "period",
      "Name": "office-hours"
    },
    {
      "Name": "weekdays",
      "Endtime": "18:00",
      "Type": "period",
      "Weekdays": [
        "mon-fri"
      ]
    }
  ]
}
```

```
    ],
    "Begintime": "09:00"
  },
  {
    "Name": "weekends",
    "Type": "period",
    "Weekdays": [
      "sat-sun"
    ],
    "Description": "Days in weekend"
  }
]
```

Zeitpläne beschreiben

Description

Listet die konfigurierten Zeitpläne für den Instance Scheduler-Stack auf.

Argumente

`--name`

- Der Name eines bestimmten Zeitplans, den Sie beschreiben möchten

Typ: Zeichenfolge

Erforderlich: Nein

Beispiel

```
$ scheduler-cli describe-schedules --stack Scheduler

{
  "Schedules": [
    {
      "OverrideStatus": "running",
      "Type": "schedule",
      "Name": "Running",
      "UseMetrics": false
    },
  ],
}
```

```
{
  "Timezone": "UTC",
  "Type": "schedule",
  "Periods": [
    "working-days@t2.micro",
    "weekends@t2.nano"
  ],
  "Name": "scale-up-down"
},
{
  "Timezone": "US/Pacific",
  "Type": "schedule",
  "Periods": [
    "office-hours"
  ],
  "Name": "seattle-office-hours"
},
{
  "OverrideStatus": "stopped",
  "Type": "schedule",
  "Name": "stopped",
  "UseMetrics": true
}
]
```

describe-schedule-usage

Description

Listet alle Perioden auf, die innerhalb eines Zeitplans laufen, und berechnet die Abrechnungsstunden für Instanzen. Verwenden Sie diesen Befehl, um einen Zeitplan zur Berechnung potenzieller Einsparungen und Laufzeiten nach der Erstellung oder Aktualisierung eines Zeitplans zu simulieren.

Argumente

--name

- Der Name des entsprechenden Zeitplans

Typ: Zeichenfolge

Erforderlich: Ja

--startdate

- Das Startdatum des für die Berechnung verwendeten Zeitraums. Das Standarddatum ist das aktuelle Datum.

Typ: Zeichenfolge

Erforderlich: Nein

--enddate

- Das Enddatum des für die Berechnung verwendeten Zeitraums. Das Standarddatum ist das aktuelle Datum.

Typ: Zeichenfolge

Erforderlich: Nein

Beispiel

```
$ scheduler-cli describe-schedule-usage --stack InstanceScheduler --name seattle-
office-hours
{
  "Usage": {
    "2017-12-04": {
      "BillingHours": 8,
      "RunningPeriods": {
        "Office-hours": {
          "Begin": "12/04/17 09:00:00",
          "End": "12/04/17 17:00:00",
          "BillingHours": 8,
          "BillingSeconds": 28800
        }
      },
      "BillingSeconds": 28800
    }
  },
  "Schedule": "seattle-office-hours"
```

Aktualisierungszeitraum

Description

Aktualisiert einen bestehenden Zeitraum

Argumente

Der `update-period` Befehl unterstützt dieselben Argumente wie der `create-period` Befehl. Weitere Informationen zu den Argumenten finden Sie im [Befehl create period](#).

Important

Wenn Sie kein Argument angeben, wird dieses Argument aus dem Punkt entfernt.

Zeitplan für die Aktualisierung

Description

Aktualisiert einen vorhandenen Zeitplan

Argumente

Der `update-schedule` Befehl unterstützt dieselben Argumente wie der `create-schedule` Befehl. Weitere Informationen zu den Argumenten finden Sie im [Befehl create schedule](#).

Important

Wenn Sie kein Argument angeben, wird dieses Argument aus dem Zeitplan entfernt.

help

Description

Zeigt eine Liste gültiger Befehle und Argumente für die Scheduler-CLI an.

Beispiel

```
$ scheduler-cli --help
```

```
usage: scheduler-cli [-h] [--version]
                    {create-period,create-schedule,delete-period,delete-
schedule,describe-periods,describe-schedule-usage,describe-schedules,update-
period,update-schedule}
                    ...

optional arguments:
  -h, --help            show this help message and exit
  --version              show program's version number and exit

subcommands:
  Valid subcommands

  {create-period,create-schedule,delete-period,delete-schedule,describe-
periods,describe-schedule-usage,describe-schedules,update-period,update-schedule}

  create-period          Creates a period
  create-schedule        Creates a schedule
  delete-period          Deletes a period
  delete-schedule        Deletes a schedule
  describe-periods       Describes configured periods
  describe-schedule-usage
                        Calculates periods and billing hours in which
                        instances are running
  describe-schedules     Described configured schedules
  update-period          Updates a period
  update-schedule        Updates a schedule
```

Bei Verwendung mit einem bestimmten Befehl zeigt das `--help` Argument gültige Unterbefehle und Argumente für diesen Befehl an.

Beispiel für einen bestimmten Befehl

```
$ scheduler-cli describe-schedules --help
usage: scheduler-cli describe-schedules [-h] [--name NAME] [--query QUERY]
                                         [--region REGION] --stack STACK

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Name of the schedule
  --query QUERY         JMESPath query to transform or filter the result
  --region REGION       Region in which the Instance Scheduler stack is
                        deployed
```

```
--stack STACK, -s STACK  
Name of the Instance Scheduler stack
```

Aktualisieren Sie die globalen Konfigurationseinstellungen

Als Sie die Hub-Vorlage von Instance Scheduler zum ersten Mal in AWS bereitgestellt haben CloudFormation, wurden eine Reihe globaler Konfigurationseinstellungen als Parametereingaben ausgewählt. Diese globalen Konfigurationsparameter können jederzeit in der CloudFormation Konsole aktualisiert werden.

Um die globale Konfiguration von Instance Scheduler zu aktualisieren, melden Sie sich bei der Bereitstellung an, die Ihren Hub account/region enthält, und rufen Sie die CloudFormation AWS-Konsole auf. Suchen Sie den Instance Scheduler Hub-Stack und wählen Sie Aktualisieren → Bestehende Vorlage verwenden aus. Aktualisieren Sie alle globalen Konfigurationsparameter, die Sie ändern möchten, und wählen Sie dann Weiter → Weiter → Senden aus, um die entsprechenden Lösungsressourcen zu CloudFormation aktualisieren.

Verwalten Sie Zeitpläne mithilfe von Infrastructure as Code (IaC)

Important

Stellen Sie Zeitpläne mithilfe einer separaten Vorlage bereit, nachdem die Hub-Stack-Bereitstellung abgeschlossen ist.

Instance Scheduler auf AWS bietet eine benutzerdefinierte Ressource (`ServiceInstanceSchedule`), mit der Sie Zeitpläne über AWS CloudFormation konfigurieren und verwalten können. Die benutzerdefinierte Ressource verwendet PascalCase Schlüssel für dieselben Daten wie die Instance Scheduler-Konfigurationstabelle in Amazon DynamoDB (Beispiele finden Sie in der Vorlage unten). [Weitere Informationen zu den Feldern für Zeitpläne finden Sie unter Zeitplandefinitionen.](#) Weitere Informationen zu den Feldern für Perioden finden Sie unter [Periodendefinitionen.](#)

Wenn Sie die benutzerdefinierte Ressource verwenden, um einen Zeitplan zu erstellen, entspricht der Name dieses Zeitplans standardmäßig dem logischen Ressourcennamen der benutzerdefinierten Ressource. Um einen anderen Namen anzugeben, verwenden Sie die Name-Eigenschaft der benutzerdefinierten Ressource. Die Lösung fügt dem Namen des Zeitplans standardmäßig auch

den Stacknamen als Präfix hinzu. Wenn Sie den Stacknamen nicht als Präfix hinzufügen möchten, verwenden Sie die `NoStackPrefix` Eigenschaft.

Wenn Sie den Namen und die `NoStackPrefix` Eigenschaften verwenden, stellen Sie sicher, dass Sie eindeutige Zeitplannamen wählen. Wenn bereits ein Zeitplan mit demselben Namen existiert, wird die Ressource nicht erstellt oder aktualisiert.

Um mit der Verwaltung von Zeitplänen mit IaC zu beginnen, kopieren Sie die folgende Beispielvorlage und fügen Sie sie ein und passen Sie so viele oder so wenige Zeitpläne an, wie Sie möchten. Speichern Sie die Datei als `.template`-Datei (z. B.: `my-schedules.template`) und stellen Sie dann Ihre neue Vorlage mit AWS CloudFormation bereit. Beispiele für fertige Zeitplanvorlagen finden Sie unter [Beispielpläne](#).

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  SampleSchedule1:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN #do not edit this line
      NoStackPrefix: 'False'
      Name: my-renamed-sample-schedule
      Description: a full sample template for creating cfn schedules showing all
possible values
      Timezone: America/New_York
      Enforced: 'True'
      Hibernate: 'True'
      RetainRunning: 'True'
      StopNewInstances: 'True'
      UseMaintenanceWindow: 'True'
      SsmMaintenanceWindow: 'my_window_name'
    Periods:
      - Description: run from 9-5 on the first 3 days of March
        BeginTime: '9:00'
        EndTime: '17:00'
        InstanceType: 't2.micro'
        MonthDays: '1-3'
```

```
Months: '3'  
- Description: run from 2pm-5pm on the weekends  
  BeginTime: '14:00'  
  EndTime: '17:00'  
  InstanceType: 't2.micro'  
  WeekDays: 'Sat-Sun'
```

SampleSchedule2:

```
Type: 'Custom::ServiceInstanceSchedule'
```

Properties:

```
ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN #do not edit this line
```

```
NoStackPrefix: 'True'
```

```
Description: a sample template for creating simple cfn schedules
```


```
Timezone: Europe/Amsterdam
```

Periods:

```
- Description: stop at 5pm every day
```

```
  EndTime: '17:00'
```

Bei der Bereitstellung der Vorlage müssen Sie ServiceToken ARN für Ihre Bereitstellung von Instance Scheduler auf AWS angeben. Sie finden diesen ARN darin, CloudFormation indem Sie zu Ihrem bereitgestellten Instance Scheduler-Stack navigieren, Outputs auswählen und nach ServiceInstanceScheduleServiceToken.

 Important

Verwenden Sie nicht die DynamoDB-Konsole oder die Scheduler-CLI, um Zeitpläne und Zeiträume zu löschen oder zu ändern, die mit der benutzerdefinierten Ressource konfiguriert wurden. Wenn Sie dies tun, entsteht ein Konflikt zwischen den gespeicherten Parametern im Stack und den Werten in der Tabelle. Verwenden Sie auch keine Zeiträume, die mit der benutzerdefinierten Ressource konfiguriert wurden, in Zeitplänen, die mit der DynamoDB-Konsole oder der Scheduler-CLI erstellt wurden.

Bevor Sie den Instance Scheduler-Hauptstapel löschen, müssen Sie alle zusätzlichen Stacks löschen, die Zeitpläne und Perioden enthalten, die mit der benutzerdefinierten Ressource erstellt wurden, da die benutzerdefinierten Ressourcenstapel Abhängigkeiten von der DynamoDB-Tabelle des Haupt-Stacks enthalten.

In der DynamoDB-Konfigurationstabelle können Zeitpläne und Zeiträume, die mit der benutzerdefinierten Ressource konfiguriert wurden, anhand des configured_in_stack-Attributs

identifiziert werden. Das Attribut enthält den Amazon-Ressourcennamen des Stacks, der zur Erstellung des Artikels verwendet wurde.

Behandlung von EC2-Fehlern mit unzureichender Kapazität

Wenn Instance Scheduler eine Instance aufgrund unzureichender Kapazität nicht starten kann, gibt er standardmäßig ein Startfehler aus (siehe [EventBridge Ereignisse](#)) und versucht es im nächsten Planungsintervall erneut. Alternativ kann Instance Scheduler so konfiguriert werden, dass die Größe Ihrer Instance auf alternative Instance-Typen angepasst wird, bevor der Startvorgang erneut versucht wird. Diese Funktion trägt dazu bei, die Verfügbarkeit von Instanzen in Umgebungen mit eingeschränkter Kapazität zu verbessern.

Konfiguration

Um alternative Instance-Typen für eine EC2-Instance zu aktivieren, fügen Sie der Instance das `IS-PreferredInstanceTypes` Tag mit einer kommasetrennten Liste von Instance-Typen in der Reihenfolge ihrer Präferenz hinzu (am meisten bevorzugt zuerst):

```
IS-PreferredInstanceTypes: t3.medium,t3.large,m5.large
```

Funktionsweise

Die Liste der alternativen Instance-Typen wird in der Reihenfolge ihrer Präferenz angezeigt, wobei der erste Typ am meisten bevorzugt wird. Wenn Instance Scheduler versucht, eine EC2-Instance zu starten:

1. Wenn die Instance derzeit nicht die bevorzugte Größe hat, wird versucht, sie vor dem Start auf die am meisten bevorzugte Größe zu skalieren
2. Wenn der Startvorgang erfolgreich ist, werden keine weiteren Alternativen versucht
3. Wenn der Startvorgang aufgrund unzureichender Kapazität fehlschlägt:
 - a. Versucht, die Größe auf den nächsten alternativen Instanztyp in der Liste zu ändern
 - b. Wiederholt den Startvorgang
 - c. Falls es immer noch nicht erfolgreich ist, versucht es mit dem nächsten alternativen Typ
 - d. Setzt fort, bis der Vorgang erfolgreich ist oder alle Alternativen ausgeschöpft sind

Anforderungen und Einschränkungen

Instanzkompatibilität: Alternative Instance-Typen müssen mit der aktuellen Konfiguration der Instance kompatibel sein (AMI, Subnetz, Sicherheitsgruppen usw.). Weitere Informationen finden Sie unter [Ändern des Instance-Typs](#) im Amazon EC2 EC2-Benutzerhandbuch.

Tag-Format: Der `IS-PreferredInstanceTypes` Tag-Wert muss eine durch Kommas getrennte Liste gültiger EC2-Instance-Typen sein.

Beispiel

Für eine Instanz, die ursprünglich als konfiguriert wurde `t3.small`, könnten Sie Folgendes konfigurieren:

```
Schedule: office-hours  
IS-PreferredInstanceTypes: t3.small,t3.medium,t3.large,m5.large
```

Wenn die `t3.small` Instanz aufgrund von Kapazitätsproblemen nicht gestartet werden kann, versucht Instance Scheduler, die Größe zu ändern und die Instanz dann `t3.large` zu starten `t3.medium`, `m5.large` bis der Vorgang erfolgreich ist oder alle Optionen ausgeschöpft sind.

EC2 Auto Scaling Scaling-Gruppenplanung

Instance Scheduler auf AWS unterstützt die Planung von EC2 Auto Scaling Scaling-Gruppen (ASGs) mithilfe von Scheduled Scaling Actions. Dies unterscheidet sich von der Implementierung von EC2/RDS-Planung und wird in diesem Abschnitt näher erläutert

Weitere Informationen zu [geplanten Skalierungsaktionen finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling](#).

Überblick über die ASG-Planung

ASGs kann geplant werden, indem ein Zeitplan-Tag angewendet wird, wie unter [Instanzen für die Planung taggen](#) beschrieben

Definition von für Running/Stopped ASGs

Wenn eine Auto Scaling Scaling-Gruppe konfiguriert ist, gibt ein Benutzer eine Mindest-, gewünschte und maximale Kapazität für diese ASG an. Instance Scheduler bezeichnet diese Werte als die `min-desired-max` oder MDM einer ASG.

Der Betriebsstatus einer ASG wird mithilfe des Control-Tags definiert. `IS-MinDesiredMax`. Dieses Tag sollte die gewünschten MDM-Werte im Format `min,desired,max` (zum Beispiel: `1,3,5`).

Wenn das `IS-MinDesiredMax` Tag nicht angegeben wird, wenn die ASG zum ersten Mal für die Planung markiert wird, wird es automatisch anhand der aktuellen Größe der ASG zum Zeitpunkt der Markierung generiert.

Für alle ASGs der Status „Gestoppt“ als MDM-Wert von `0-0-0` definiert.

ASG-Verhalten Start/Stop

Wenn Instance Scheduler eine ASG startet oder stoppt, ändert er die Kapazitätseinstellungen der ASG:

Starten einer ASG: Legt die minimale, gewünschte und maximale Kapazität auf die im `IS-MinDesiredMax` Tag definierten Werte fest (oder auf die automatisch generierten Werte, als die ASG zum ersten Mal markiert wurde).

Stoppen einer ASG: Setzt die minimale, gewünschte und maximale Kapazität auf `0-0-0`, wodurch alle Instanzen in der ASG beendet werden.

Einschränkungen

Die ASG-Planung erfolgt durch die Konvertierung von Instance Scheduler auf AWS-Zeitplänen in geplante Skalierungsregeln, die mit dem ASG-Service kompatibel sind. Diese Übersetzung eignet sich am besten für einfache Zeitpläne mit einem Zeitraum, die keine komplexen Cron-Ausdrücke verwenden.

Die folgenden Zeitplanfunktionen werden für die ASG-Planung nicht unterstützt:

- Erweiterte Zeitplankennzeichnungen wie „erzwungen“ und „Dauerbetrieb“.
- Ausdrücke für den N-ten Wochentag, den nächsten Wochentag und den letzten Wochentag in Perioden.
- Zeitpläne mit mehreren Perioden mit unmittelbar benachbarten oder sich überschneidenden Perioden.
 - Bei der Konfiguration von geplanten Skalierungsaktionen für Zeitpläne mit mehreren Perioden übersetzt Instance Scheduler auf AWS die `beginning/end` Zeiträume direkt in `start/stop` Aktionen für die ASG, auch wenn ein anderer überlappender oder angrenzender Zeitraum normalerweise dazu führen würde, dass diese Aktion übersprungen wird.

Überwachen Sie die Lösung

Protokollierung und Benachrichtigungen

Instance Scheduler verwendet eine strukturierte Protokollierung, die für CloudWatch Logs Insights-Abfragen optimiert ist. Diese Lösung protokolliert Verarbeitungsinformationen für jede mit Tags versehene Instanz, die Ergebnisse der Periodenbewertung für die Instance, den gewünschten Status der Instanz während dieses Zeitraums, die angewandte Aktion und Debugging-Meldungen.

Protokolle werden in zwei Protokollgruppen in Amazon CloudWatch Logs geschrieben:

`{stackName}-{namespace}-administrative-logs`

Protokolle für die Registrierung und Deregistrierung von Ressourcen, benutzerdefinierte Ressourcenoperationen, CLI-Anfragen und andere administrative Aktivitäten.

`{stackName}-{namespace}-scheduling-logs`

Protokolle für Planungsvorgänge, einschließlich Orchestrierung und Ausführung von Anforderungshandlern.

Warn- und Fehlerprotokolle werden auch an ein von der Lösung erstelltes Amazon SNS SNS-Thema weitergeleitet, das so konfiguriert werden kann, dass Nachrichten an eine abonnierte E-Mail-Adresse gesendet werden. Den Namen des Amazon SNS SNS-Themas finden Sie auf der Registerkarte Outputs des Lösungstapels.

Informations-Tags

Wenn informatives Tagging aktiviert ist (Standardeinstellung), schreibt Instance Scheduler Tags direkt in verwaltete Ressourcen, um at-a-glance Einblick in die Planungsaktivitäten der Lösung zu erhalten. Diese Tags werden mithilfe der AWS Resource Groups Tagging API angewendet und jedes Mal aktualisiert, wenn der Scheduler eine Ressource verarbeitet.

Sie können diese Funktion mithilfe des Parameters Enable information Tagging im Hub-Stack aktivieren oder deaktivieren. Weitere Informationen finden Sie unter [Globale Konfigurationseinstellungen aktualisieren](#).

Tag-Schlüssel zu Informationszwecken

Die folgenden Tags werden in verwaltete Ressourcen geschrieben:

Tag-Schlüssel	Description
IS-ManagedBy	Der ARN des Instance Scheduler-Hub-Stacks, der diese Ressource verwaltet. Wird angewendet, wenn eine Ressource zum ersten Mal für die Planung registriert wird, und bei jeder nachfolgenden Planungsaktion.
IS-LastAction	Die letzte Planungsaktion, die für die Ressource ausgeführt wurde, zusammen mit einem UTC-Zeitstempel. Beispiel: <code>Started 2025-06-15 09:00:00 UTC</code> oder <code>Stopped 2025-06-15 17:00:00 UTC</code> . Dieses Tag wird nur aktualisiert, wenn der Scheduler eine Ressource aktiv startet oder stoppt (nicht, wenn er eine Ressource auswertet und feststellt, dass keine Aktion erforderlich ist).
IS-Error	Wenn der Scheduler bei der Verarbeitung einer Ressource auf einen Fehler stößt, enthält dieses Tag den Fehlercode und einen UTC-Zeitstempel. Beispiel: <code>StartFailed 2025-06-15 09:00:05 UTC</code> . Dieses Tag wird bei der nächsten erfolgreichen Planungsaktion automatisch gelöscht.
IS-ErrorMessage	Eine menschenlesbare Beschreibung des Fehlers. Dieses Tag ist nur vorhanden, wenn <code>IS-Error</code> es auch vorhanden ist, und wird daneben gelöscht.

Fehlercodes

Die folgenden Fehlercodes können im `IS-Error` Tag erscheinen:

Fehlercode	Description
UnknownSchedule	Der im Zeitplan-Tag der Ressource angegebene Zeitplaname stimmt mit keinem in der Konfigurationstabelle definierten Zeitplan überein.
UnsupportedResource	Der Ressourcentyp wird für die Planung nicht unterstützt (z. B. eine Read Replica einer anderen RDS-Instance).

Fehlercode	Description
IncompatibleSchedule	Der der Ressource zugewiesene Zeitplan ist nicht mit dem Ressourcentyp kompatibel (z. B. ein ASG-Zeitplan, der nicht unterstützte Cron-Ausdrücke verwendet).
StartFailed	Der Scheduler hat versucht, die Ressource zu starten, aber der Vorgang ist fehlgeschlagen.
StopFailed	Der Scheduler hat versucht, die Ressource zu stoppen, aber der Vorgang ist fehlgeschlagen.
ConfigurationFailed	Der Scheduler hat versucht, geplante Skalierungsregeln für eine Auto Scaling Scaling-Gruppe zu konfigurieren, aber der Vorgang schlug fehl.
UnknownError	Bei der Verarbeitung der Ressource ist ein unerwarteter Fehler aufgetreten.

Verhalten von Tags

- Wenn eine Ressource zum ersten Mal für die Planung registriert wird, wird das IS-ManagedBy Tag sofort angewendet.
- Wenn die Registrierung einer Ressource aufgehoben wird (das Zeitplan-Tag wird entfernt), werden alle Informations-Tags (IS-ManagedBy, IS-LastActionIS-Error,IS-ErrorMessage) aus der Ressource entfernt.
- Fehler-Tags werden nicht in jedem Planungsintervall neu geschrieben, wenn derselbe Fehler weiterhin besteht und das vorhandene Tag immer noch auf der Ressource vorhanden ist. Sie werden nur aktualisiert, wenn sich der Fehlercode ändert.
- Alle Tag-Werte werden auf 256 Zeichen gekürzt, um die AWS-Tagging-Beschränkungen einzuhalten.

Überlegungen zur Tag-Governance

Important

Instance Scheduler erstellt und aktualisiert die oben aufgeführten Tags auf verwalteten Ressourcen im Rahmen des normalen Betriebs. Wenn Ihre Organisation die Tag-Governance durch AWS Config-Regeln, Tag-Richtlinien, Service-Kontrollrichtlinien oder automatisierte Problembehebungen durchsetzt, stellen Sie sicher, dass Ihre Change-Management-Kontrollen so konfiguriert sind, dass sie die folgenden Tag-Schlüssel zulassen:

- IS-ManagedBy
- IS-LastAction
- IS-Error
- IS-ErrorMessage
- IS-PreferredInstanceTypes(wenn Sie alternative Instance-Typen verwenden)
- IS-MinDesiredMax(wenn Auto Scaling-Gruppen geplant werden)

Wenn Sie diese Tags nicht in Ihren Governance-Richtlinien berücksichtigen können, deaktivieren Sie das Tagging zu Informationszwecken, indem Sie den Parameter Informations-Tagging aktivieren auf dem Hub-Stack No auf setzen. Beachten Sie, dass dadurch auch das IS-ManagedBy Tag deaktiviert wird, das zur Bestätigung der Ressourcenregistrierung verwendet wird.

Kontroll-Tags

Zusätzlich zu den Informations-Tags verwendet Instance Scheduler die folgenden Kontroll-Tags für bestimmte Funktionen:

Tag-Schlüssel	Description
IS-PreferredInstanceTypes	Eine durch Kommas getrennte Liste alternativer EC2-Instance-Typen, die ausprobiert werden sollen, wenn das Starten einer Instance aufgrund unzureichender Kapazität fehlschlägt. Weitere Informationen finden Sie unter Behandlung von EC2-Fehlern mit unzureichender Kapazität.

Tag-Schlüssel	Description
IS-MinDesiredMax	Die minimalen, gewünschten und maximalen Kapazitätswerte für eine Auto Scaling Scaling-Gruppe im Format <code>min,desired,max</code> . Weitere Informationen finden Sie unter EC2 Auto Scaling Group Scheduling .

Tag-Kapazität

Important

AWS-Ressourcen haben in der Regel ein Limit von 50 Tags pro Ressource. Instance Scheduler kann bis zu 6 Tags für eine Ressource verwenden (4 Informations-Tags plus bis zu 2 Kontroll-Tags). Stellen Sie sicher, dass Ihre Ressourcen über ausreichend Tag-Kapazität verfügen, um Instance Scheduler-Tags zusätzlich zu Ihrer bestehenden Tagging-Strategie aufzunehmen.

Wenn eine Ressource das Limit von 50 Tags erreicht oder fast erreicht, schlägt das Schreiben von Informationstags möglicherweise fehl. Der Scheduler protokolliert diese Fehler, setzt aber die Planungsvorgänge fort. Überprüfen Sie die CloudWatch Protokolle, wenn Sie Probleme mit der Kennzeichnung vermuten.

CloudWatch Protokolliert Insights-Abfragen

Das strukturierte Protokollierungsformat von Instance Scheduler ermöglicht effizientes Abfragen mithilfe von CloudWatch Logs Insights. Sie können Logs Insights verwenden, um Protokolldaten zu suchen, zu analysieren und zu visualisieren, um Betriebsprobleme zu beheben und die Planungsaktivitäten zu überwachen.

Instance Scheduler bietet vorformatierte Protokollabfragen, auf die Sie über den Abschnitt Gespeicherte Abfragen in der CloudWatch Logs-Konsole zugreifen können:

SchedulingHistory

Aktionen zur Abfrageplanung, die an Ressourcen ausgeführt werden, einschließlich Start- und Stoppvorgängen.

RegistrationEvents

Fragen Sie Ereignisse zur Registrierung und Abmeldung von Ressourcen ab.

Errors

Fragen Sie Fehlerprotokolle ab, um Probleme mit der Lösung zu beheben.

Weitere Informationen zu CloudWatch Logs Insights finden Sie unter [Analysieren von Protokolldaten mit CloudWatch Logs Insights](#) im Amazon CloudWatch Logs-Benutzerhandbuch.

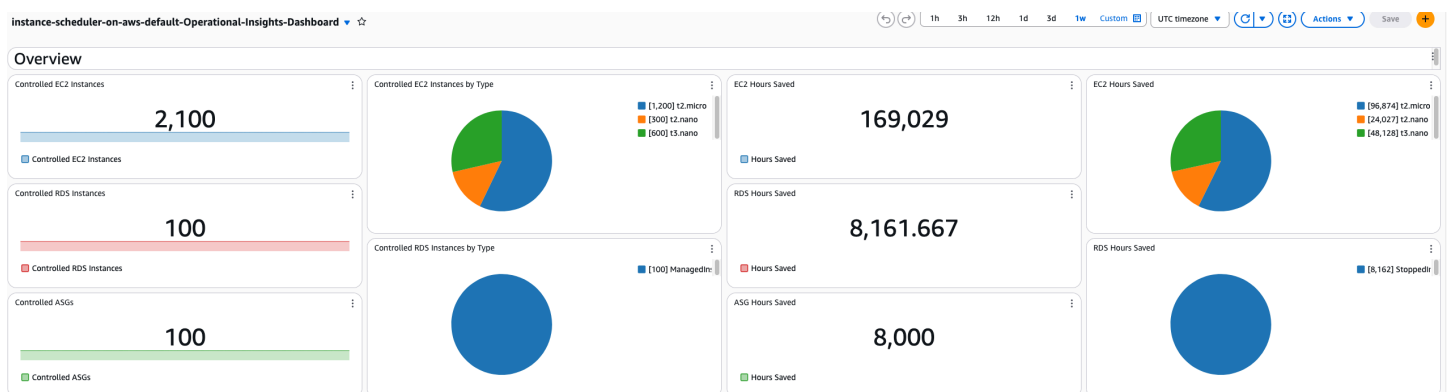
Dashboard mit operativen Erkenntnissen

Das Operational Insights-Dashboard bietet einen Überblick über die Lösungsleistung und die Kosteneinsparungen durch die geplante Instanzverwaltung.

Um auf das Dashboard zuzugreifen, stellen Sie sicher, dass Operational Monitoring in den Hub-Stack-Parametern auf „aktiviert“ gesetzt ist. Navigieren Sie zu „Dashboards“ CloudWatch und wählen Sie es im Navigationsmenü aus. Der Name des Dashboards lautet * {stack-name} -Operational-Insights-Dashboard*.

Das Dashboard zeigt die Anzahl der verwalteten Instanzen, die gespeicherten Betriebsstunden und Leistungskennzahlen der Lambda-Funktionen an.

Übersicht über das Dashboard mit operativen Erkenntnissen



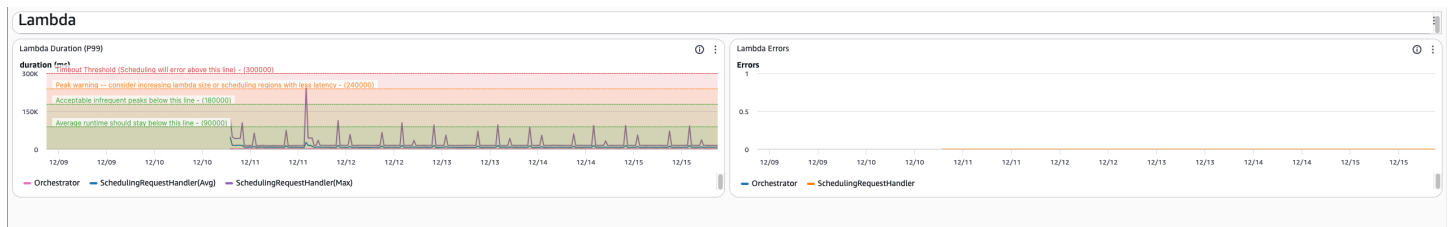
Note

Die Informationen in diesen Diagrammen hängen vom Planungsintervall ab, das auf dem Solution Hub-Stack konfiguriert ist. Bei der Aktualisierung des Planungsintervalls der Lösung

zeigt das Dashboard nur Planungsmetriken von der Zeit nach der letzten Aktualisierung bis zum Planungsintervall an.

Überwachen Sie die Lambda-Ausführungszeiten, um eine optimale Leistung sicherzustellen (siehe [Kontingente](#)). Wenn sich die Ausführungszeiten durchweg dem Timeout-Schwellenwert nähern, sollten Sie erwägen, die Lambda-Größeneigenschaft zu erhöhen oder Instance Scheduler in einer Region mit geringerer Latenz für Ihre verwalteten Regionen bereitzustellen.

Lambda-Metriken mit Dauer und Fehleranzahl



Zusätzliche Kosten im Zusammenhang mit dieser Funktion

Dieses operative Dashboard basiert auf benutzerdefinierten CloudWatch Kennzahlen, die von der Lösung erfasst wurden und für die zusätzliche Kosten anfallen. Diese Funktion kann ausgeschaltet werden, indem „Operational Monitoring“ auf dem Solution Hub-Stack deaktiviert wird. Diese Funktion kostet zusätzlich 3,00 USD/Monat zuzüglich zusätzlicher Skalierungskosten, die von der Größe Ihrer Bereitstellung abhängen. Die Kosten stellen sich wie folgt dar:

Benutzerdefiniertes CloudWatch Dashboard	3\$
Per-instance-type Metriken	0,90\$ pro Instanztyp*
API-Nutzung	~0,10 \$ pro aktivem Ziel**

*Diese Kosten werden pro Servicekategorie (EC2/RDS) und nur für Instance-Typen erfasst, die tatsächlich für die Planung verwendet werden.

*

EventBridge Ereignisse überwachen

Instance Scheduler veröffentlicht Planungs- und Registrierungsereignisse für EventBridge Event-Busse, um Einblick in den Lösungsbetrieb zu erhalten und die Integration mit anderen AWS-Services zu ermöglichen.

Event types (Ereignistypen)

Die Lösung veröffentlicht zwei Hauptkategorien von Ereignissen:

Ereignisse planen: Wird veröffentlicht, wenn Instance Scheduler Maßnahmen zum Starten, Stoppen oder Konfigurieren verwalteter Ressourcen ergreift. Diese Ereignisse enthalten Details zur Instanz, zum Zeitplan und zu den durchgeführten Aktionen. Starten, Stoppen oder Konfigurieren verwalteter Ressourcen. Zu diesen Ereignissen gehören Details zur Instanz, zum Zeitplan und zu den ergriffenen Maßnahmen.

Registrierungsereignisse: Werden veröffentlicht, wenn Ressourcen für die Planung auf der Grundlage von Tagging-Vorgängen registriert oder deren Registrierung aufgehoben wird.

Eventziel

LocalEvents IS-LocalEvents IS-Event-Busse: In jeder verwalteten Region jedes Mitgliedskontos (einschließlich des Hub-Kontos) wird ein Event-Bus bereitgestellt. Jeder Bus empfängt Ereignisse zur Planung von Aktionen und zur Registrierung von Ressourcen innerhalb dieser Region.

GlobalEvents IS-Event-Bus: Der IS-GlobalEvents Event-Bus im Hub-Konto empfängt eine Kopie jedes Ereignisses, das an einen beliebigen IS-LocalEvents Event-Bus gesendet wird, und ermöglicht so eine zentrale Überwachung über alle Konten und Regionen hinweg.

EventBridge Ereignisse verwenden

Sie können EventBridge Regeln erstellen, um:

- Überwachen Sie die Planungsvorgänge in Ihrer gesamten Infrastruktur
- Löst Benachrichtigungen aus, wenn Instanzen gestartet oder gestoppt werden
- Integration mit anderen AWS-Services für automatisierte Workflows
- Implementieren Sie Compliance-Überwachung und Warnmeldungen

Ereignisstruktur

Alle Ereignisse verwenden das EventBridge Standardformat. Die folgenden Beispiele zeigen die Struktur für jeden Ereignistyp:

Veranstaltung planen:

```
{
  "Source": "instance-scheduler",
  "DetailType": "Scheduling Action",
  "Resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"],
  "Detail": {
    "account": "123456789012",
    "region": "us-east-1",
    "service": "ec2",
    "resource_id": "i-1234567890abcdef0",
    "requested_action": "Start",
    "action_taken": "Started",
    "schedule": "office-hours"
  }
}
```

Veranstaltung zur Registrierung:

```
{
  "Source": "instance-scheduler",
  "DetailType": "Resource Registered",
  "Resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"],
  "Detail": {
    "account": "123456789012",
    "region": "us-east-1",
    "service": "ec2",
    "resource_id": "i-1234567890abcdef0",
    "schedule": "office-hours"
  }
}
```

Jede Veranstaltung enthält die folgenden Schlüsselfelder:

- **Source**— Identifiziert die Ereignisquelle als „Instance-Scheduler“
- **DetailType**— Gibt die Ereigniskategorie an: „Aktion planen“ für Instanzoperationen oder „Resource Registered“ für Tagging von Ereignissen

- **Resources-** Array mit den ARNs betroffenen AWS-Ressourcen
- **Detail-** Enthält die Nutzdaten des Ereignisses mit Konto-ID, Region, Servicetyp (ec2/rds), Ressourcen-ID und Namen des Zeitplans. Für die Planung von Ereignissen werden sowohl die angeforderte Aktion als auch das tatsächliche Ergebnis angezeigt

Mögliche `requested_action` Werte für die Planung von Ereignissen:

- **Start:** Der Scheduler beabsichtigt, die Instanz zu starten
- **Stop:** Der Scheduler wollte die Instanz stoppen
- **Configure:** Der Scheduler soll die Instanz konfigurieren

Mögliche `action_taken` Werte für die Planung von Ereignissen:

- **Started:** Die Instanz wurde gestartet
- **Stopped:** Die Instanz wurde gestoppt
- **Hibernated:** Die Instanz wurde in den Ruhezustand versetzt
- **Configured:** Die Instanzkonfiguration wurde geändert
- **Error:** Während des Planungsvorgangs ist ein Fehler aufgetreten

EventBridge Regeln erstellen

So überwachen Sie Instance Scheduler-Ereignisse:

1. Navigieren Sie in Ihrem AWS-Konto zur EventBridge Konsole
2. Erstellen Sie eine neue Regel, die entweder auf den `IS-GlobalEvents` Event Bus (für die zentrale Überwachung) oder den `IS-LocalEvents` Event Bus (für die lokale Überwachung) abzielt
3. Definieren Sie Ereignismuster, die den Ereignissen von Instance Scheduler entsprechen
4. Konfigurieren Sie Ziele wie SNS-Themen, Lambda-Funktionen oder Logs CloudWatch

Weitere Informationen zu EventBridge finden Sie unter [Was ist Amazon EventBridge?](#) im EventBridge Amazon-Benutzerhandbuch.

Fehlerbehebung

Dieser Abschnitt enthält Anweisungen zur Fehlerbehebung bei der Bereitstellung und Verwendung der Lösung.

Die Lösung bekannter Probleme enthält Anweisungen zur Behebung bekannter Fehler. Wenn diese Anweisungen Ihr Problem nicht lösen, erhalten Sie von [Contact Support](#) Anweisungen zum Öffnen einer Support-Anfrage für diese Lösung.

Lösung eines bekannten Problems

Problem: Instanzen werden nicht in einem Remote-Konto geplant (v1.4-v3.0)

Wenn Sie feststellen, dass Instanzen nicht in einem Remote-Konto geplant werden.

Auflösung

Aktualisieren Sie den Hub-Stack mit der sekundären Konto-ID oder führen Sie die folgende Aufgabe aus:

1. Navigieren Sie im primären Konto zur [CloudWatch Konsole](#)
2. Wählen Sie im Navigationsbereich Protokolle > Protokollgruppen aus.
3. Wählen Sie die Protokollgruppe mit dem Namen `<STACK_NAME>-logs`
4. Suchen Sie nach dem Protokollstream für die Konto-ID (Remote-Konto).
5. Wenn es beispielsweise keinen Log-Stream gibt, der mit der Konto-ID benannt ist, gehen Sie zur DynamoDB-Konsole und wählen Sie die Tabelle mit dem Namen aus. `<STACK_NAME>-<ConfigTable>-<RANDOM>`
6. Wählen Sie Elemente durchsuchen und anschließend Ausführen aus.
7. Wählen Sie den Artikeltyp Config aus.
8. Prüfen Sie, ob das Attribut `remote_account_ids` die Konto-ID hat.
9. Prüfen Sie, ob die Konto-ID in diesem Attribut nicht sichtbar ist.
10. Wenn die Lösung für AWS-Organisationen konfiguriert ist, deinstallieren Sie die Remote-Vorlage und installieren Sie sie erneut im Remote-Konto.

11. Wenn die Lösung für die Verwendung eines Remote-Kontos konfiguriert ist IDs, aktualisieren Sie den Cloudformation-Parameter Provide Organization Id OR List of Remote Account IDs mit der Liste der Konten IDs, für die die Instances geplant werden sollen und wo die Remote-Vorlage bereitgestellt wird.

Problem: Instanzen werden nicht geplant (v3.1+)

Wenn Sie feststellen, dass Instanzen nicht geplant werden.

Auflösung

1. Stellen Sie sicher, dass auf die Ressource ein ManagedBy IS-Tag angewendet wurde.
2. Wenn das Tag nicht vorhanden ist, löschen Sie das Schedule-Tag und erstellen Sie es neu, um die Registrierung erneut auszulösen.
3. Wenn das Tag immer noch nicht angewendet wird, überprüfen Sie, ob die Region für die Planung aktiviert ist:
 - a. Überprüfen Sie die hub/spoke Stack-Konfiguration für die Region, oder
 - b. Navigieren Sie zur [EventBridge Konsole](#) in derselben Region wie die Ressource und überprüfen Sie, ob der Standardereignisbus über Ereignisregeln mit dem Präfix IS-Tagging verfügt.
4. Wenn die Region nicht aktiviert ist, aktualisieren Sie den Instance Scheduler-Stack, sodass die Region in den Parameter regions aufgenommen wird. CloudFormation
5. Wenn das Problem weiterhin besteht, überprüfen Sie die [Lösungsverwaltungsprotokolle](#) auf Fehler bei der Hub-Registrierung.
6. Vergewissern Sie sich, dass in Ihrer Organisation keine Richtlinien gelten, die verhindern würden, dass Ereignisse von Ihrem Konto an das Solution Hub-Konto weitergeleitet werden.

Problem: Verschlüsselte EC2-Instances werden nicht gestartet

Instance Scheduler meldet, dass EC2-Instances mit verschlüsselten EBS-Volumes gestartet werden, aber sie werden nie wirklich gestartet.

Auflösung

Unter [Verschlüsselte EC2-EBS-Volumes](#) erfahren Sie, wie Sie Instance Scheduler Zugriff gewähren, um EC2-Instances mit verschlüsselten EBS-Volumes planen zu können

Problem: Unerwartete API-Kosten aufgrund von Informations-Tagging

Unerwartet hohe Kosten aufgrund von API-Aufrufen mit Tagging von AWS Resource Groups, AWS Config-Evaluierungen oder ähnlichen Abhilfemaßnahmen.

Auflösung

Instance Scheduler schreibt in jedem [Planungsintervall Informations-Tags](#) auf verwaltete Ressourcen. Wenn Ihre Umgebung Tag-Governance durch AWS Config-Regeln, Tag-Richtlinien oder automatische Problembhebungen erzwingt, stellen Sie sicher, dass die Tag-Schlüssel von Instance Scheduler zulässig sind. Eine vollständige Liste der Tag-Schlüssel und Anleitungen zur Konfiguration finden Sie unter Überlegungen zur [Tag-Governance](#).

Wenn Sie Ihre Tag-Governance-Richtlinien nicht aktualisieren können, deaktivieren Sie informatives Tagging, indem Sie den Parameter Informationstagging aktivieren auf dem Hub-Stack **No** auf „Tagging aktivieren“ setzen.

Problem: RDS-Instances werden nicht gestoppt, wenn „RDS-Snapshots erstellen“ aktiviert ist

RDS-Instances werden nicht gestoppt, und in den Scheduler-Protokollen der Lösung werden Fehler (AccessDenied) gemeldet, wenn der StopDBInstance Vorgang aufgerufen wird, weil keine Genehmigung vorliegt. `rds:CreateDBSnapshot`

Auflösung

Aktualisieren Sie die Lösung auf Version 3.0.5 oder neuer oder fügen Sie alternativ die `rds:CreateDBSnapshot` Berechtigung zur Scheduler-Rolle der Lösung in jedem geplanten Konto hinzu.

Kontaktieren Sie AWS Support.

Wenn Sie über [AWS Business Support+](#), [AWS Enterprise Support](#) oder [Unified Operations](#) verfügen, können Sie das AWS Support Center nutzen, um fachkundige Support zu dieser Lösung zu erhalten. In den folgenden Abschnitten finden Sie entsprechende Anweisungen.

Fall erstellen

1. Melden Sie sich im [Support Center](#) an.

2. Wählen Sie Create case (Fall erstellen) aus.

Wie können wir helfen?

1. Wählen Sie Technisch.
2. Wählen Sie für Service die Option Lösungen aus.
3. Wählen Sie als Kategorie Instance Scheduler auf AWS (Linux oder Windows) aus.
4. Wählen Sie unter Schweregrad die Option aus, die Ihrem Anwendungsfall am besten entspricht.
5. Wenn Sie den Service, die Kategorie und den Schweregrad eingeben, werden in der Benutzeroberfläche Links zu häufig gestellten Fragen zur Fehlerbehebung angezeigt. Wenn Sie Ihre Frage mit diesen Links nicht lösen können, wählen Sie Nächster Schritt: Zusätzliche Informationen.

Zusätzliche Informationen

1. Geben Sie als Betreff einen Text ein, der Ihre Frage oder Ihr Problem zusammenfasst.
2. Beschreiben Sie unter Beschreibung das Problem detailliert, einschließlich des Namens dieses Produkts und der Version, die Sie verwenden, z. B. dieses Beispiel: Instance Scheduler on AWS vX.Y.Z.
3. Wählen Sie Dateien anhängen.
4. Fügen Sie die Informationen bei, die der Support zur Bearbeitung der Anfrage benötigt.

Helfen Sie uns, Ihren Fall schneller zu lösen

1. Geben Sie die angeforderten Informationen ein.
2. Klicken Sie auf Next step: Solve now or contact us (Nächster Schritt): Jetzt lösen oder Support kontaktieren).

Löse es jetzt oder kontaktiere uns

1. Sehen Sie sich die Solve Now-Lösungen an.
2. Wenn Sie Ihr Problem mit diesen Lösungen nicht lösen können, wählen Sie Kontaktieren Sie uns, geben Sie die angeforderten Informationen ein und klicken Sie auf Absenden.

Aktualisieren Sie die Lösung

Instance Scheduler ist so konzipiert, dass es sicher ist, direkt mit AWS aktualisiert zu werden. CloudFormation Das allgemeine Verfahren hierfür ist wie folgt:

1. Melden Sie sich bei der [CloudFormation AWS-Konsole](#) an, auf der account/region Ihr Hub-Stack installiert ist `instance-scheduler-on-aws`, wählen Sie Stack aktualisieren aus und wählen Sie dann Stack aktualisieren aus.
2. Wählen Sie Direktes Update durchführen aus.
3. Wählen Sie Bestehende Vorlage ersetzen aus.
4. Gehen Sie unter Vorlage angeben wie folgt vor:
 - Wählen Sie Amazon S3 S3-URL aus.
 - Kopieren Sie den Link der [neuesten Vorlage](#).
 - Fügen Sie den Link in das Amazon S3 S3-URL-Feld ein.
 - Vergewissern Sie sich, dass die richtige Vorlagen-URL im Textfeld Amazon S3 S3-URL angezeigt wird, und wählen Sie Weiter. Wählen Sie erneut Next (Weiter).
5. Überprüfen Sie unter Parameter die Parameter für die Vorlage und ändern Sie sie nach Bedarf (alle erforderlichen Parameter-Aktualisierungen finden Sie unten in der Liste der wichtigsten Änderungen). Einzelheiten zu den einzelnen Parametern Weitere Informationen zu den Parametern finden Sie in [Schritt 1. Starten Sie den Instance-Scheduler-Stack](#).
6. Wählen Sie Weiter aus.
7. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
8. Überprüfen und bestätigen Sie die Einstellungen auf der Seite Review. Markieren Sie das Kästchen, um zu bestätigen, dass die Vorlage AWS Identity and Access Management (IAM) - Ressourcen erstellt.
9. Wählen Sie Änderungssatz anzeigen und überprüfen Sie die Änderungen.
10. Wählen Sie Stack aktualisieren, um den Stack bereitzustellen.

Sie können den Status des Stacks in der CloudFormation AWS-Konsole in der Spalte Status anzeigen. Sie sollten in wenigen Minuten den Status UPDATE_COMPLETE erhalten.

Wiederholen Sie die obigen Schritte für die `aws-instance-scheduler-remote` Stacks in jedem Ihrer Spoke-Konten.

Wichtigste Änderungen in bestimmten Versionen

Bei der Aktualisierung der Lösung können Sie direkt von einer älteren Version auf eine neuere Version aktualisieren, ohne dass wichtige Daten verloren gehen oder die Planung unterbrochen wird. Im Folgenden finden Sie eine Liste der Verhaltensänderungen und der wichtigsten Änderungen in den einzelnen Hauptversionen.

Ein vollständiges Changelog kann auf [der Seite der Lösung](#) eingesehen werden GitHub

v1.5.0

Version 1.5.0 ersetzt die Notwendigkeit, eine Liste von kontoübergreifenden Scheduling-Rollen bereitzustellen, und bietet ARNs nun die Möglichkeit, diese automatisch über Ihre AWS-Organisation zu verwalten. Wenn Sie AWS Organizations nicht verwenden möchten, können Sie stattdessen eine Liste mit Spoke-Konten bereitstellen, IDs und Instance Scheduler verwaltet die Scheduling-Rollen für Sie.

Bei der Aktualisierung auf Version 1.5.0 oder neuer müssen Sie:

1. Aktualisieren Sie die Hub-Vorlage mithilfe der normalen Aktualisierungsanweisungen und aktualisieren Sie dabei die folgenden Parameter:
 - a. Wählen Sie einen eindeutigen Namespace für die Lösung.
 - b. Wählen Sie aus, ob Sie in Zukunft AWS Organizations verwenden möchten, um die Spoke-Registrierung zu verwalten.
 - i. Wenn Sie Ja ausgewählt haben, ersetzen Sie das ID/Remote IDsOrganisationskonto durch die ID Ihrer AWS-Organisation.
 - ii. Wenn Sie Nein ausgewählt haben, ersetzen Sie OrganizationID/ RemoteAccount IDs durch eine kommasetrennte Liste der Konten Ihrer Spoke-Konten IDs .
2. Aktualisieren Sie alle Remote-Stacks mithilfe der normalen Aktualisierungsanweisungen und aktualisieren Sie dabei die folgenden Parameter:
 - a. Namespace — derselbe, den Sie für das Hub-Konto ausgewählt haben.
 - b. Verwenden Sie AWS Organizations — genauso wie das Hub-Konto.
 - c. Hub-Konto-ID — Konto-ID des Hub-Kontos (sollte gegenüber zuvor unverändert bleiben).

v3.0.0

v3.0.0 Fügt Unterstützung für EC2-Autoscaling-Gruppen hinzu und unterteilt die Lambda-Kernfunktion der Lösung in separate Funktionen mit eigenen Verantwortlichkeiten, um eine bessere Sicherheitsisolierung für jede einzelne Funktion zu gewährleisten. In dieser Version wird auch das Verhalten von Planungsprotokollen aktualisiert und enthält nun "SchedulingDecision" -Protokolle, um einen besseren Einblick in Planungsvorgänge zu erhalten.

V3.0.0 enthält die folgenden grundlegenden Änderungen im Vergleich zu früheren Versionen:

- [Die Funktion „CloudWatch Metriken“ in 1.5.x wurde durch das Operational Insights Dashboard ersetzt.](#)
- Zeitplanspezifische Metriken wurden von CloudWatch dort verschoben. Schedule/Service/MetricName → Schedule/Service/SchedulingInterval/MetricName
- Alle vorhandenen Metriken bleiben erhalten, aber neue Metriken werden jetzt unter dem neuen Namespace gesammelt und im Lösungs-Dashboard verfügbar gemacht.
- KMS-Schlüssel-ARNs zur Verwendung mit verschlüsselten EBS-Volumes auf EC2-DB-Instances müssen nun dem hub/spoke CloudFormation Stack in den jeweiligen Konten zur Verfügung gestellt werden. (Weitere Informationen finden Sie unter [Verschlüsselte EC2-EBS-Volumes](#).)
 - Wenn Sie EC2s mit verschlüsselten EBS-Volumes planen, müssen Sie die verwendeten KMS-Schlüsselarns in Ihre Stack-Parameter kopieren. hub/spoke
- Der CloudFormation Parameter für geplante Dienste wurde in einzelne Parameter für jeden unterstützten Dienst aufgeteilt.
 - Alle Dienste sind standardmäßig aktiviert und können einzeln deaktiviert werden.
- Instance Scheduler 3.0 ist nicht abwärtskompatibel mit älteren Versionen der Instance Scheduler CLI.
 - Sie müssen auf die neueste Version der Instance Scheduler-CLI aktualisieren, um weiterhin CLI-Befehle verwenden zu können.

Darüber hinaus wurde das Schema der Tabelle mit dem Wartungsfenster aktualisiert und wird im Rahmen des Updates ersetzt. Dadurch wird das Tracking für EC2-Wartungsfenster in den ersten Minuten nach dem Update auf Version 3.x zurückgesetzt. In seltenen Fällen kann dies dazu führen, dass Instanzen, die sich derzeit in einem Wartungsfenster befinden, unmittelbar nach dem Update vorzeitig beendet werden. Nachdem diese Daten neu generiert wurden, wird der Planungsvorgang wie gewohnt fortgesetzt.

v3.1.0

v3.1.0 überarbeitet die Kerninfrastruktur der Lösung, sodass AWS-Tagging-Ereignisse verwendet werden, um zu verfolgen, wann Ressourcen für die Planung markiert wurden. Bitte stellen Sie sicher, dass die Berechtigungen Ihrer Organisation das Senden dieser Tagging-Ereignisse von Mitgliedskonten an Ihr zentrales Hub-Konto zulassen.

Bei der Aktualisierung auf Version 3.1.0 oder neuer:

- Spoke-Konten deklarieren jetzt geplante Regionen unabhängig vom Hub-Konto. Jeder Spoke-Stack muss mithilfe des Parameters `Region (s)` angeben, welche Regionen in diesem Konto eingeplant werden sollen.
- Der AWS-Organisationsmodus ist jetzt für Bereitstellungen mit insgesamt mehr als 40 Konten erforderlich. Wenn Sie mehr als 40 Konten haben und den Organisationsmodus nicht verwenden, müssen Sie ihn während des Updates aktivieren.
- Wenn Sie EC2-Instances in AWS License Manager verwaltet haben, die Sie planen möchten, fügen Sie die License Manager Manager-Konfiguration ARNs dem License ARNs Manager-Konfigurationsparameter in Ihren hub/spoke CloudFormation Stacks hinzu. Weitere Informationen finden Sie unter [EC2 License Manager](#).
- Die Lösung wendet automatisch ein `ManagedBy IS`-Tag auf Ressourcen an, nachdem sie für die Planung markiert wurden, um anzuzeigen, dass sie vom Scheduler verwaltet werden.
- (In Version 3.2.0 wiederhergestellt) Die Größenänderung der geplanten Instanz (`Definition period-name@size` in einem Zeitplan) wurde in Version 3.1.0 vorübergehend entfernt, in Version 3.2.0 und neueren Versionen jedoch erneut implementiert. [Weitere Informationen](#) finden Sie unter Instanztyp.
- Das Auflisten von Mitgliedskonten über einen SSM-Parameter (Übergabe `{param: ssm-param-name}` an den `Accounts`-Parameter auf dem Hub-Stack) wird nicht mehr unterstützt. Alle vertrauenswürdigen Konten müssen bei der Bereitstellung an den Hub-Stack übergeben werden.
- Instance Scheduler benötigt während der Planung bis zu 6 eindeutige Tags für Ressourcen. Bitte stellen Sie sicher, dass in Kombination mit der übrigen Tagging-Strategie Ihres Unternehmens genügend Tagging-Kapazität für Ressourcen vorhanden ist.
- Zeitplanspezifische Kennzahlen wurden entfernt von. CloudWatch
- Die Lösungsprotokolle wurden in separate Gruppen für Verwaltung und Terminplanung zusammengefasst und für Abfragen mit CloudWatch Log Insights optimiert. Weitere Informationen finden Sie unter [Überwachung der Lösung](#).

- Start- und Stopp-Tags können nicht mehr über CloudFormation Parameter konfiguriert werden. Die Lösung verwendet jetzt feste Tagnamen mit umfassenderen Informationen für die Nachverfolgung von Planungsaktionen.

 **Important**

Instance Scheduler schreibt im Normalbetrieb bis zu 6 eindeutige Tags auf verwaltete Ressourcen. Stellen Sie sicher, dass Ihre Tag-Governance-Richtlinien (wie AWS Config-Regeln, Tag-Richtlinien oder automatische Problembeseitigung) so konfiguriert sind, dass sie diese Tags zulassen. Eine vollständige Liste der Tags und wichtiger Überlegungen zur Verwaltung finden Sie unter [Informations-Tags](#).

Deinstalliere die Lösung

Important

Achten Sie bei der Deinstallation der Lösung darauf, alle benutzerdefinierten Zeitplanstapel zu deinstallieren, bevor Sie die Lösung selbst deinstallieren.

Sie können die Instance Scheduler on AWS-Lösung über die AWS-Managementkonsole oder über die AWS-Befehlszeilenschnittstelle deinstallieren. Um die Lösung zu deinstallieren, löschen Sie den Hub-Stack in AWS Cloud Formation zusammen mit allen installierten Remote-Stacks. Anschließend können Sie alle Scheduling-Tags entfernen, die zu Planungszwecken auf Instances angewendet wurden.

Note

Wenn Protect DynamoDB Tables auf dem Hub-Stack der Lösung aktiviert ist, CloudFormation werden die DynamoDB-Tabellen und der KMS-Schlüssel der Lösung beibehalten, anstatt sie zu löschen. Wenn Sie diese Ressourcen löschen möchten, stellen Sie sicher, dass diese Eigenschaft auf Deaktiviert gesetzt ist, bevor Sie den Hub-Stack löschen. Alternativ können Sie sie manuell löschen, nachdem der Hub-Stack bereits gelöscht wurde.

Verwendung der AWS-Managementkonsole

1. Melden Sie sich bei der [CloudFormation AWS-Konsole](#) an.
2. Wählen Sie auf der Seite Stacks den Installations-Stack dieser Lösung aus.
3. Wählen Sie Löschen aus.

Verwenden der AWS-Befehlszeilenschnittstelle

Stellen Sie fest, ob die AWS-Befehlszeilenschnittstelle (AWS CLI) in Ihrer Umgebung verfügbar ist. Installationsanweisungen finden Sie unter [Was ist die AWS-Befehlszeilenschnittstelle](#) im AWS-CLI-Benutzerhandbuch. Nachdem Sie bestätigt haben, dass die AWS-CLI verfügbar ist, führen Sie den folgenden Befehl aus.

```
$ aws cloudformation delete-stack --stack-name  
  <installation-stack-name>
```

Entwicklerhandbuch

Dieser Abschnitt enthält den Quellcode für die Lösung und eine Liste der hier hinzugefügten Abschnitte sowie Links zu den einzelnen Unterthemen.

Quellcode

Besuchen Sie unser [GitHub Repository](#), um die Quelldateien für diese Lösung herunterzuladen und Ihre Anpassungen mit anderen zu teilen.

Die Vorlagen für den Instance Scheduler auf AWS werden mithilfe des [AWS-CDK](#) generiert. Weitere Informationen finden Sie unter dem Link: [README.md-Datei](#).

Referenz

[Dieser Abschnitt enthält Informationen zur Datenerfassung, Verweise auf verwandte Ressourcen und eine Liste der Entwickler, die zu dieser Lösung beigetragen haben.](#)

Datenerfassung

Diese Lösung sendet Betriebsmetriken (die „Daten“) über die Verwendung dieser Lösung an AWS. Wir verwenden diese Daten, um besser zu verstehen, wie Kunden diese Lösung und die damit verbundenen Dienstleistungen und Produkte nutzen. Die Erfassung dieser Daten durch AWS unterliegt der [AWS-Datenschutzerklärung](#).

Zugehörige Ressourcen

Der [Resource Scheduler](#) ähnelt dem Instance Scheduler auf AWS, seine Implementierung unterscheidet sich jedoch in folgenden Punkten:

Instance Scheduler auf AWS verwendet eine Lambda-Funktion, um die in seiner Konfiguration gespeicherten Zeitpläne häufig auszuwerten und zu überprüfen, ob sich die Instances im gewünschten Zustand befinden. Das Quicksetup für den Resource Scheduler verwendet Start- und Stoppzeiten, um Start- und Stoppaktionen mithilfe von SSM-Runbooks auszuführen. Dies tritt einmal auf, wenn die aktuelle Uhrzeit der Startzeit entspricht oder die aktuelle Uhrzeit nach der Startzeit liegt.

Instance Scheduler auf AWS ermöglicht derzeit die Planung für EC2-, RDS- und Aurora-Cluster. Der Resource Scheduler plant oder startet und stoppt nur EC2-Instances.

Verwenden Sie den Resource Scheduler, um EC2-Instances und start/stop diese zu bestimmten Zeiten zu identifizieren.

Verwenden Sie Instance Scheduler auf AWS, wenn die Konten regelmäßig nach Instances gescannt werden müssen start/stop .

In der Tabelle wird anhand von Szenarien angegeben, welche Lösung besser ist.

Szenario	Ressourcenplaner	Instance Scheduler auf AWS
Amazon Neptune Neptune-Instances planen	Nein	Ja

Szenario	Ressourcenplaner	Instance Scheduler auf AWS
Amazon DocumentDB DocumentDB-Instances planen	Nein	Ja
Auto Scaling Scaling-Gruppeninstanzen planen	Nein	Ja
Planen Sie EC2-Instanzen	Ja	Ja
Planen Sie RDS-Instanzen	Nein	Ja
Aurora-Cluster planen	Nein	Ja
Zeitpläne in einem einzigen Konto verwalten (Hub-Konto)	Nein	Ja
Zeitpläne in einzelnen Konten verwalten	Ja	Nein
Kalenderintegration ändern	Ja	Nein
Nur Aktionen starten und beenden	Ja	Nein
Überwachen Sie Instances regelmäßig und starten und beenden Sie sie basierend auf dem aktuellen Status der Instanz	Nein	Ja

Mitwirkende

- Arie Leeuwesteijn
- Mahmoud ElZayet
- Ruald Andreae
- Nikhil Reddy

- Kaleb Pearson
- Jason DiDomenico
- Max Granat
- Pratyusch Das
- Amanda Jones
- Kevin Hargita
- Lee Beomseok
- Abbe Wubshet

Überarbeitungen

Datum der Veröffentlichung: Oktober 2020

In der Datei [CHANGELOG.md](#) im GitHub Repository finden Sie alle wichtigen Änderungen und Aktualisierungen der Software. Das Changelog enthält eine übersichtliche Aufzeichnung der Verbesserungen und Korrekturen für jede Version.

Hinweise

Kunden sind dafür verantwortlich, Ihre eigene unabhängige Bewertung der Informationen in diesem Dokument vorzunehmen. Dieses Dokument: (a) dient nur zu Informationszwecken, (b) stellt aktuelle Produktangebote und Praktiken von AWS dar, die ohne vorherige Ankündigung geändert werden können, und (c) stellt keine Verpflichtungen oder Zusicherungen von AWS und seinen verbundenen Unternehmen, Lieferanten oder Lizenzgebern dar. AWS-Produkte oder -Services werden „wie sie sind“ ohne ausdrückliche oder stillschweigende Garantien, Zusicherungen oder Bedingungen jeglicher Art bereitgestellt. Die Verantwortlichkeiten und Verbindlichkeiten von AWS gegenüber seinen Kunden werden durch AWS-Verträge geregelt, und dieses Dokument ist weder Teil einer Vereinbarung zwischen AWS und seinen Kunden noch ändert es diese.

Instance Scheduler auf AWS ist unter den Bedingungen der [Apache License Version 2.0](#) lizenziert.

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.