



Guía de implementación

Programador de instancias de AWS



Programador de instancias de AWS: Guía de implementación

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Información general de la solución	1
Características y ventajas	2
Casos de uso	3
Conceptos y definiciones	3
Costo	4
Factores de escalamiento de costos	4
Calcular los objetivos de programación	4
Estrategias de optimización de costos	5
Ejemplos de precios de referencia (mensuales)	6
Estimación del costo de su implementación	8
Cuotas	8
Limitaciones de escalado	8
Consideraciones adicionales	9
Cuotas de servicio de AWS	10
Regiones de AWS admitidas	10
Cross-account programación de instancias mediante ID de cuenta o ID de organización de AWS	11
Habilitación de la programación entre cuentas mediante ID de cuenta	12
Habilitación de la programación entre cuentas mediante el ID de AWS Organizations	12
Administración de ID de cuenta con Almacén de parámetros de AWS Systems Manager	12
Servicios compatibles con la programación	13
Comportamiento de cierre de instancias	13
Amazon EC2	13
Amazon RDS, Amazon Neptune y Amazon DocumentDB	13
Ventana de mantenimiento de Amazon RDS	14
Grupos de Amazon EC2 Auto Scaling	15
Arquitectura	16
Diagrama de arquitectura	16
Consideraciones Well-Architected de diseño de AWS	18
Excelencia operativa	18
Seguridad	19
Fiabilidad	19
Eficiencia del rendimiento	19
Optimización de costos	20

Sostenibilidad	20
Tabla de configuración del programador	20
CLI del programador	21
Servicios de AWS usados en esta solución	21
Seguridad	23
AWS KMS	23
Amazon IAM	24
Volúmenes de EBS de EC2 cifrados	24
License Manager de EC2	26
Introducción	28
Información general del proceso de implementación	28
CloudFormation Plantillas de AWS	29
Paso 1: Lanzamiento de la pila central del programador de instancias	29
Paso 2 (opcional): Lanzamiento de la pila remota en cuentas secundarias	37
Configuración de la solución	40
Guía del operador	41
Configuración de programaciones	41
Uso de Infraestructura como código (recomendado)	42
Uso de la consola de Amazon DynamoDB y del programador de instancias en la CLI de AWS	42
Etiquetado de instancias para la programación	43
Establecimiento del valor de etiqueta	43
Instancias de EC2 con volúmenes de EBS cifrados	44
Instancias EC2 administradas en License Manager	44
Referencia de programación	44
Periodos	44
Zona horaria	45
Campo «Detener nuevas instancias»	45
Campo Hibernate	45
Campo Enforced	45
Campo Retain running	46
Campo de ventanas de mantenimiento de Systems Manager (aplicable solo a las instancias de EC2)	46
Tipo de instancia	47
Definiciones de programación	47
Referencia de período	50

Horas de inicio y detención	50
Días de la semana	51
Días del mes	52
Meses	52
Definiciones de período	52
Programaciones de ejemplo	55
Horas de trabajo estándar de 9:00 h a 17:00 h	55
Detención de instancias después de las 17:00 h	57
Detención de instancias durante el fin de semana	59
Recursos para soluciones	62
CLI del programador	64
Requisitos previos	64
Credenciales	64
Instalación de la CLI del programador	64
Estructura de comandos	65
Argumentos comunes	66
Comandos disponibles	67
create-period	67
create-schedule	69
delete-period	72
delete-schedule	72
describe-periods	73
describe-schedules	74
describe-schedule-usage	76
update-period	77
update-schedule	77
help	78
Actualización de valores de configuración global	79
Administración de programaciones mediante Infraestructura como código (IaC)	79
Cómo gestionar los errores de capacidad insuficiente de EC2	82
Configuración	82
Funcionamiento	82
Requisitos y limitaciones	83
Ejemplo	83
Programación de grupos de EC2 Auto Scaling	83
Información general de la programación de ASG	84

Definición de Running/Stopped para los ASG	84
Comportamiento del ASG Start/Stop	84
Supervisión de la solución	85
Registro y notificaciones	85
Etiquetas informativas	86
CloudWatch Registra las consultas de Insights	89
Panel de información operativa	90
Supervisión de eventos EventBridge	92
Resolución de problemas	95
Resolución de problemas conocidos	95
Problema: las instancias no se programan en una cuenta remota (v1.4-v3.0)	95
Resolución	95
Problema: las instancias no se están programando (versión 3.1 o posterior)	96
Resolución	96
Problema: Las instancias de EC2 cifradas no se inician	96
Resolución	96
Problema: el etiquetado informativo genera costes de API inesperados	96
Resolución	97
Problema: Las instancias de RDS no se detienen cuando está habilitado el permiso para crear instantáneas de RDS	97
Resolución	97
Póngase en contacto con AWS Support.	97
Crear caso	97
¿Cómo podemos ayudarle?	97
Información adicional	98
Ayúdenos a resolver su caso más rápido	98
Resuelva ahora o póngase en contacto con nosotros	98
Actualización de la solución	99
Cambios importantes en versiones específicas	100
v1.5.0	100
v3.0.0	101
v3.1.0	102
Desinstalar la solución	104
Uso de Consola de administración de AWS	104
Uso de la Interfaz de la línea de comandos de AWS	104
Guía para desarrolladores	106

Código fuente	106
Referencia	107
Recopilación de datos	107
Recursos relacionados	107
Colaboradores	109
Revisiones	110
Avisos	111
.....	cxii

Automatización del inicio y la detención de instancias de AWS

La solución del programador de instancias de AWS automatiza el inicio y la detención de varios servicios de AWS, incluidas las instancias de [Amazon Elastic Compute Cloud](#) (Amazon EC2) y de [Amazon Relational Database Service](#) (Amazon RDS).

Esta solución contribuye a reducir los costos operativos al detener recursos que no están en uso e iniciar recursos cuando se necesita su capacidad. Por ejemplo, una empresa puede usar el programador de instancias de AWS para detener las instancias de forma automática fuera del horario laboral todos los días. Si deja todas las instancias funcionando a pleno rendimiento, esta solución puede suponer un ahorro de costos de hasta un 70 % en el caso de las instancias que solo son necesarias durante el horario laboral habitual (ya que su uso semanal se reduce de 168 a 50 horas).

El programador de instancias de AWS utiliza etiquetas de recursos de Amazon Web Services (AWS) y [AWS Lambda](#) para detener y reiniciar instancias de forma automática en varias regiones y cuentas de AWS en función de una programación definida por el cliente. Esta solución también permite utilizar la hibernación para las instancias de EC2 detenidas.

En esta guía de implementación se proporcionan una descripción general de la solución del programador de instancias de AWS, su arquitectura y componentes de referencia, consideraciones para la planificación de la implementación y pasos de configuración para implementar la solución en la nube de AWS.

Esta guía está destinada a arquitectos, administradores y DevOps profesionales de infraestructuras de TI que desean implementar Instance Scheduler en AWS en su entorno.

Utilice esta tabla de navegación para encontrar rápidamente las respuestas a estas preguntas:

Si quiere...	Lea...
Conocer el costo de ejecutar esta solución. El costo estimado de ejecución de esta solución en la región Este de EE. UU. (Norte de Virginia) es de 13,15 USD al mes.	Costo
Comprender las consideraciones de seguridad de esta solución.	Well-Architected Seguridad y

Si quiere...	Lea...
	seguridad de AWS
Configuración de programaciones.	Tabla de configuración del programador
Saber qué regiones de AWS son compatibles con esta solución.	Regiones de AWS admitidas
Consulte o descargue la CloudFormation plantilla de AWS incluida en esta solución para implementar automáticamente los recursos de infraestructura (la «pila») de esta solución.	CloudFormation Plantillas de AWS
Acceder al código fuente y, opcionalmente, utilizar AWS Cloud Development Kit (AWS CDK) para implementar la solución.	GitHub repositorio

Características y ventajas

La solución del programador de instancias de AWS ofrece las siguientes características:

Cross-account programación de instancias

Esta solución incluye una plantilla que crea los roles de [AWS Identity and Access Management](#) (AWS IAM) necesarios para iniciar y detener instancias en cuentas secundarias. Para obtener más información, consulta la sección de [programación de Cross-account instancias](#).

Etiquetado automatizado

El programador de instancias de AWS puede agregar etiquetas de forma automática a todas las instancias que inicia o detiene. La solución también incluye macros que permiten agregar información de las variables a las etiquetas.

Configuración de programaciones o períodos mediante la CLI del programador

Esta solución incluye una interfaz de la línea de comandos (CLI) que proporciona comandos para configurar programaciones y períodos. La CLI permite a los clientes estimar el ahorro de costos para una programación determinada. Para obtener más información, consulte la [CLI del programador](#).

Administración de programaciones mediante Infraestructura como código (IaC)

Esta solución proporciona un recurso CloudFormation personalizado de AWS que puede utilizar para gestionar los horarios mediante la infraestructura como código (IaC). Para obtener más información, consulte [Administración de programaciones mediante Infraestructura como código](#).

Integración con Ventanas de mantenimiento de Systems Manager

En el caso de las instancias de Amazon EC2, el programador de instancias de AWS puede integrarse con ventanas de mantenimiento de [AWS Systems Manager](#), definidas en la misma región que esas instancias, para iniciarlas y detenerlas de acuerdo con la ventana de mantenimiento.

Casos de uso

Ejecución de instancias solo durante el horario laboral

Si deja todas las instancias funcionando a pleno rendimiento, esta solución puede suponer un ahorro de costos de hasta un 76 % en el caso de las instancias que solo son necesarias durante el horario laboral habitual (ya que su uso semanal se reduce de 168 a 40 horas). Para obtener más información, consulte [Programaciones de ejemplo](#).

Detención de instancias después del horario laboral

Si desea asegurarse de que las instancias de desarrollo se desactiven fuera del horario laboral hasta que vuelvan a ser necesarias, puede usar esta solución para establecer un período de finalización sin un período de inicio. Para obtener más información, consulte [Programaciones de ejemplo](#).

Conceptos y definiciones

En esta sección se describen los conceptos clave y se define la terminología específica de esta solución:

programar

Grupo de uno o varios períodos a los que está vinculada una instancia.

periodo

Períodos de ejecución definidos por una hora de inicio y de detención.

instancia

Recurso compatible que se puede programar. Por ejemplo, una instancia de Amazon EC2 o un clúster de Amazon RDS (Amazon EC2 y Amazon RDS).

horario laboral habitual

De 9:00 h a 17:00 h (ET) los días laborables

Para ver una referencia general de los términos de AWS, consulte el [Glosario de AWS](#).

Costo

Usted es responsable del coste de los servicios de AWS que se utilicen al ejecutar Instance Scheduler. Entender cómo los costes se escalan en función del tamaño de la implementación le ayuda a planificar y optimizar la implementación.

Factores de escalamiento de costos

Los costos de Instance Scheduler se escalan en función de varios factores:

Número de objetivos de programación: el número de combinaciones únicas de cuentas, regiones y servicios que se gestionan. Cada destino requiere una invocación de Lambda independiente por intervalo de programación.

Recursos por objetivo: la cantidad de recursos (instancias EC2, bases de datos RDS, etc.) de cada objetivo influye en el tiempo de ejecución de Lambda y en los costes de duración.

Complejidad de las métricas operativas: los costos de CloudWatch las métricas opcionales aumentan en función del número de tipos de instancias únicos y de los cronogramas activos de los que se hace un seguimiento a lo largo de la implementación.

Frecuencia de programación: la solución se ejecuta en función de la frecuencia configurada (predeterminada: 5 minutos). Las comprobaciones más frecuentes aumentan las invocaciones a Lambda de 24 veces al día (cada hora) a 288 veces al día (intervalos de 5 minutos).

Calcular los objetivos de programación

Un objetivo de programación es una combinación única de cuenta-región-servicio que contiene al menos una instancia gestionada activamente. Varias instancias dentro de la misma combinación de cuenta, región y servicio cuentan como un único objetivo de programación.

Ejemplo de cálculo:

- Cuenta A, us-east-1, 5 instancias EC2 = 1 destino de programación
- Cuenta A, us-east-1, 3 bases de datos RDS = 1 objetivo de programación
- Cuenta A, us-east-1, 2 grupos de Auto Scaling = 1 objetivo de programación
- Cuenta A, us-west-2, 2 instancias EC2 = 1 destino de programación
- Cuenta B, us-east-1, 10 instancias EC2 = 1 destino de programación

Total: 5 objetivos de programación

Esto significa que la solución invocará 5 funciones Lambda independientes por intervalo de programación para administrar todos los recursos en estas combinaciones de cuenta-región-servicio.

Note

Los objetivos pueden estar incluidos en el ámbito de la programación, pero no se consideran «activos» en los cálculos de costes hasta que se etiquete al menos un recurso para su programación en ese objetivo.

Para optimizar los costes, Instance Scheduler agrupa todos los RDS-related servicios de Amazon en una sola invocación. Por lo tanto, la programación de [Amazon RDS](#), [Amazon Aurora](#), [Amazon Neptune](#) y [Amazon DocDB](#) cuenta como un solo servicio de «RDS» para el cálculo de costes.

Estrategias de optimización de costos

1. Implemente en una región con precios de Lambda más bajos
2. Utilice la configuración de memoria Lambda de 512 MB predeterminada, a menos que sea necesario aumentarla debido a las limitaciones de la escala de un solo objetivo
3. Minimice la cantidad de programas y tipos de instancias únicos en uso activo
4. Ajuste la frecuencia de programación en función de sus requisitos
5. Desactive el panel de métricas operativas si no planea usarlo

Consulte la página web de precios de cada [servicio de AWS de esta solución](#).

Recomendamos elaborar un [presupuesto](#) a través del Explorador de costos de AWS para ayudar a administrar los costos. Los precios están sujetos a cambios.

Ejemplos de precios de referencia (mensuales)

Los siguientes ejemplos muestran cómo los costos se escalan en diferentes tamaños de implementación. Úselos como puntos de referencia para estimar los costos de su implementación específica.

Note

Todos los precios de referencia son estimaciones aproximadas del costo de los servicios principales utilizados por la solución.

Implementación pequeña (aproximadamente 9\$ al mes)

Este ejemplo representa una implementación típica de desarrollo o producción pequeña:

- 5 objetivos activos
- 20 recursos gestionados
- 3 horarios activos
- 2 tipos de instancias
- Intervalo de programación de 5 minutos
- Función Lambda de 512 MB, tiempo de ejecución medio de 5 segundos

Servicio de AWS	Costo mensual [USD]
AWS Lambda	~2,00 \$
AWS KMS	1,50 USD aprox.
CloudWatch Registros	~0,30 \$
CloudWatch Métricas	~5,30 \$
Amazon DynamoDB	~0,05 \$
Total:	~9,15 \$

Despliegue medio (~161 \$ mensuales)

Este ejemplo representa una implementación empresarial de tamaño mediano:

- 250 objetivos activos
- 1000 recursos gestionados
- 15 horarios activos
- 15 tipos de instancias
- Intervalo de programación de 5 minutos
- Función Lambda de 512 MB, tiempo de ejecución medio de 5 segundos
- 5 ventanas de mantenimiento de EC2

Servicio de AWS	Costo mensual [USD]
AWS Lambda	~95,00 \$
Amazon DynamoDB	~1,00 \$
CloudWatch Registros	10,00 USD aprox.
CloudWatch Métricas	40,00 USD aprox.
AWS KMS	~15,00 \$
Total:	~161,00 \$

Gran despliegue (~630 dólares mensuales)

Este ejemplo representa una implementación empresarial de gran tamaño:

- 1000 objetivos activos
- 5000 recursos gestionados
- 500 programas activos
- 50 tipos de instancias
- Intervalo de programación de 5 minutos

- Función Lambda de 512 MB, tiempo de ejecución medio de 5 segundos
- 100 ventanas de mantenimiento de EC2

Servicio de AWS	Costo mensual [USD]
AWS Lambda	~380,00 \$
Amazon DynamoDB	~5,00 \$
CloudWatch Registros	~50,00 \$
CloudWatch Métricas	~140,00 \$
AWS KMS	~55,00 \$
Total:	~630,00 \$

Estimación del costo de su implementación

Para estimar los costos de su implementación específica:

1. Cuente el total de los recursos gestionados (instancias de EC2, bases de datos de RDS, etc.)
2. Determine la cantidad de cuentas y regiones que administrará
3. Tenga en cuenta la frecuencia de programación requerida
4. Decida si necesita métricas operativas
5. Utilice los ejemplos de referencia anteriores para interpolar los costes esperados

Cuotas

Limitaciones de escalado

Instance Scheduler se basa en dos ejes principales para gestionar las implementaciones de grandes empresas:

Escalamiento vertical (recursos por objetivo)

El escalado vertical está limitado por la cantidad de recursos que una sola función Lambda de solicitud de programación puede procesar de manera eficiente dentro de un único objetivo de programación account/region (combinación /servicio).

[El programador de instancias está diseñado para poder gestionar miles de EC2, cientos de ASG y cientos de RDS dbs/clusters en un único objetivo de programación, pero puede estar limitado por la latencia entre regiones.](#)

Para garantizar un rendimiento óptimo, recomendamos supervisar el tiempo de ejecución de la solicitud de programación Lambda (consulte el [panel Operational Insights](#)). Recomendamos mantener el tiempo de ejecución promedio por debajo de los 90 segundos con un tiempo máximo máximo de no más de 4 minutos.

Escalado horizontal (número de objetivos)

El escalado horizontal está limitado por el número de [objetivos de programación activos](#) que se gestionan. Un objetivo activo es una combinación de account/region /service con al menos un recurso etiquetado activamente. El programador de instancias se puede implementar en muchas más cuentas y regiones, pero solo los objetivos con recursos etiquetados activamente afectan al rendimiento.

Con la cuota de simultaneidad Lambda predeterminada de 1000, puede ejecutar 1000 objetivos activos simultáneamente. Lambda pone en cola las ejecuciones adicionales automáticamente, lo que le permite escalar más allá de este límite. Recomendamos mantener el retraso acumulado por debajo de los 3 minutos para obtener un rendimiento óptimo.

Por ejemplo, con un tiempo de ejecución medio de 15 segundos y un límite de simultaneidad lambda de 1000, puede gestionar hasta 12 000 objetivos activos y, al mismo tiempo, mantener el retraso acumulado en menos de 3 minutos ($3 \text{ minutos} \div 15 \text{ segundos} \times 1000 = 12\,000 \text{ objetivos}$).

Para implementaciones más grandes, puede solicitar un aumento de la cuota de simultaneidad de Lambda a AWS Support.

Consideraciones adicionales

Etiquetas de recursos de AWS: los recursos de AWS suelen tener un límite de 50 etiquetas por recurso. El programador de instancias requiere 6 etiquetas informativas y de control para operar la

solución. Asegúrese de que sus recursos tengan una capacidad de etiquetas suficiente para incluir tanto las etiquetas del programador de instancias como su estrategia de etiquetado actual.

Límites de ejecución de Lambda: cada función Lambda del controlador de solicitudes de programación tiene un tiempo de espera de ejecución de 5 minutos.

Escalado de DynamoDB: la solución utiliza el escalado bajo demanda para sus tablas de Amazon [DynamoDB y ajusta automáticamente la capacidad](#) en función de la carga de trabajo.

Límites de velocidad de API: la limitación de las API de los servicios de AWS puede producirse en implementaciones muy grandes. La solución incluye una lógica de reintentos para gestionar las limitaciones temporales, pero una limitación excesiva puede reducir los límites superiores de escalado de la solución.

Cuotas de servicio de AWS

Cuotas de servicio para los servicios de AWS

Las cuotas de servicio (que también se denominan límites) establecen el número máximo de recursos u operaciones de servicio para su cuenta de AWS. Asegúrese de tener una cuota suficiente para cada uno de los servicios implementados en esta solución. Para obtener más información, consulte [Service Quotas de AWS](#).

CloudFormation Cuotas de AWS

Su cuenta de AWS tiene CloudFormation cuotas de AWS que debe tener en cuenta al lanzar la pila de esta solución. Si comprende estas cuotas, puede evitar errores de limitación que le impidan implementar esta solución correctamente. Para obtener más información, consulte [CloudFormation las cuotas de AWS](#) en la Guía del CloudFormation usuario de AWS.

Cuotas de AWS Lambda

Su cuenta tiene una cuota de ejecución simultánea de AWS Lambda predeterminada de 1000. Para implementaciones más grandes, recomendamos implementar Instance Scheduler en una cuenta dedicada para evitar competir con otras cargas de trabajo por la simultaneidad de Lambda. Este valor es ajustable. Para obtener más información, consulte la guía de [introducción a AWS Lambda](#).

Regiones de AWS admitidas

Puede implementar el programador de instancias en cualquier región de AWS, incluidas las regiones de AWS GovCloud (EE. UU.) y algunas [regiones opcionales \(regiones\)](#) que están deshabilitadas de


forma predeterminada). Tras implementar la solución, puede configurarla para aplicar las acciones de inicio o detención adecuadas a instancias de base de datos de EC2 y RDS etiquetadas en cualquier región de su cuenta. Si utiliza la programación de instancias entre cuentas, la solución aplica acciones a las instancias en todas las regiones configuradas de todas las cuentas.

 Important

Las acciones del programador de instancias de AWS afectan a las instancias debidamente etiquetadas en todas las regiones de AWS de su cuenta, aunque la función de Lambda se ejecute en una sola región.

Puede utilizar varias implementaciones de la solución para programar un gran número de instancias o instancias en diversas cuentas y regiones. Al implementar varios programadores, utilice un nombre de etiqueta diferente para cada pila y configure para cada implementación un conjunto de regiones no superpuestas.

Cada implementación revisa todas las instancias de cada región configurada en la cuenta a fin de encontrar la clave de etiqueta que identifica los recursos que debe programar. Si las regiones de varias implementaciones se superponen, varias implementaciones comprobarán cada instancia.

 Note

El programador de instancias de AWS puede segmentar instancias de cualquier región de suscripción voluntaria, incluso cuando las pilas de soluciones se implementan en regiones estándar de AWS.

Cross-account programación de instancias mediante ID de cuenta o ID de organización de AWS

Esta solución incluye una plantilla ([instance-scheduler-on-aws-remote.template](#)) que crea los roles de [AWS Identity and Access Management \(AWS IAM\)](#) y otros recursos necesarios para habilitar la solución de forma que comience a programar en las cuentas secundarias. Puede revisar y modificar los permisos en la plantilla remota antes de lanzar la pila.

Habilitación de la programación entre cuentas mediante ID de cuenta

Para aplicar programas automatizados de inicio y detención a recursos de cuentas secundarias:

1. Inicie sesión en la [consola de administración de AWS](#) y seleccione el botón para lanzar la plantilla [instance-scheduler-on-aws CloudFormation en](#) la cuenta principal.
2. Lance la plantilla remota ([instance-scheduler-on-aws-remote](#)) en cada cuenta secundaria aplicable. Al lanzar cada una de las pilas remotas, crea un nombre de recurso de Amazon (ARN) del rol entre las cuentas.
3. Actualice la pila de soluciones principal con el ID de cuenta en los parámetros Proporcionar ID de organización o Lista de ID de cuentas remotas para que la solución pueda realizar acciones de inicio y detención en instancias de las cuentas secundarias.

Habilitación de la programación entre cuentas mediante el ID de AWS Organizations

Para aplicar programas automatizados de inicio y detención a recursos de cuentas secundarias:

1. Inicie sesión en la [consola de administración de AWS](#) y seleccione el botón para lanzar la plantilla [instance-scheduler-on-aws CloudFormation en](#) la cuenta principal.
2. ¿Establecer el CloudFormation parámetro Using AWS Organizations? como Sí, e introduzca el identificador de la organización en los CloudFormation parámetros Proporcionar el identificador de la organización o la lista de identificadores de cuentas remotas.
3. Tras implementar la pila en la cuenta principal, lance la plantilla remota ([instance-scheduler-on-aws-remote](#)) en cada cuenta secundaria aplicable en la misma región que la solución de la cuenta principal. Cuando cada una de las pilas remotas se lance correctamente, la cuenta de la solución principal se actualizará con el ID de cuenta sin más cambios en la cuenta principal.

Administración de ID de cuenta con Almacén de parámetros de AWS Systems Manager

Utilice la característica Almacén de parámetros de AWS Systems Manager para almacenar los ID de las cuentas remotas. Puede almacenar ID de cuentas remotas como parámetro de lista donde cada elemento es un ID de cuenta, o bien como parámetro de cadena que contiene una lista delimitada

por comas de ID de cuentas remotas. El parámetro tiene el formato {param:_nombre_}, donde el nombre es el del parámetro en el almacén de parámetros.

Para utilizar esta característica, debe lanzar la pila central del programador de instancias de AWS en la misma cuenta que su almacén de parámetros.

Servicios compatibles con la programación

En la actualidad, el programador de instancias de AWS admite la programación de los siguientes servicios:

- Amazon EC2
- Grupos de Amazon EC2 Auto Scaling
- Amazon RDS
- Clústeres de Amazon Aurora
- Amazon DocumentDB
- Amazon Neptune

Comportamiento de cierre de instancias

Amazon EC2

Esta solución está diseñada para detener las instancias de EC2 de forma automática y supone que el comportamiento de cierre de las instancias está establecido en Detener, no en Terminar. Tenga en cuenta que no podrá reiniciar una instancia de Amazon EC2 una vez que se haya terminado.

De forma predeterminada, las instancias de EC2 están configuradas para detenerse al apagar, no para terminarse, pero puede [modificar este comportamiento](#). Por lo tanto, asegúrese de que las instancias que controla mediante el programador de instancias de AWS estén configuradas con el comportamiento de cierre Detener; de lo contrario, se terminarán.

Amazon RDS, Amazon Neptune y Amazon DocumentDB

Esta solución está diseñada para detener de forma automática las instancias de RDS, Neptune y DocDB, no para eliminarlas. Puede usar el parámetro de CloudFormation plantilla AWS Create RDS Instance Snapshot para crear instantáneas de instancias de base de datos de RDS antes de que la

solución detenga las instancias. Las instantáneas se conservan hasta la próxima vez que se detenga la instancia y se cree una nueva instantánea.

Note

Las instantáneas no están disponibles para los clústeres de Amazon Aurora. Puede usar el parámetro de plantilla Programar clústeres de Aurora para iniciar y detener instancias de base de datos de RDS que formen parte de un clúster de Aurora o que administren bases de datos de Aurora. Debe etiquetar el clúster (no las instancias individuales) con la clave de etiqueta que ha definido durante la configuración inicial y el nombre de la programación como valor de etiqueta para programar ese clúster.

Para obtener más información sobre las limitaciones al iniciar y detener una instancia de base de datos de RDS, consulte [Parada de una instancia de base de datos de Amazon RDS temporalmente](#) en la Guía del usuario de Amazon RDS.

Al detener una instancia de base de datos de RDS se borra la caché, lo que puede provocar un rendimiento más lento al reiniciarse la instancia.

Ventana de mantenimiento de Amazon RDS

Cada instancia de base de datos de RDS tiene una [ventana de mantenimiento](#) semanal durante la que se aplican los cambios del sistema. Durante la ventana de mantenimiento, Amazon RDS iniciará de forma automática las instancias que se hayan detenido durante más de siete días para aplicar tareas de mantenimiento. Amazon RDS no detendrá la instancia una vez que se complete el evento de mantenimiento.

La solución le permite especificar si se desea agregar la ventana de mantenimiento preferida de una instancia de base de datos de RDS como período de ejecución a su programación. La solución iniciará la instancia al comienzo de la ventana de mantenimiento y la detendrá al final de la misma si ningún otro período de ejecución especifica que la instancia deba ejecutarse y si el evento de mantenimiento se ha completado.

Si el evento de mantenimiento no se ha completado al final de la ventana de mantenimiento, la instancia se ejecutará hasta el intervalo de programación una vez que se haya completado el evento de mantenimiento. Para obtener más información acerca de la ventana de mantenimiento de Amazon RDS, consulte [Mantenimiento de una instancia de base de datos](#) de la Guía del usuario de Amazon RDS.

Grupos de Amazon EC2 Auto Scaling

Esta solución se ha diseñado para detener grupos de Amazon EC2 Auto Scaling de forma automática mediante acciones de escalado programadas. Puede utilizar la solución para configurar acciones de escalado programadas en el grupo de escalado automático (ASG). Cuando una acción de escalado programada detiene un ASG, sus capacidades mínima, deseada y máxima se establecerán en 0 hasta que el ASG se vuelva a iniciar de forma automática. Esto devolverá las capacidades mínima, deseada y máxima a sus valores originales.

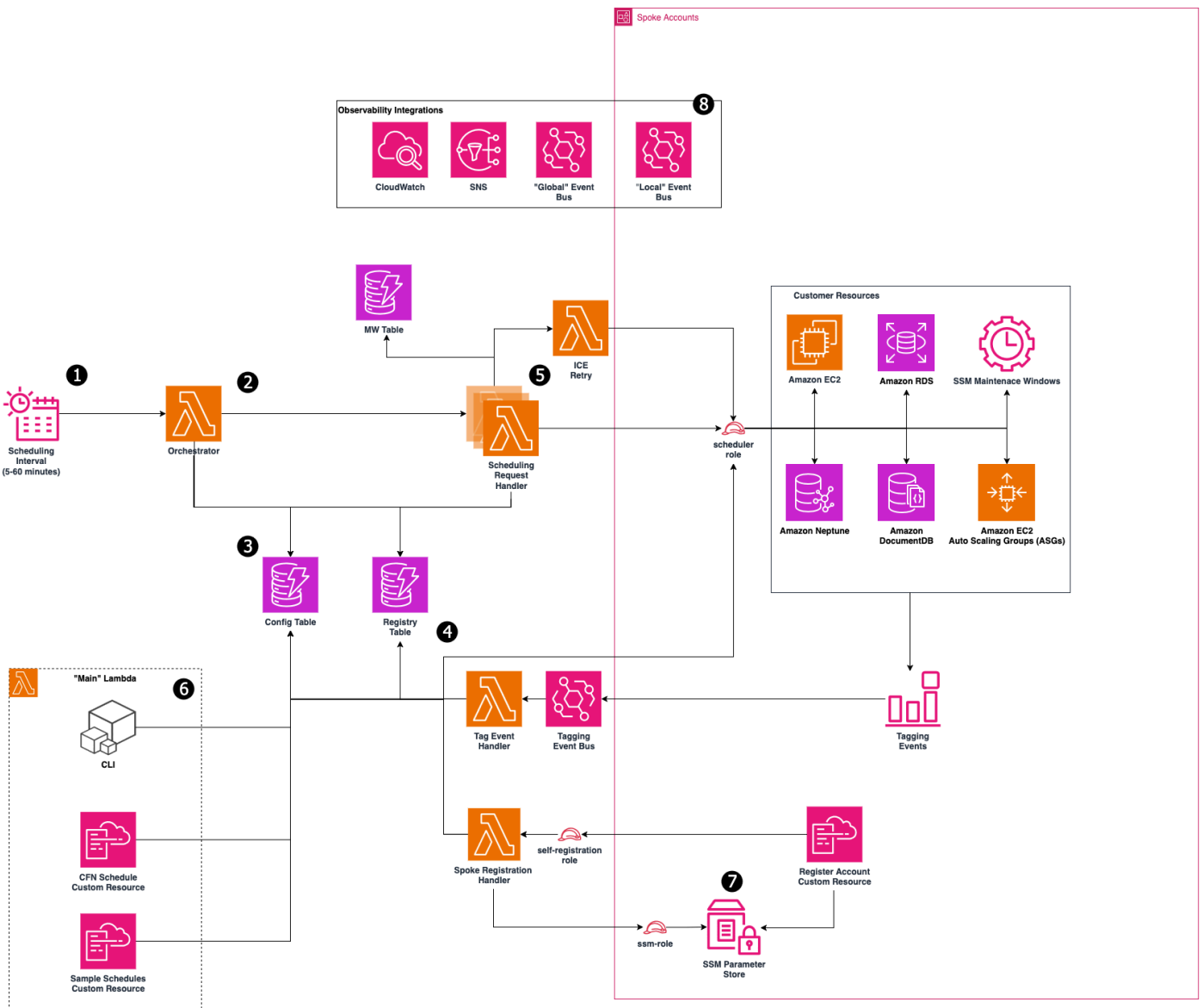
Arquitectura

En esta sección se proporciona un diagrama de arquitectura de implementación de referencia, [consideraciones de Well-Architected diseño de AWS](#), [componentes de seguridad](#), [configuraciones de planificadores](#) y [servicios de AWS utilizados en esta solución](#).


Diagrama de arquitectura

Al implementar esta solución, se implementan los siguientes componentes en su cuenta de AWS:

Programador de instancias en la nube de AWS



1. Una [EventBridge regla de Amazon](#) activa la función Lambda de orquestación a intervalos configurables (predeterminado: cada 5 minutos).
2. [La EventBridge regla invoca una función de orquestación de AWS Lambda que consulta la tabla de configuración de DynamoDB para identificar los objetivos de programación activos.](#) A continuación, el orquestador invoca funciones Lambda de programación paralela para cada objetivo activo.
3. Las definiciones y los períodos de programación se almacenan en una tabla de configuración de [Amazon DynamoDB](#). Puede definir cualquier número de programas y períodos en esta tabla para controlar cuándo se inician y se detienen las instancias.
4. Una tabla de registro de DynamoDB realiza un seguimiento automático de todos los recursos gestionados. Cuando los recursos se [etiquetan para su programación](#), se registran en esta tabla en respuesta a los eventos de etiquetado de AWS.
5. Cada función Lambda de programación describe los recursos etiquetados, evalúa sus planificaciones con respecto a la hora actual y ejecuta las acciones de inicio o parada adecuadas.
 - a. En el caso de las instancias EC2, si una operación de inicio falla debido a una capacidad insuficiente, se puede configurar la solución para intentar cambiar el tamaño de la instancia a [tipos de instancia alternativos](#) antes de volver a intentar la operación de inicio.
6. [La administración de la programación está disponible a través de la consola de DynamoDB, la herramienta CLI del programador o los recursos personalizados de AWS. CloudFormation](#) La solución se implementa con varios programas de ejemplo preconfigurados.
7. Cross-account las implementaciones utilizan una arquitectura hub-spoke en la que las cuentas radiales se autoregistran automáticamente en la cuenta hub. Los spoke stacks deben implementarse en la misma región que el hub stack y deben contar con la aprobación previa del hub stack o de miembros de la misma organización de [AWS](#).
8. La solución publica los [eventos de programación y registro](#) de los EventBridge autobuses en la cuenta hub (eventos globales) y en las cuentas de spoke (eventos locales por región).

 Note

Los CloudFormation recursos de AWS se crean a partir de construcciones ([AWS CDK](#)).

Todas las funciones de Lambda de esta solución utilizan AWS IAM para los requisitos de permisos de sus recursos y AWS KMS para el cifrado de las tablas de DynamoDB y [Amazon Simple Notification Service](#) (tema de Amazon SNS).

Cada vez que la solución realiza un intervalo de programación, comprueba el estado actual de cada instancia debidamente etiquetada con respecto al estado objetivo (definido por uno o varios [períodos](#) en una programación de la etiqueta de instancia) en la programación asociada. A continuación, el intervalo de programación aplica la acción de inicio o detención adecuada, según sea necesario.

Por ejemplo, si la función de Lambda se invoca un viernes a las 9:00 h (ET) e identifica una instancia de base de datos de RDS o EC2 detenida con la etiqueta Schedule=office-hours, buscará en Amazon DynamoDB los detalles de configuración de la programación del horario de oficina. Si la programación del horario de oficina contiene un período que indica que la instancia debe ejecutarse de lunes a viernes de 9:00 a 17:00 h (ET), la función de Lambda iniciará esa instancia.

La función Lambda también registra información sobre sus recursos y la muestra en un panel opcional de [Amazon CloudWatch Custom](#). En la información registrada se incluyen el número de instancias etiquetadas para cada programación, los tamaños de esas instancias y si actualmente se encuentran en estado de ejecución o detenidas. Para obtener más información sobre este panel personalizado, consulte [Panel de información operativa](#).

Note

Detener una instancia de Amazon EC2 y terminar una instancia de Amazon EC2 son dos cosas distintas. De forma predeterminada, las instancias de Amazon EC2 están configuradas para detenerse al apagar, no para terminarse, pero este comportamiento se puede modificar. Antes de usar esta solución, compruebe que las instancias están configuradas para detenerse o terminarse, según corresponda.

Consideraciones Well-Architected de diseño de AWS

Diseñamos esta solución con las mejores prácticas del [Well-Architected marco de AWS](#), que ayuda a los clientes a diseñar y operar cargas de trabajo confiables, seguras, eficientes y rentables en la nube.

En esta sección, se describe cómo se aplicaron los principios de diseño y las mejores prácticas del Well-Architected marco al crear esta solución.

Excelencia operativa

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del [pilar de excelencia operativa](#).

- La solución envía las métricas a Amazon CloudWatch para proporcionar observabilidad a sus componentes (como su infraestructura y las funciones Lambda).
- AWS X-Ray rastrea las funciones de Lambda.
- Se utiliza Amazon SNS para los informes de errores.

Seguridad

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del [pilar de seguridad](#).

- Todas las comunicaciones entre servicios utilizan roles de IAM.
- Todas las comunicaciones entre varias cuentas utilizan roles de IAM.
- Todos los roles que utiliza la solución siguen el acceso con privilegios mínimos. En otras palabras, solo contienen los permisos mínimos necesarios para que el servicio pueda funcionar correctamente.
- Todo el almacenamiento de datos, incluidas las tablas de DynamoDB, cuenta con cifrado en reposo.

Fiabilidad

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del [pilar de fiabilidad](#).

- La solución utiliza servicios de AWS sin servidor siempre que es posible (como Lambda y DynamoDB) para garantizar una alta disponibilidad y recuperación en caso de errores del servicio.
- Para el procesamiento de datos se utilizan funciones de Lambda. La solución almacena los datos en DynamoDB, por lo que permanecen en varias zonas de disponibilidad de forma predeterminada.

Eficiencia del rendimiento

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del [pilar de eficiencia del rendimiento](#).

- La solución utiliza una arquitectura sin servidor.

- Puede lanzar la solución en cualquier región de AWS que admita los servicios de AWS utilizados en esta (como Lambda y DynamoDB). Para obtener más información, consulte [Regiones de AWS admitidas](#).
- La solución se prueba e implementa de forma automática todos los días. Nuestros arquitectos de soluciones y expertos en la materia revisan la solución en busca de áreas para experimentación y mejora.

Optimización de costos

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del [pilar de optimización de costos](#).

- La solución utiliza una arquitectura sin servidor y los clientes solo pagan por lo que usan.
- El valor predeterminado de la capa de computación es Lambda, que utiliza un modelo de pago por uso.

Sostenibilidad

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las mejores prácticas del [pilar de sostenibilidad](#).

- La solución utiliza servicios administrados y sin servidor a fin de minimizar el impacto ambiental de los servicios de backend.
- El diseño sin servidor de la solución tiene como objetivo reducir la huella de carbono en comparación con la de los servidores en las instalaciones en continuo funcionamiento.

Tabla de configuración del programador

Cuando se implementa, el programador de instancias de AWS crea una tabla de Amazon DynamoDB que contiene valores de configuración global.

Los elementos de configuración global contienen un atributo de tipo con el valor config en la tabla de configuración. Las programaciones y los períodos contienen atributos de tipo con los valores schedule y period, respectivamente. Puede agregar, actualizar o eliminar programaciones y períodos de la tabla de configuración mediante la consola de DynamoDB o la [interfaz de la línea de comandos](#)

de la solución. Sin embargo, no es necesario editar ningún elemento con un tipo config, ya que los administra la solución.

CLI del programador

La solución incluye una CLI que proporciona comandos para configurar programaciones y períodos. La CLI permite estimar el ahorro de costos para un programación determinada. Las estimaciones de costos que proporciona la CLI de programación son solo orientativas. Para obtener más información sobre la configuración y el uso de la CLI del programador, consulte [CLI del programador](#).

Servicios de AWS usados en esta solución

Servicio de AWS	Description (Descripción)
AWS Lambda	Principal. La solución implementa una función Lambda que contiene toda la lógica necesaria para programar las instancias y administrar las actualizaciones de CloudFormation la pila mediante una función de recursos personalizada.
Amazon DynamoDB	Principal. La solución crea tablas de DynamoDB para almacenar la configuración de programación, la información de estado, las últimas acciones realizadas de las instancias y una tabla para almacenar ventanas de mantenimiento de Systems Manager con fines de programación.
Amazon CloudWatch	Principal. La solución almacena registros de depuración e información.
AWS IAM	Principal. La solución usa IAM con el fin de obtener permisos para la programación de instancias.
Amazon SNS	Principal. La solución crea un tema de SNS a fin de enviar mensajes de error para que los

Servicio de AWS	Description (Descripción)
	usuarios se suscriban y solucionen cualquier problema en caso de errores.
AWS KMS	Principal. La solución crea una clave de AWS KMS para cifrar el tema de SNS.
Amazon EventBridge	Principal. La solución crea una EventBridge solución: crea reglas EventBridge programadas que invocan AWS lambda en un intervalo constante»
AWS Systems Manager	Admite. Proporciona monitorización de recursos a nivel de aplicación, así como visualización de operaciones de recursos y datos de costos.
Amazon EC2	Programado. La solución se utiliza para iniciar y detener instancias de EC2. Las instancias se identifican mediante etiquetas específicas key/values que se configuran en la solución.
Amazon RDS	Programado. La solución se utiliza para cambiar el estado de las instancias de base de datos de RDS a Disponible o Detenido. Las instancias se identifican mediante etiquetas específicas key/values que se configuran en la solución.
Amazon Aurora	Programado. La solución se utiliza para cambiar el estado de los clústeres de Aurora a Disponible o Detenido. Los clústeres se identifican mediante etiquetas específicas key/values que se configuran en la solución.

Servicio de AWS	Description (Descripción)
Amazon Neptune	Programado. La solución se utiliza para cambiar el estado de las instancias de Neptune a Disponible o Detenido. Las instancias se identifican mediante etiquetas específicas key/values que se configuran en la solución.
Amazon DocumentDB	Programado. La solución se utiliza para cambiar el estado de las instancias de DocumentDB a Disponible o Detenido. Las instancias se identifican mediante etiquetas específicas key/values que se configuran en la solución.
Grupos de Amazon EC2 Auto Scaling	Programado. La solución se utiliza con el fin de administrar reglas de escalado programado o para grupos de escalado automático de EC2. Estas reglas agruparán a start/stop Auto Scaling de acuerdo con un cronograma asociado. Los grupos se identifican mediante etiquetas específicas key/values que se configuran en la solución.

Seguridad

Cuando crea sistemas en la infraestructura de AWS, las responsabilidades de seguridad se comparten entre usted y AWS. Este [modelo de responsabilidad compartida](#) reduce su carga operativa, ya que AWS opera, administra y controla los componentes, desde el sistema operativo host y la capa de virtualización hasta la seguridad física en las instalaciones en las que operan los servicios. Para obtener más información sobre la seguridad de AWS, visite [Seguridad en la nube de AWS](#).

AWS KMS

La solución crea una clave administrada tanto por AWS como por el cliente, que se usa a fin de configurar el cifrado del servidor para el tema de SNS y las tablas de DynamoDB.

Amazon IAM

Las funciones Lambda de la solución requieren permisos para acceder a los recursos de la cuenta hub y acceder a los parámetros de get/put Systems Manager, acceder a los grupos de CloudWatch registros, a la clave encryption/decryption de AWS KMS y publicar mensajes en SNS. Además, el programador de instancias también creará funciones de programación en todas las cuentas administradas que permitirán acceder a start/stop EC2, RDS, los recursos de escalado automático y las instancias de base de datos, modificarán los atributos de las instancias y actualizarán las etiquetas de esos recursos. La solución proporciona todos los permisos necesarios al rol de servicio de Lambda creado como parte de la plantilla de solución.

Durante el despliegue, Instance Scheduler implementará funciones de IAM con ámbito reducido para cada una de sus funciones de Lambda, junto con las funciones de programador que solo se pueden asumir mediante Lambdas de programación específicas en la plantilla de hub implementada. Los nombres de estos roles de programación siguen los patrones `{namespace}-Scheduler-Role` y `{namespace}-ASG-Scheduling-Role`.

[Para obtener información detallada sobre el permiso otorgado a cada función de servicio, consulte las plantillas. CloudFormation](#)

Volúmenes de EBS de EC2 cifrados

Al programar instancias de EC2 asociadas a volúmenes de EBS cifrados por AWS KMS, debe conceder permiso al programador de instancias para usar las claves de AWS KMS asociadas. Esto permite a Amazon EC2 descifrar los volúmenes de EBS asociados mientras la función permanece iniciada. Este permiso debe concederse al rol de programación en la misma cuenta que las instancias de EC2 que utilizan la clave.

Para conceder permiso para usar una clave de AWS KMS con Instance Scheduler, añada el ARN de la clave de AWS KMS a la pila del programador de instancias (hub o spoke) en la misma cuenta que las instancias de EC2 utilizando las claves:

ARN de clave de KMS para 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

Esto generará automáticamente la siguiente política y la agregará al rol de programación correspondiente a esa cuenta:

```
{
  "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"
    }
  ]
}
```

```
]
}
```

License Manager de EC2

Al programar instancias EC2 que se administran en AWS License Manager, debe conceder permiso al Programador de Instancias para usar las configuraciones de licencia asociadas. Esto permite que la solución inicie y detenga las instancias correctamente y, al mismo tiempo, mantenga el cumplimiento de las licencias. Este permiso debe concederse a la función de programación en la misma cuenta que las instancias EC2 que utilizan License Manager.

Para conceder permiso para usar AWS License Manager con Instance Scheduler, añada los ARN de configuración del License Manager a la pila (hub o spoke) del programador de instancias en la misma cuenta que las instancias de EC2 mediante License Manager:

ARN de configuración de License Manager para 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

Esto generará automáticamente la siguiente política y la agregará al rol de programación correspondiente a esa cuenta:

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

Para obtener más información sobre los permisos de License Manager, consulte [Administración de identidad y acceso para AWS License Manager](#) en la Guía del usuario de AWS License Manager.



Introducción

En esta guía se incluye una breve descripción general e instrucciones para implementar la solución de forma rápida. Esta solución utiliza [CloudFormation plantillas y pilas de AWS](#) para automatizar su implementación. Las CloudFormation plantillas especifican los recursos de AWS incluidos en esta solución y sus propiedades. La CloudFormation pila aprovisiona los recursos que se describen en las plantillas.

Información general del proceso de implementación

Important

Esta solución incluye una opción para enviar métricas operativas anonimizadas a AWS. Utilizamos estos datos para comprender mejor cómo utilizan los clientes esta solución, así como los servicios y productos relacionados. Los datos recopilados a través de esta encuesta son propiedad de AWS. La recopilación de datos está sujeta al [Aviso de privacidad](#).

Para excluirse de esta función, descargue la plantilla, modifique la sección de CloudFormation mapeo de AWS y, a continuación, utilice la CloudFormation consola de AWS para cargar la plantilla actualizada e implementar la solución.

Siga las instrucciones paso a paso de esta sección para configurar e implementar la solución en su cuenta.

Tiempo de implementación: aproximadamente de cinco a diez minutos (sin incluir la configuración).

[Paso 1: Lanzamiento de la pila del programador de instancias](#)

1. Inicie la CloudFormation plantilla de AWS en su cuenta de AWS.
2. Especifique valores para los parámetros requeridos.
3. Revise el resto de los parámetros de la plantilla y ajústelos si es necesario.

[Paso 2 \(opcional\): Lanzamiento de la pila remota en cuentas secundarias](#)

1. Inicie la CloudFormation plantilla de AWS en su cuenta de AWS.

2. Especifique valores para los parámetros requeridos.

CloudFormation Plantillas de AWS

Esta solución utiliza [CloudFormation plantillas y pilas de AWS](#) para automatizar su implementación. Las CloudFormation plantillas especifican los recursos de AWS incluidos en esta solución y sus propiedades. La CloudFormation pila aprovisiona los recursos que se describen en las plantillas.

Puede descargar las CloudFormation plantillas de esta solución antes de implementarla.

[View template](#)

instance-scheduler-on-aws.template: utilice esta plantilla para lanzar la solución y todos los componentes asociados. La configuración predeterminada implementa una función de AWS Lambda, una tabla de Amazon DynamoDB, un evento de CloudWatch Amazon y métricas personalizadas CloudWatch , pero también puede personalizar la plantilla en función de sus necesidades específicas.

[View template](#)

instance-scheduler-on-aws-remote.template: utilice esta plantilla para lanzar el rol de acceso entre cuentas que utiliza la solución para programar instancias en cuentas radiales. En el caso de las implementaciones que utilizan AWS Organizations, la implementación de la plantilla también registra la cuenta radial en el centro, lo que no requiere ninguna configuración manual.

Note

Si ya ha implementado esta solución anteriormente, consulte [Actualización de la solución](#) para obtener instrucciones de actualización.

Paso 1: Lanzamiento de la pila central del programador de instancias

Siga las instrucciones paso a paso de esta sección para implementar la solución en su cuenta.

Tiempo de implementación: aproximadamente cinco minutos



1. Inicie sesión en la [consola de administración de AWS](#) y seleccione el botón para lanzar la plantilla de AWS* `instance-scheduler-on-aws.template*`. CloudFormation
2. La plantilla se lanza en la región Este de EE. UU. (Norte de Virginia) de forma predeterminada. Para lanzar la solución en una región de AWS diferente, utilice el selector de regiones de la barra de navegación de la consola.
3. En la página Crear pila, verifique que la dirección URL de la plantilla correcta se encuentre en el cuadro de texto URL de Amazon S3 y elija Siguiente.
4. En la página Especificar los detalles de la pila, especifique un nombre para la pila. Para obtener información acerca de las limitaciones de nomenclatura de caracteres, consulte [IAM y cuotas de AWS STS](#) de la Guía del usuario de AWS Identity and Access Management.
5. En Parámetros, revise los parámetros de esta plantilla de solución y modifíquelos según sea necesario. Esta solución utiliza los siguientes valores predeterminados.

Parámetro	Predeterminado	Description (Descripción)
Infraestructura		
Namespace	default	Proporcione un identificador único para diferenciar entre varias implementaciones de una solución (sin espacios). Ejemplo: Dev.
Uso de AWS Organizations	No	Utilice AWS Organizations para automatizar el registro de cuentas radiales.
ID de cuenta de la organización ID/Remote	<Optional Input>	Si utiliza AWS Organizations, este campo es obligatorio. Proporcione el identificador de la organización, por ejemplo, o-xxxxyyy . De lo contrario, proporcione una lista separada por

Parámetro	Predeterminado	Description (Descripción)
		comas de identificadores de cuentas de confianza que puedan registrarse por sí mismos para la programación (máximo 40), como 1111111111, 2222222222
Clave de etiqueta de programación	Schedule	La clave de etiqueta que lee la solución a fin de determinar la programación para un recurso. El valor de un recurso especifica el nombre de la programación. Si decide modificar el valor predeterminado, asigne un nombre que sea fácil de aplicar de forma coherente y correcta en todas las instancias necesarias. Nota: La clave de etiqueta distingue entre mayúsculas y minúsculas.
Conserve los datos y los registros	Enabled	Habilite la protección contra eliminación para las tablas de DynamoDB que utiliza la solución. De este modo, las tablas se retienen al eliminar esta pila. Para eliminar las tablas al eliminar la pila, deshabilite primero este parámetro.

Configuración global

Parámetro	Predeterminado	Description (Descripción)
Habilite la programación	Yes	NoConfigúrelo para suspender todas las operaciones de programación.
Zona horaria predeterminada	UTC	Identificador de zona horaria de IANA (Autoridad Internacional de Números Asignados) predeterminado para programaciones que no especifican una zona horaria. Para obtener una lista de identificadores de zona horaria válidos, consulte la columna TZ identifier de List of tz database time zones .
Intervalo de programación (minutos)	5	Intervalo en minutos entre las ejecuciones del programador. Los intervalos más cortos aumentan la precisión y la capacidad de respuesta, pero también aumentan los costes. Los despliegues de producción requieren un mínimo de 5 minutos para un funcionamiento estable; los valores más cortos son solo para pruebas a pequeña escala.

Parámetro	Predeterminado	Description (Descripción)
Habilitar ventanas de mantenimiento de SSM de EC2	No	Permita que las programaciones especifiquen uno o varios nombres de las ventanas de mantenimiento de Systems Manager. El programador de instancias de AWS se asegurará entonces de que las instancias etiquetadas con esa programación se inicien al menos diez minutos antes que las ventanas de mantenimiento asociadas.
Creación de instantáneas de instancias de RDS al detenerse	No	Elija si quiere crear una instantánea antes de detener las instancias de base de datos de RDS. Nota: Las instantáneas no están disponibles para los clústeres de Amazon Aurora.
Prefijo del nombre de acción de ASG	IS-	Prefijo que la solución utiliza al asignar nombres a las acciones de escalado programadas para grupos de escalado automático. La solución agregará y eliminará las acciones con este prefijo según sea necesario.
Clave de etiqueta programada de ASG	scheduled	Obsoleto. Este parámetro existe únicamente con fines de migración y no debe editarse.

Parámetro	Predeterminado	Description (Descripción)
Hub-Account Programación		
Regiones	<Optional Input>	Lista de regiones en las que se programarán las instancias. Por ejemplo, <code>us-east-1</code> , <code>us-west-1</code> . NOTA: Si deja este parámetro en blanco, la solución usa la región actual.
ARN de clave de KMS para EC2	<Optional Input>	Comma-separated lista de ARN de KMS para conceder al programador de instancias en AWS kms: CreateGrant permisos para proporcionar al servicio EC2 permisos de descifrado para volúmenes de EBS cifrados. Esto permite al programador iniciar instancias de EC2 con volúmenes de EBS cifrados asociados. Proporcione (*) para conceder acceso limitado a todas las claves de KMS; déjelo en blanco para deshabilitarlo. Para obtener más información sobre la política creada, consulte Volúmenes de EBS de EC2 cifrados .

Parámetro	Predeterminado	Description (Descripción)
RANs de License Manager para EC2	<Optional Input>	Comma-separated lista de ARN de configuración de License Manager para conceder permisos al Programador de Instancias para iniciar instancias EC2 administradas por License Manager. Déjelo en blanco para deshabilitarlo. Para obtener más información, consulte EC2 License Manager .
Supervisión		
Habilite el etiquetado informativo	Yes	Cuando está activado, el programador de instancias escribe etiquetas informativas en los recursos gestionados que indican la última acción de programación realizada y los errores encontrados. Para obtener más información, consulte las etiquetas informativas .
Habilite los CloudWatch registros de depuración	No	Habilite el registro a nivel de depuración en los registros. CloudWatch
Período de retención de registros (días)	30	El período de retención de CloudWatch registros para los registros en días.

Parámetro	Predeterminado	Description (Descripción)
Supervisión operativa	Enabled	Implemente un panel de información operativa para CloudWatch recopilar datos métricos personalizados sobre el funcionamiento de la solución. Si lo desea, puede deshabilitar el panel para reducir los costos asociados .
Otros		
SchedulingRequestHandler Tamaño de memoria (MB)	512	El tamaño de la memoria de la función de AWS Lambda que programa los recursos. Aumente este valor si el uso de memoria o los tiempos de espera son elevados.
Tamaño de memoria de Orchestrator (MB)	512	El tamaño de la memoria de la función Lambda del orquestador. Aumente este valor si el uso de memoria o los tiempos de espera son elevados.

6. Elija Siguiente.
7. En la página Configurar opciones de pila, elija Siguiente.
8. En la página Revisar y crear, revise y confirme la configuración. Seleccione la casilla para reconocer que la plantilla creará recursos de IAM.
9. Elija Crear para implementar la pila.

Puede ver el estado de la pila en la CloudFormation consola de AWS en la columna Estado. Debería recibir el estado CREATE_COMPLETE en aproximadamente cinco minutos.

Paso 2 (opcional): Lanzamiento de la pila remota en cuentas secundarias

Important

La pila remota debe implementarse en la misma región que la pila central.

Esta CloudFormation plantilla de AWS automatizada configura los permisos de las cuentas secundarias que permitirán al hub stack programar instancias en otras cuentas. Instale la plantilla remota solo después de que la primary/hub pila se haya instalado correctamente en la cuenta de Hub.

Launch solution

1. Inicie sesión en la consola de administración de AWS de la cuenta secundaria correspondiente y seleccione el botón para lanzar la plantilla de AWS instance-scheduler-on-aws-remote. CloudFormation
2. La plantilla se lanza en la región Este de EE. UU. (Norte de Virginia) de forma predeterminada. Para lanzar la solución en una región de AWS diferente, utilice el selector de regiones de la barra de navegación de la consola. Si la pila central está configurada para usar AWS Organizations, implemente la plantilla remota en la misma región que la pila central.
3. En la página Crear pila, verifique que la dirección URL de la plantilla correcta se encuentre en el cuadro de texto URL de Amazon S3 y elija Siguiente.
4. En la página Especificar detalles, asigne un nombre a la pila remota.
5. En Parámetros, revise el parámetro de la plantilla y modifíquelo.
6. Si la opción de AWS Organizations está habilitada y la pila central está configurada de forma similar, no se requieren más cambios en la pila principal para iniciar la programación.
7. Si la opción de AWS Organization está establecida en No, la pila central debe actualizarse con el nuevo ID de cuenta.

Parámetro	Predeterminado	Description (Descripción)
Infraestructura		
Namespace	default	Identificador único que se usa para diferenciar entre varias implementaciones de la solución. Debe establecerse en el mismo valor que la pila central.
Uso de AWS Organizations	No	Utilice AWS Organizations para automatizar el registro de cuentas radiales. Debe establecerse en el mismo valor que la pila central.
ID de cuenta central	<Requiere que se introduzcan datos>	ID de cuenta de la pila central del programador de instancia s de AWS que programará los recursos en esta cuenta.
Clave de etiqueta de programación	Schedule	La clave de etiqueta que lee la solución a fin de determinar la programación para un recurso. Debe establecerse en el mismo valor que la pila central.
Member-Account Programación		
Regiones	<Optional Input>	Lista de regiones en las que se programarán las instancias. Por ejemplo,us-east-1 ,us-west-1 . (no tiene que ser la misma lista de regiones que el hub). Si deja este parámetro

Parámetro	Predeterminado	Description (Descripción)
		en blanco, la solución utilizará la región actual.
ARN de clave de KMS para EC2	<Optional Input>	Comma-separated lista de ARN de KMS para conceder a la solución CreateGrant permisos kms: para proporcionar al servicio EC2 permisos de descifrado para volúmenes de EBS cifrados. Esto permite al programador iniciar instancias de EC2 con volúmenes de EBS cifrados asociados. Proporcione (*) para conceder acceso limitado a todas las claves de KMS; déjelo en blanco para deshabilitarlo. Para obtener más información, consulte Volúmenes EBS cifrados de EC2 .
RANs de License Manager para EC2	<Optional Input>	Comma-separated lista de ARN de configuración de License Manager para conceder permisos al Programador de Instancias para iniciar instancias EC2 administradas por License Manager. Déjelo en blanco para deshabilitarlo. Para obtener más información, consulte EC2 License Manager .

1. Elija Siguiente.
2. En la página Opciones, seleccione Siguiente.
3. En la página Revisar y crear, revise y confirme la configuración. Asegúrese de marcar la casilla para reconocer que la plantilla creará recursos de IAM.
4. Elija Crear para implementar la pila.

Puede ver el estado de la pila en la CloudFormation consola de AWS en la columna Estado. Debería aparecer el estado CREATE_COMPLETE en aproximadamente cinco minutos.

Configuración de la solución

Ahora que la solución se ha implementado, puede empezar a configurar programaciones y a etiquetar instancias para el programador. Para obtener más información sobre cómo realizar estas operaciones, consulte [Configuración de programaciones](#) y [Etiquetado de instancias para la programación](#).

Guía del operador

Esta guía está destinada a los usuarios y operadores de esta solución y contiene detalles sobre cómo [configurar los horarios](#) y [monitorear la solución](#).

Configuración de programaciones

Important

Los horarios mal configurados pueden provocar que las instancias se ejecuten de forma continua e incurran en costes inesperados. Antes de aplicar los cronogramas a sus recursos, compruebe lo siguiente:

- El nombre del programa en la etiqueta de recursos coincide exactamente con un programa definido en la tabla de configuración. Si los nombres de los programas están mal escritos o no existen, se producirá un `UnknownSchedule` error y el programador no detendrá la instancia. Comprueba la `IS-Error` etiqueta en tus recursos para identificar esta condición.
- Si `stop_new_instances` se establece en `false`, las instancias que se estén ejecutando fuera de un período programado cuando se etiquetaron por primera vez no se detendrán hasta la siguiente transición de parada programada. Esto puede provocar que las instancias se ejecuten durante más tiempo del esperado.
- Si `retain_running` se establece en `true`, las instancias que se inicien manualmente antes de que comience un período de ejecución no se detendrán al final de ese período. Esto se debe a un diseño, pero puede provocar que las instancias se ejecuten indefinidamente si no se monitorizan.
- Cuando se usa `enforced: false` (el valor predeterminado), el programador no reiniciará las instancias que se detengan manualmente durante un período de ejecución ni detendrá las instancias que se inicien manualmente fuera de un período de ejecución después de la transición de parada inicial.

Recomendamos activar el [etiquetado informativo](#) (activado de forma predeterminada) y revisar periódicamente las `IS-LastAction` etiquetas `IS-Error` y de los recursos para confirmar que la programación funciona según lo previsto.

Una vez que la solución se haya implementado correctamente, puede empezar a configurar programaciones. El programador de instancias de AWS admite dos métodos de administración de programaciones, tal y como se describe a continuación:

Note

La solución puede admitir un número cualquiera de programaciones, cada una de las cuales puede contener uno o varios períodos que definan cuándo deben ejecutarse las instancias que controla esa programación. Para obtener más información, consulte [Programaciones](#) y [Períodos](#).

Uso de Infraestructura como código (recomendado)

El programador de instancias de AWS proporciona un AWS CloudFormation CustomResource que puede usar para administrar sus horarios y períodos mediante la infraestructura como código (IaC).

Para obtener información sobre cómo administrar las programaciones mediante IaC, consulte [Administración de programaciones mediante Infraestructura como código \(IaC\)](#).


Uso de la consola de Amazon DynamoDB y del programador de instancias en la CLI de AWS

Important

Si ha utilizado el recurso personalizado para administrar cualquier programación mediante IaC, no debe usar la consola de DynamoDB ni la CLI del programador para eliminar o modificar esas programaciones o sus períodos. Si lo hace, generará un conflicto entre los parámetros almacenados CloudFormation y los valores de la tabla. Además, no utilice períodos gestionados por planificaciones creadas mediante CloudFormation la consola de DynamoDB o la CLI del programador.

Al implementar el programador de instancias en la pila central de AWS, la solución ha creado una tabla de Amazon DynamoDB que contiene varios ejemplos de períodos y programaciones que puede usar como referencia para crear sus propios períodos y programaciones personalizados. Para crear una programación en DynamoDB, modifique una de las programaciones de la tabla de configuración

ConfigTable () o cree una nueva. Para crear una programación mediante la CLI, primero [instale la CLI del programador](#) y, a continuación, utilice los [comandos disponibles](#).

 Note

[Para ver ejemplos de cómo crear varios programas de ejemplo mediante laC, DynamoDB y la InstanceScheduler CLI, consulte Ejemplos de programas.](#)

En esta sección se proporcionan instrucciones y referencias sobre cómo usar, supervisar y actualizar la solución, así como información de solución de problemas y soporte.

Etiquetado de instancias para la programación

Cuando implementó la CloudFormation plantilla de AWS, definió el nombre (clave de etiqueta) de la etiqueta personalizada de la solución. Para que el programador de instancias de AWS reconozca una instancia de Amazon EC2 o Amazon RDS, la clave de etiqueta de esa instancia debe coincidir con esta clave de etiqueta personalizada. Por lo tanto, es importante que aplique las etiquetas de forma coherente y correcta a todas las instancias aplicables. Puede seguir utilizando las [prácticas recomendadas de etiquetado](#) existentes para sus instancias mientras utiliza esta solución. Para obtener más información, consulte [Etiquetar los recursos de Amazon EC2](#) y [Etiquetado de los recursos de Amazon RDS](#).

En la consola de administración de AWS, utilice el [Editor de etiquetas](#) a fin de aplicar o modificar etiquetas para varios recursos a la vez. También puede aplicar y modificar las etiquetas de forma manual en la consola.

Poco después de etiquetar un recurso, el programador de instancias aplicará una IS-ManagedBy etiqueta al recurso para indicar que el programador ahora administra el recurso. Puede buscar esta etiqueta para confirmar que el recurso se ha registrado correctamente para su programación.

Establecimiento del valor de etiqueta

Al aplicar una etiqueta a una instancia, utilice la clave de etiqueta que definió durante la configuración inicial (de forma predeterminada, la clave de etiqueta es Schedule) y establezca el valor de etiqueta en el nombre de la programación que debe aplicarse a la instancia. Si desea cambiar la clave de la etiqueta, [actualice los parámetros de la solución](#).

Note

En el caso de las instancias de Amazon RDS, el valor de etiqueta puede tener una longitud de entre 1 y 256 caracteres Unicode y no puede llevar el prefijo aws: La cadena solo puede incluir el conjunto de letras de Unicode, dígitos y espacio en blanco, “_”, “:”, “/”, “=”, “”, “-” (expresión regular de Java: “`^([\p{L}\p{Z}\p{N}_:/=\-]*)$`”). Para obtener más información, consulte [Etiquetado de los recursos de Amazon RDS](#).

Instancias de EC2 con volúmenes de EBS cifrados

Si sus instancias de base de datos EC2 tienen volúmenes de EBS cifrados con claves KMS administradas por el cliente, debe conceder el KMS:CreateGrant permiso al rol de programador de instancias para poder iniciar esas instancias. Para obtener más información, consulte [Volúmenes de EBS de EC2 cifrados](#).

Instancias EC2 administradas en License Manager

Si sus instancias EC2 se administran en AWS License Manager, debe conceder a la función de programador de instancias los permisos de administrador de licencias adecuados para poder iniciar y detener esas instancias. Para obtener más información, consulte [EC2 License Manager](#).

Referencia de programación

Las programaciones especifican cuándo deben ejecutarse las instancias etiquetadas con esa programación. Cada programación debe tener un nombre único, que se usa como el valor de etiqueta que identifica la programación que desea aplicar al recurso etiquetado.

Periodos

Cada programación debe contener al menos un período que defina las horas a las que debe ejecutarse la instancia. Una programación puede contener más de un período. Cuando se utilice más de un período en una programación, el programador de instancias de AWS aplicará la acción de inicio adecuada cuando al menos uno de los períodos esté establecido en true. Para obtener más información, consulte [Referencia de período](#).

Zona horaria

También puede especificar una zona horaria para la programación. Si no especifica una zona horaria, la programación utilizará la predeterminada que especifique al lanzar la solución. Para obtener una lista de valores de zona horaria aceptables, consulte la columna TZ de [List of TZ database time zones](#).

Campo «Detener nuevas instancias»

El campo `stop_new_instances` controla si el programador de instancias debe detener una instancia la primera vez que se etiqueta para su programación si actualmente se está ejecutando fuera de un período de ejecución. De forma predeterminada, este campo se establece en `true`.

Si se establece en `true`, si etiqueta una instancia en ejecución que está fuera del período de ejecución programado, el Programador de instancias la detendrá inmediatamente. Si se establece en `false`, el Programador de instancias dejará la instancia en ejecución hasta la siguiente hora de parada programada.

Campo Hibernate

El campo `hibernate` permite utilizar la hibernación para las instancias de Amazon EC2 detenidas. Si este campo se establece en `true`, las instancias de EC2 deben usar una imagen de máquina de Amazon (AMI) que admita la hibernación. Para obtener más información, consulte las [AMI de Linux compatibles](#) en la Guía del usuario de Amazon EC2. La hibernación guarda el contenido de la memoria de la instancia (RAM) en su volumen raíz de Amazon Elastic Block Store (Amazon EBS). Si este campo se establece en `true`, las instancias se pondrán en hibernación en lugar de detenerse cuando la solución las detenga.

Si establece la solución para usar esta característica, pero las instancias no están [configuradas para la hibernación](#) o no cumplen los [requisitos previos para la hibernación](#), la solución registra una advertencia y las instancias se detienen sin hibernación. Para obtener más información, consulte [Hibernate your On-Demand instance o spot instance](#) en la Guía del usuario de Amazon EC2.

Campo Enforced

Las programaciones contienen un campo `enforced` que permite impedir que una instancia se inicie de forma manual fuera de un período de ejecución o que se detenga manualmente durante dicho período. Si este campo se establece en `true` y un usuario inicia de forma manual una instancia fuera

de un período de ejecución, la solución detendrá la instancia. Si este campo se establece en true, también reinicia una instancia si se detiene manualmente durante un período de ejecución.

Campo Retain running

El campo `retain_running` impide que la solución detenga una instancia al final de un período de ejecución si la instancia se inició de forma manual antes de comenzar el período. Por ejemplo, si una instancia con un período de ejecución de 9:00 a 17:00 h se inicia manualmente antes de las 9.00 h, la solución no detendrá la instancia a las 17:00 h.

Campo de ventanas de mantenimiento de Systems Manager (aplicable solo a las instancias de EC2)

El campo `ssm-maintenance-window` permite agregar de forma automática ventanas de mantenimiento de AWS Systems Manager como períodos de ejecución a una programación. Al especificar el nombre de un período de mantenimiento que existe en la misma cuenta y región de AWS que sus instancias de Amazon EC2, la solución iniciará la instancia al menos diez minutos antes del inicio de la ventana de mantenimiento y la detendrá al final de dicha ventana si ningún otro período de ejecución especifica que la instancia debe ejecutarse.

Una vez que se haya creado la ventana de mantenimiento de SSM y se haya configurado la programación con el nombre de la ventana de mantenimiento de SSM, los cambios se recogerán en la siguiente ejecución programada de la instancia de Lambda. Por ejemplo, si ha seleccionado una frecuencia de cinco minutos para la ejecución de la función de Lambda del programador, esta detectará los cambios en la ventana de mantenimiento en el siguiente intervalo de cinco minutos.

El programador de instancias de AWS se asegurará de que sus instancias se inicien al menos diez minutos antes de comenzar la ventana de mantenimiento. Según el valor que establezcas para el CloudFormation parámetro Intervalo de programación de AWS, esto puede provocar que la instancia se inicie con más de 10 minutos de intervalo antes del inicio del período de mantenimiento para garantizar que la instancia se inicie al menos 10 minutos antes. Por ejemplo, si establece el intervalo de programación en treinta minutos, el programador iniciará la instancia entre diez y cuarenta minutos antes de comenzar la ventana de mantenimiento.

Note

Para poder utilizar esta función, el CloudFormation parámetro `Enable EC2 SSM Maintenance Windows` de la pila de concentradores de soluciones debe estar establecido en `yes`.

Para obtener más información, consulte [Ventanas de mantenimiento de AWS Systems Manager](#) de la Guía del usuario de AWS Systems Manager.

Tipo de instancia

Solo en el caso de las instancias de Amazon EC2, una programación permite especificar un tipo de instancia deseado opcional para cada uno de los períodos de una programación. Al especificar un tipo de instancia en el período, la solución cambiará de forma automática el tamaño de las instancias de EC2 para que coincidan con el tipo de instancia solicitado.

Para especificar un tipo de instancia, utilice la sintaxis `<nombre-de-período>@<tipo-de-instancia>`. Por ejemplo, `weekends@t2.nano`. Tenga en cuenta que si especifica un tipo de instancia para un período que programa instancias de Amazon EC2 e instancias de Amazon RDS, el tipo de instancia se ignorará para las instancias de Amazon RDS.

Si el tipo de una instancia en ejecución es distinto al tipo de instancia especificado para el período, la solución detendrá la instancia en ejecución y la reiniciará con el tipo de instancia especificado. Para obtener más información, consulte [Cambios de tipos de instancias](#) de la Guía del usuario de Amazon EC2 para instancias de Linux.

Definiciones de programación

La tabla de configuración del programador de instancias de AWS en Amazon DynamoDB contiene definiciones de programación. Una definición de programación puede incluir los siguientes campos:

Campo	Description (Descripción)
<code>description</code>	Descripción opcional de la programación.
<code>hibernate</code>	Elija si desea hibernar aquellas instancias de Amazon EC2 que ejecutan Amazon Linux. Cuando este campo se establezca en <code>true</code> , el programador hibernará las instancias al detenerlas. Tenga en cuenta que sus instancias deben activar la hibernación y cumplir los requisitos previos para la hibernación .
<code>enforced</code>	Elija si desea aplicar la programación. Cuando este campo se establezca en <code>true</code> ,

Campo	Description (Descripción)
	<p>el programador detendrá una instancia en ejecución si se inicia de forma manual fuera del período de ejecución, o bien iniciará una instancia si se detiene manualmente durante dicho período.</p>
name	<p>Nombre que se utiliza para identificar la programación. Este nombre debe ser único y contener únicamente caracteres alfanuméricos, guiones (-) y guiones bajos (_).</p>
periods	<p>Nombre de los períodos que se utilizan en esta programación. Introduzca los nombres exactamente como aparecen en el campo de nombre de período.</p> <p>También puede especificar un tipo de instancia para el período mediante la sintaxis <nombre-de-e-período>@<tipo-de-instancia>. Por ejemplo, <code>weekdays@t2.large</code> .</p>
retain_running	<p>Elija si desea impedir que la solución detenga una instancia al final de un período de ejecución si la instancia se inició manualmente antes de comenzar el período.</p>
use_maintenance_window	<p>Elija si desea incluir la ventana de mantenimiento de Amazon RDS como período de ejecución en una programación de instancias de Amazon RDS, o bien una ventana de mantenimiento de AWS Systems Manager como período de ejecución en una programación de instancias de Amazon EC2. Este campo está habilitado de forma predeterminada y, para deshabilitarlo, establezca su valor en "false".</p>

Campo	Description (Descripción)
<code>ssm_maintenance_window</code>	<p>Elija si desea agregar ventanas de mantenimiento de AWS Systems Manager como período de ejecución adicional para esta programación. Acepta StringSet los nombres de las ventanas de mantenimiento que se compararán con los nombres de las ventanas de las account/region mismas instancias de EC2 programadas.</p> <p>Nota: Esta característica solo se aplica a las instancias de EC2.</p>
<code>stop_new_instances</code>	<p>Elija si desea detener una instancia la primera vez que se etiquete si se ejecuta fuera del período de ejecución. De forma predeterminada, este campo se establece en true.</p>
<code>timezone</code>	<p>Zona horaria que la programación utilizará. Si no se especifica ninguna zona horaria, se utiliza la zona horaria predeterminada (UTC). Para obtener una lista de valores de zona horaria aceptables, consulte la columna TZ de List of TZ database time zones.</p>
<code>use_metrics</code>	<p>Elija si desea activar las CloudWatch métricas a nivel de programación. Este campo sobrescribe la configuración de CloudWatch métricas que especificó en el despliegue.</p> <p>Nota: Al habilitar esta función, se incurrirá en cargos de 0\$. 90/month por horario o servicio programado.</p>

Referencia de período

Los períodos contienen condiciones que permiten establecer las horas, los días y los meses específicos en los que se ejecutará una instancia. Un período puede contener varias condiciones, pero todas ellas deben cumplirse para que el programador de instancias de AWS aplique la acción de inicio o detención adecuada.

Horas de inicio y detención

Los campos `begintime` y `endtime` definen cuándo iniciará y detendrá las instancias el programador de instancias de AWS. Si solo especifica una hora de inicio, la instancia debe detenerse de forma manual. Tenga en cuenta que si especifica un valor en el campo [weekdays](#), la solución utiliza ese valor para determinar cuándo detener la instancia. Por ejemplo, si especifica las 9:00 h como `begintime` sin `endtime` y un valor de días laborables de lunes a viernes, la instancia se detendrá el viernes a las 23:59 h, a menos que haya programado un período adyacente.

De forma similar, si solo especifica una hora de detención, la instancia debe iniciarse manualmente. Si no especifica ninguna hora, esta solución utiliza las reglas de los días de la semana, los días del mes o los meses para iniciar y detener `beginning/end` las instancias cada día, según corresponda.

Los valores `begintime` y `endtime` de su período deben estar en la zona horaria especificada en la programación. Si no especifica una zona horaria en la programación, la solución utilizará la especificada al lanzar la solución.

Si su programación contiene varios períodos, recomendamos que especifique siempre tanto un valor `begintime` como `endtime` en sus períodos.

Si inicia una instancia antes de la hora de inicio especificada, la instancia se ejecutará hasta el final del período de ejecución. Por ejemplo, un usuario puede definir un período en el que se inicie una instancia todos los días a las 9:00 h y se detenga a las 17:00 h.

Inicio y detención programados de 9:00 h a 17:00 h



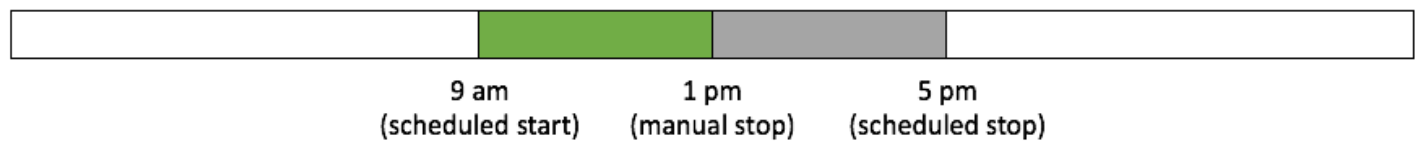
Si inicia esa instancia de forma manual a las 5:00 h, la solución detendrá la instancia a las 17:00 h. Si utiliza el [campo `retain running`](#), la solución no detendrá la instancia a las 17:00 h.

Detención programada a las 5:00 h



Si detiene una instancia antes de la hora de detención especificada, la instancia no se ejecutará hasta el comienzo del siguiente período de ejecución. Prosigamos con el ejemplo anterior: si el usuario detiene la instancia el miércoles a las 13:00 h, la solución no iniciará la instancia hasta el jueves a las 9:00 h.

Cronología que muestra el inicio programado a las 9:00 h., la detención manual a las 13:00 h y la detención programada a las 17:00 h.



Períodos adyacentes

La solución no detendrá las instancias en ejecución si la programación contiene dos períodos de ejecución adyacentes. Por ejemplo, si tiene una programación con un período cuyo valor `endtime` son las 23:59 h y otro período cuyo valor `begintime` es la medianoche del día siguiente, la solución no detendrá las instancias en ejecución si no hay reglas de `weekdays`, `monthdays` o `months` que detengan las instancias.

Para implementar una programación que ejecute instancias de lunes a las 9:00 h a viernes a las 17:00 h, la solución requiere tres períodos. El primer período ejecuta las instancias aplicables el lunes de 9:00 h a 23:59 h. El segundo período ejecuta las instancias de martes a medianoche a jueves a las 23:59 h. El tercer período ejecuta las instancias el viernes de medianoche a 17:00 h. Para obtener más información, consulte [Programaciones de ejemplo](#).

Días de la semana

El campo `weekdays` define los días de la semana en los que se ejecutará una instancia. Puede especificar una lista de días, un intervalo de días, la aparición ^X de ese día en un mes o su última aparición en un mes. La solución admite nombres de días abreviados (lun.) y números (0).

Días del mes

El campo `monthdays` define los días del mes durante los que se ejecutará una instancia. Puede especificar una lista de días, un intervalo de días, cada día X del mes, el último día del mes o el día laborable más próximo a una fecha específica.

Meses

El campo `months` define los meses en los que se ejecutará una instancia. Puede especificar una lista de meses, un intervalo de meses o cada mes X. La solución admite nombres de meses abreviados (en.) y números (1).

Definiciones de período

La tabla de configuración del programador de instancias de AWS de Amazon DynamoDB contiene definiciones de período. Una definición de período puede incluir los campos siguientes. Tenga en cuenta que algunos campos admiten [caracteres cron no estándar](#).

Important

Debe especificar al menos uno de los siguientes elementos: `begintime`, `endtime`, `weekdays`, `months` o `monthdays`.

Campo	Description (Descripción)
<code>begintime</code>	La hora, en HH:MMformato, a la que se iniciará la instancia.
<code>description</code>	Descripción opcional del período.
<code>endtime</code>	La hora, en HH:MMformato, en la que se detendrá la instancia.
<code>months</code>	Especifique una lista delimitada por comas de meses, o un intervalo de meses separados por guiones, durante los cuales se ejecutará la instancia. Por ejemplo, especifique <code>jan, feb, mar</code> o <code>1, 2, 3</code> para ejecutar una instancia

Campo	Description (Descripción)
	<p>durante esos meses. O bien, puede especificar <code>jan-mar</code> o <code>1-3</code>.</p> <p>También puede programar una instancia para que se ejecute cada mes X o cada mes X de un intervalo. Por ejemplo, especifique <code>Jan/3</code> o <code>1/3</code> para ejecutar una instancia cada tres meses a partir de enero. Especifique <code>Jan-Jul/2</code> para que la ejecución se realice una vez cada dos meses, de enero a julio.</p>
<p><code>monthdays</code></p>	<p>Especifique una lista delimitada por comas de los días del mes, o un intervalo de días separados por guiones, durante los cuales se ejecutará la instancia. Por ejemplo, especifique <code>1, 2, 3</code> o <code>1-3</code> para ejecutar una instancia durante los tres primeros días del mes.</p> <p>También puede especificar varios intervalos. Por ejemplo, especifique <code>1-3, 7-9</code> para ejecutar una instancia del 1 al 3 y del 7 al 9.</p> <p>También puede programar una instancia para que se ejecute cada día X del mes o cada día X del mes de un intervalo. Por ejemplo, especifique <code>1/7</code> para ejecutar una instancia cada siete días a partir del 1. Especifique <code>1-15/2</code> para ejecutar una instancia cada dos días del 1 al 15.</p> <p>Especifique <code>L</code> para ejecutar una instancia el último día del mes. Especifique una fecha y <code>W</code> para ejecutar una instancia el día laborable más próximo a la fecha especificada. Por ejemplo, especifique <code>15W</code> para ejecutar una instancia el día laborable más próximo al 15.</p>

Campo	Description (Descripción)
name	Nombre que se utiliza para identificar el período. Este nombre debe ser único y contener únicamente caracteres alfanuméricos, guiones (-) y guiones bajos (_).
weekdays	<p>Especifique una lista delimitada por comas de días de la semana, o un intervalo de días de la semana, durante los cuales se ejecutará la instancia. Por ejemplo, especifique <code>0, 1, 2</code> o <code>0-2</code> para ejecutar una instancia de lunes a miércoles. También puede especificar varios intervalos. Por ejemplo, especifique <code>0-2, 4-6</code> para ejecutar una instancia todos los días excepto los jueves.</p> <p>También puede programar una instancia para que se ejecute cada aparición X de un día laborable del mes. Por ejemplo, especifique <code>Mon#1</code> o <code>0#1</code> para ejecutar una instancia el primer lunes del mes.</p> <p>Especifique un día y L para ejecutar una instancia en la última aparición de ese día laborable del mes. Por ejemplo, especifique <code>friL</code> o <code>4L</code> para ejecutar una instancia el último viernes del mes.</p>

Cuando un período contiene varias condiciones, tenga en cuenta que todas ellas deben cumplirse para que el programador de instancias de AWS aplique la acción adecuada. Por ejemplo, un período que contiene un campo `weekdays` con un valor de `Mon#1` y un campo `months` con un valor de `Jan/3` aplicarán la acción el primer lunes del trimestre.

Programaciones de ejemplo

El programador de instancias de AWS permite iniciar y detener instancias de Amazon Elastic Compute Cloud (Amazon EC2) y de Amazon Relational Database Service (Amazon RDS) de forma automática. En la sección siguiente se proporcionan algunas programaciones de ejemplo que se pueden adaptar a muchos casos de uso habituales.

Horas de trabajo estándar de 9:00 h a 17:00 h

Esta programación muestra cómo ejecutar instancias los días laborables de 9:00 h a 17:00 h en Londres.

Periodos

Este período iniciará las instancias a las 9 a.m. y las detendrá a las 5 p.m., de lunes a viernes (Mon-Fri).

Campo	Tipo	Valor
begintime	String	09:00
endtime	String	16:59
name	String	weekdays-9-5
weekdays	StringSet	mon-fri

Schedule

El nombre de la programación proporciona el valor de etiqueta que se debe aplicar a las instancias y la zona horaria que se utilizará.

Campo	Tipo	Valor
name	String	london-working-hours
periods	StringSet	weekdays-9-5
timezone	String	Europe/London

Etiqueta de instancia

Para aplicar esta programación a las instancias, debe agregar la etiqueta `Schedule=london-working-hours` a estas. Si cambia el nombre de etiqueta predeterminado en el parámetro de nombre de etiqueta del Programador de CloudFormation instancias de AWS, su etiqueta será diferente. Por ejemplo, si ha especificado `Sked` como su nombre de etiqueta, esta será `Sked=london-working-hours`. Para obtener más información, consulte [Etiquetar los recursos](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

CLI del programador

Para configurar la programación anterior mediante la [CLI del programador de instancias](#), utilice los siguientes comandos:

```
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
```

Recurso personalizado

La siguiente CloudFormation plantilla creará el cronograma anterior utilizando el [recurso personalizado del cronograma](#).

Para implementar esta plantilla, tendrá que proporcionar el `ServiceInstanceScheduleServiceTokenARN` que se encuentra en la CloudFormation consola de AWS. Para ello, seleccione la [pila de concentradores de programadores de instancias implementada anteriormente](#) y, a continuación, seleccione Salidas.

```
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

```

Detención de instancias después de las 17:00 h

Las instancias se pueden iniciar libremente en cualquier momento del día y esta programación garantizará que se les envíe un comando de detención de forma automática todos los días a las 17:00 h (ET).

Periodos

Este período detendrá las instancias todos los días a las 17:00 h.

Campo	Tipo	Valor
endtime	String	16:59
name	String	stop-at-5

Schedule

El nombre de la programación proporciona el valor de etiqueta que se debe aplicar a las instancias y la zona horaria que se utilizará.

Campo	Tipo	Valor
name	String	stop-at-5-new-york

Campo		Valor
periods	StringSet	stop-at-5
timezone	String	America/New York

Etiqueta de instancia

Para aplicar esta programación a las instancias, debe agregar la etiqueta `Schedule=stop-at-5-new-york` a estas. Si ha cambiado el nombre de etiqueta predeterminado en el parámetro de nombre de etiqueta del Programador de CloudFormation instancias de AWS, su etiqueta será diferente. Por ejemplo, si ha especificado `Sked` como su nombre de etiqueta, esta será `Sked=stop-at-5-new-york`. Para obtener más información, consulte [Etiquetar los recursos](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

CLI del programador

Para configurar la programación anterior mediante la [CLI del programador de instancias](#), utilice los siguientes comandos:

```
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
```

Recurso personalizado

La siguiente CloudFormation plantilla creará el cronograma anterior utilizando el [recurso personalizado del cronograma](#).

Para implementar esta plantilla, tendrá que proporcionar el `ServiceInstanceScheduleServiceTokenARN` que se encuentra en la CloudFormation consola de AWS. Para ello, haga clic en la [pila de concentradores del programador de instancias implementada anteriormente](#) y seleccione Salidas.

```
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'

```

Detención de instancias durante el fin de semana

Esta programación muestra cómo ejecutar instancias de lunes a las 9:00 h (ET) a viernes a las 17:00 h (ET). Como los lunes y viernes no son días completos, este programa incluye tres períodos para adaptarlos: lunes y viernes. Tuesday-Thursday

Periodos

El primer período inicia las instancias etiquetadas el lunes a las 9:00 h y las detiene a medianoche. Este período incluye los siguientes campos y valores:

Campo	Tipo	Valor
begintime	String	09:00
endtime	String	23:59
name	String	mon-start-9am
weekdays	StringSet	mon

El segundo período ejecuta las instancias etiquetadas todo el día de martes a jueves. Este período incluye los siguientes campos y valores:

Campo		Valor
name	String	tue-thu-full-day
weekdays	StringSet	tue-thu

El tercer período detiene las instancias etiquetadas el viernes a las 17:00 h. Este período incluye los siguientes campos y valores:

Campo		Valor
begintime	String	00:00
endtime	String	16:59
name	String	fri-stop-5pm
weekdays	StringSet	fri

Schedule

La programación combina los tres períodos en la programación para instancias etiquetadas. La programación incluye los siguientes campos y valores:

Campo		Valor
name	String	mon-9am-fri-5pm
periods	StringSet	mon-start-9am,tue-thu-full-day,fri-stop-5pm
timezone	String	America/New_York

Etiqueta de instancia

Para aplicar esta programación a las instancias, debe agregar la etiqueta `Schedule=mon-9am-fri-5pm` a estas. Tenga en cuenta que si ha cambiado el nombre de etiqueta predeterminado en

el parámetro de nombre de etiqueta del Programador de CloudFormation instancias de AWS, su etiqueta será diferente. Por ejemplo, si ha especificado Sked como su nombre de etiqueta, esta será Sked=mon-9am-fri-5pm. Para obtener más información, consulte [Etiquetar los recursos](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

CLI del programador

Para configurar la programación anterior mediante la [CLI del programador de instancias](#), utilice los siguientes comandos:

```
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
```

Recurso personalizado

La siguiente CloudFormation plantilla creará el cronograma anterior utilizando el [recurso personalizado del cronograma](#).

Para implementar esta plantilla, tendrá que proporcionar el ServiceInstanceScheduleServiceTokenARN que se encuentra en la CloudFormation consola de AWS. Para ello, seleccione la [pila de concentradores de programadores de instancias implementada anteriormente](#) y, a continuación, seleccione Salidas.

```
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

Recursos para soluciones

Los siguientes recursos se crean como parte de la pila del programador de instancias de AWS.

Nombre del recurso	Tipo	Description (Descripción)
Principal	AWS::Lambda::Function	Función de Lambda del programador de instancias de AWS.
Ayudante de configuración del programador	Custom::ServiceSetup	Almacena valores de configuración global en Amazon DynamoDB.
Permiso de invocación del programador	AWS::Lambda::Permission	Permite que el CloudWatch evento de Amazon invoque la función AWS Lambda del programador de instancias.
Registros del programador	AWS::Logs::LogGroup	CloudWatch Grupo de registros para el programador de instancias.

Nombre del recurso	Tipo	Description (Descripción)
Política del programador	<code>AWS::IAM::Policy</code>	Política que permite al programador realizar acciones de inicio y detención, cambiar atributos de instancia de Amazon EC2, establecer etiquetas y acceder a recursos del programador.
Regla del programador	<code>AWS::Events::Rule</code>	Regla de EventBridge eventos de Amazon que invoca la función Lambda del programador.
Regla de eventos de métricas de configuración	<code>AWS::Events::Rule</code>	Regla de EventBridge eventos de Amazon que invoca periódicamente la función de métricas anónimas de la descripción de la configuración. Se deshabilita cuando las métricas anonimizadas están deshabilitadas.
Tabla de estados	<code>AWS::DynamoDB::Table</code>	Tabla de DynamoDB que almacena el último estado deseado de las instancias.
Tabla de configuración	<code>AWS::DynamoDB::Table</code>	Tabla de DynamoDB que almacena datos globales en relación con la configuración, las programaciones y los períodos.
Tema de SNS del programador de instancias	<code>AWS::SNS::Topic</code>	Envía mensajes de advertencia y error a las direcciones de correo electrónico suscritas.

CLI del programador

La interfaz de la línea de comandos (CLI) del programador de instancias de AWS permite configurar programaciones y períodos, así como estimar el ahorro de costos para una programación determinada.

Requisitos previos

La CLI de esta solución requiere Python 3.8 o versiones posteriores y la última versión de boto3.

Credenciales

Para usar la CLI del programador, debe tener credenciales para la CLI de AWS. Para obtener más información, consulte [Opciones de los archivos de configuración y credenciales](#) de la Guía del usuario de AWS CLI.

Sus credenciales deben tener los siguientes permisos:

- `lambda:InvokeFunction`- Para invocar la `InstanceSchedulerMain` función en la pila de planificadores y actualizar la información de programación y período en la base de datos de configuración del programador desde la línea de comandos
- `cloudformation:DescribeStackResource`: para recuperar el ID de recurso físico de la función de AWS Lambda de la pila a fin de controlar la solicitud de la CLI

Las solicitudes realizadas por la CLI del programador y las respuestas se registran en el flujo de registro `AdminCliRequestHandler-yyyyymmdd`.

Note

Si especifica un perfil mediante el argumento `profile-name`, el perfil que especifique debe tener estos permisos. Para obtener más información sobre el argumento `profile-name`, consulte [Argumentos comunes](#).

Instalación de la CLI del programador

1. [Descargue](#) el paquete de la CLI del programador (`instance_scheduler_cli.zip`) y colóquelo en un directorio de su computadora.

⚠ Important

Si no coloca los archivos en su propio directorio y, después, los instala desde este, se producirá un error en la instalación.

2. Descomprima el archivo zip en su propio directorio (`instance_scheduler_cli`).
3. En el mismo directorio en el que colocó el paquete de la CLI descomprimido, instale `scheduler-cli` en su entorno:

📘 Note

Scheduler-CLI requiere Python 3.8 o superior y las últimas versiones de pip y boto3. Si no las tiene todas instaladas en su máquina local, consulte las instrucciones de instalación en la [documentación oficial de pip](#) antes de intentar instalarlas. Scheduler-CLI

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

4. Compruebe que la instalación se ha realizado correctamente con:

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

📘 Note

Si lo prefiere, se puede instalar un [archivo de distribución de código fuente \(sdist\) de la CLI](#) mediante el mismo proceso descrito anteriormente.

Estructura de comandos

La CLI del programador usa una estructura multiparte en la línea de comandos. En la parte siguiente se especifica el script de Python de la CLI del programador. La CLI del programador tiene comandos que especifican las operaciones que se deben realizar en los períodos y las programaciones. Los argumentos específicos de una operación se pueden especificar en la línea de comandos en cualquier orden.

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

Argumentos comunes

La CLI del programador admite los siguientes argumentos que pueden usar todos los comandos:

Argumento	Description (Descripción)
<code>--stack [replaceable]<stackname></code>	<p>Nombre de la pila del programador.</p> <p>Importante: Este argumento es obligatorio para todos los comandos.</p>
<code>--region [replaceable]<regionname></code>	<p>Nombre de la región en la que se implementa la pila del programador.</p> <p>Nota: Debe usar este argumento cuando los archivos de configuración y credenciales predeterminados no estén instalados en la misma región que la pila de soluciones.</p>
<code>--profile-name [replaceable] <profilename></code>	<p>Nombre del perfil que se utilizará para ejecutar comandos. Si no se especifica ningún nombre de perfil, se utiliza el perfil predeterminado.</p>
<code>--query</code>	<p>Expresión JMESPath que controla la salida de comandos. Para obtener más información sobre cómo controlar la salida, consulte Control de la salida de comandos en la CLI de AWS de la Guía del usuario de AWS CLI.</p>
<code>--help</code>	<p>Muestra comandos y argumentos válidos para la CLI del programador. Cuando se usa con un comando específico, muestra subcomandos y argumentos válidos para ese comando.</p>
<code>--version</code>	<p>Muestra el número de versión de la CLI del programador.</p>

Comandos disponibles

- [create-period](#)
- [create-schedule](#)
- [delete-period](#)
- [delete-schedule](#)
- [describe-periods](#)
- [describe-schedules](#)
- [describe-schedule-usage](#)
- [update-period](#)
- [update-schedule](#)
- [help](#)

create-period

Description (Descripción)

Crea un período. Un período debe contener al menos uno de los siguientes elementos: `begintime`, `endtime`, `weekdays`, `months` o `monthdays`.

Argumentos

`--name`

- Nombre del período

Tipo: cadena

Obligatorio: sí

`--description`

- Descripción del período

Tipo: cadena

Requerido: no

--begintime

- Hora de inicio del período de ejecución. Si no se especifican `begintime` y `endtime`, el período de ejecución es de 0:00 h a 23:59 h.

Tipo: cadena

Restricciones: formato H:MM o HH:MM

Obligatorio: no

--endtime

- Hora de detención del período de ejecución. Si no se especifican `begintime` y `endtime`, el período de ejecución es de 0:00 h a 23:59 h.

Tipo: cadena

Restricciones: formato H:MM o HH:MM

Obligatorio: no

--weekdays

- Días de la semana correspondientes al período

Tipo: cadena

Restricciones: Comma-delimited lista de nombres abreviados de días (`mon`) o números (`0`). Utilice `-` para especificar un intervalo. Utilice `/` para especificar cada día X de la semana.

Obligatorio: no

--months

- Meses del período

Tipo: cadena

Restricciones: Comma-delimited lista de nombres de meses abreviados (`jan`) o números (`1`). Utilice `-` para especificar un intervalo. Utilice `/` para especificar cada mes X.

Obligatorio: no

--monthdays

- Días del mes correspondientes al período

Tipo: cadena

Restricciones: Comma-delimited lista de nombres de meses abreviados (ene) o números (1).
Utilice - para especificar un intervalo. Utilice / para especificar cada día X del mes.

Obligatorio: no

Ejemplo

```
$ 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"  
    ]  
  }  
}
```

create-schedule

Description (Descripción)

Crea una programación.

Argumentos

--name

- Nombre de la programación

Tipo: cadena

Obligatorio: sí

--description

- Descripción de la programación

Tipo: cadena

Requerido: no

`--enforced`

- Aplica el estado programado para la instancia

Obligatorio: no

`--use-metrics`

- Recopila CloudWatch las métricas de Amazon

Obligatorio: no

`--periods`

- Lista de períodos de ejecución de la programación. Si se especifican varios períodos, la solución iniciará una instancia si uno de los períodos se evalúa como `true`.

Tipo: cadena

Restricciones: Comma-delimited lista de períodos. Utilice `<period-name>@[replaceable]<instance type>` a fin de especificar un tipo de instancia para un período. Por ejemplo, `weekdays@t2.large`.

Obligatorio: sí

`--retain-running`

- Impide que la solución detenga una instancia al final de un período de ejecución si la instancia se inició de forma manual antes de comenzar el período.

Obligatorio: no

`--ssm-maintenance-window`

- Agrega una ventana de mantenimiento de AWS Systems Manager como período de ejecución a una programación de instancias de Amazon EC2.

Tipo: cadena

Requerido: no

--do-not-stop-new-instances

- No detenga una instancia la primera vez que se etiqüete si se ejecuta fuera de un período de ejecución

Obligatorio: no

--timezone

- Zona horaria que la programación utilizará

Tipo: matriz de cadenas

Obligatorio: No (si no se usa este argumento, se utilizará la zona horaria predeterminada de la pila de soluciones principal).

--use-maintenance-window

- Agrega una ventana de mantenimiento de Amazon RDS como período de ejecución a una programación de instancias de Amazon RDS, o bien una ventana de mantenimiento de AWS Systems Manager como período de ejecución a una programación de instancias de Amazon EC2.

Tipo: true/false

Obligatorio: No (el valor predeterminado es true)

Ejemplo

```
$ 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"
  }
}
```

delete-period

--name

- Nombre del período aplicable

Tipo: cadena

Obligatorio: sí

Important

Si el período se utiliza en las programaciones existentes, debe eliminarlo de dichas programaciones antes de eliminarlo.

Ejemplo

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

delete-schedule

Description (Descripción)

Elimina una programación existente

Argumentos

--name

- Nombre de la programación aplicable

Tipo: cadena

Obligatorio: sí

Ejemplo

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

describe-periods

Description (Descripción)

Enumera los períodos configurados para la pila del programador de instancias

Argumentos

`--name`

- Nombre de un período específico que desea que se describa

Tipo: cadena

Requerido: no

Ejemplo

```
$ 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"
}
]
}

```

describe-schedules

Description (Descripción)

Enumera las programaciones configuradas para la pila del programador de instancias.

Argumentos

--name

- Nombre de una programación específica que desea que se describa

Tipo: cadena

Requerido: no

Ejemplo

```
$ 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 (Descripción)

Enumera todos los períodos que se ejecutan en una programación y calcula las horas de facturación correspondientes a las instancias. Utilice este comando a fin de simular una programación para calcular los posibles ahorros y los períodos de ejecución después de crear o actualizar una programación.

Argumentos

--name

- Nombre de la programación aplicable

Tipo: cadena

Obligatorio: sí

--startdate

- Fecha de inicio del período utilizado para el cálculo. La fecha predeterminada es la fecha actual.

Tipo: cadena

Requerido: no

--enddate

- Fecha de finalización del período utilizado para el cálculo. La fecha predeterminada es la fecha actual.

Tipo: cadena

Requerido: no

Ejemplo

```
$ 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"
```

update-period

Description (Descripción)

Actualiza un período existente

Argumentos

El comando `update-period` admite los mismos argumentos que el comando `create-period`. Para obtener más información sobre los argumentos, consulte el [comando de creación de períodos](#).

Important

Si no especifica un argumento, ese argumento se eliminará del período.

update-schedule

Description (Descripción)

Actualiza una programación existente

Argumentos

El comando `update-schedule` admite los mismos argumentos que el comando `create-schedule`. Para obtener más información sobre los argumentos, consulte el [comando de creación de programaciones](#).

⚠ Important

Si no especifica un argumento, ese argumento se eliminará de la programación.

help

Description (Descripción)

Muestra una lista de comandos y argumentos válidos para la CLI del programador.

Ejemplo

```
$ 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
```

Cuando se usa con un comando específico, el argumento `--help` muestra subcomandos y argumentos válidos para a ese comando.

Ejemplo de comando específico

```
$ 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
```

Actualización de valores de configuración global

Cuando implementó por primera vez la plantilla Instance Scheduler's Hub en AWS CloudFormation, se seleccionaron varios ajustes de configuración global como entradas de parámetros. Estos parámetros de configuración global se pueden actualizar en cualquier momento desde la CloudFormation consola.

Para actualizar la configuración global de Instance Scheduler, inicie sesión en la implementación account/region que contiene su hub y vaya a la CloudFormation consola de AWS. Busque la pila central del programador de instancias y seleccione **Actualizar** → **Usar plantilla existente** . Actualice los parámetros de configuración global que desee cambiar y, a continuación, seleccione **siguiente** → **siguiente** → **enviar** para CloudFormation actualizar los recursos de la solución pertinentes.

Administración de programaciones mediante Infraestructura como código (IaC)

Important

Implemente programaciones mediante una plantilla independiente una vez que la implementación de la pila central se haya completado.

El programador de instancias en AWS proporciona un recurso personalizado (`ServiceInstanceSchedule`) que puede usar para configurar y administrar los horarios a través de AWS CloudFormation. El recurso personalizado utiliza PascalCase claves para los mismos datos que la tabla de configuración del programador de instancias de Amazon DynamoDB (consulte la plantilla siguiente para ver ejemplos). Para obtener más información sobre los campos correspondientes a las programaciones, consulte [Definiciones de programación](#). Para obtener más información sobre los campos correspondientes a los períodos, consulte [Definiciones de período](#).

Cuando utiliza el recurso personalizado para crear una programación, el nombre de esta es el nombre de recurso lógico del recurso personalizado de forma predeterminada. Para especificar un nombre diferente, utilice la propiedad `Name` del recurso personalizado. La solución también agrega el nombre de la pila al nombre de la programación como prefijo de forma predeterminada. Si no desea agregar el nombre de la pila como prefijo, utilice la propiedad `NoStackPrefix`.

Cuando utilice las propiedades `Name` y `NoStackPrefix`, asegúrese de elegir nombres de programación únicos. Si ya existe una programación con el mismo nombre, el recurso no se creará ni se actualizará.

Para empezar a administrar programaciones mediante IaC, copie y pegue la siguiente plantilla de ejemplo y personalice tantas programaciones como desee. Guarde el archivo como un `archivo.template` (por ejemplo: `my-schedules.template`) y, a continuación, implemente la nueva plantilla con AWS CloudFormation. Para ver ejemplos de plantillas de programaciones completadas, consulte [Programaciones de ejemplo](#).

```
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'

```

Al implementar la plantilla, debe proporcionar el ServiceToken ARN para la implementación de Instance Scheduler en AWS. Para encontrar este ARN, navegue hasta la pila CloudFormation de programadores de instancias implementada, seleccione Salidas y busque ServiceInstanceScheduleServiceToken.

Important

No utilice la consola de DynamoDB ni la CLI del programador para eliminar o modificar las programaciones y los períodos que se configuraron mediante el recurso personalizado. Si lo hace, creará un conflicto entre los parámetros almacenados de la pila y los valores

de la tabla. Además, no utilice los períodos configurados con el recurso personalizado en programaciones creadas mediante la consola de DynamoDB o la CLI del programador.

Antes de eliminar la pila del programador de instancias principal, debe eliminar todas las pilas adicionales que contengan programaciones y períodos creados con el recurso personalizado, ya que las pilas de recursos personalizadas contienen dependencias de la tabla de DynamoDB de la pila principal.

En la tabla de DynamoDB de configuración, las programaciones y los períodos que se configuraron con el recurso personalizado se pueden identificar mediante el atributo `configured_in_stack`. El atributo contiene el nombre de recurso de Amazon de la pila que se utilizó para crear el elemento.

Cómo gestionar los errores de capacidad insuficiente de EC2

Cuando el programador de instancias no puede iniciar una instancia debido a una capacidad insuficiente, su comportamiento predeterminado es emitir un evento de inicio fallido (consulte [EventBridge Eventos](#)) e intentarlo de nuevo en el siguiente intervalo de programación. Como alternativa, el programador de instancias se puede configurar para cambiar el tamaño de la instancia a tipos de instancia alternativos antes de volver a intentar la operación de inicio. Esta función ayuda a mejorar la disponibilidad de las instancias en entornos con capacidad limitada.

Configuración

Para habilitar tipos de instancias alternativos para una instancia EC2, añada la `IS-PreferredInstanceTypes` etiqueta a la instancia con una lista de tipos de instancias separados por comas en orden de preferencia (primero el más preferido):

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

Funcionamiento

La lista de tipos de instancias alternativos se proporciona en orden de preferencia, siendo el primer tipo el más preferido. Cuando el programador de instancias intenta iniciar una instancia EC2:

1. Si la instancia no tiene el tamaño preferido actualmente, intenta cambiarlo al tamaño preferido antes de empezar

2. Si la operación de inicio se realiza correctamente, no se intentarán realizar más alternativas
3. Si la operación de inicio falla debido a una capacidad insuficiente:
 - a. Intenta cambiar el tamaño al siguiente tipo de instancia alternativo de la lista
 - b. Vuelve a intentar la operación de inicio
 - c. Si sigue sin éxito, intente con el siguiente tipo alternativo
 - d. Continúa hasta que tenga éxito o se agoten todas las alternativas

Requisitos y limitaciones

Compatibilidad de instancias: los tipos de instancias alternativos deben ser compatibles con la configuración actual de la instancia (AMI, subred, grupos de seguridad, etc.). Para obtener más información, consulte [Cambiar el tipo de instancia](#) en la Guía del usuario de Amazon EC2.

Formato de etiqueta: el valor de la `IS-PreferredInstanceTypes` etiqueta debe ser una lista separada por comas de tipos de instancias EC2 válidos.

Ejemplo

Para una instancia configurada originalmente como `t3.small`, puede configurar:

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

Si la `t3.small` instancia no se inicia debido a problemas de capacidad, el Programador de Instancias intentará cambiar el tamaño e iniciar la instancia a partir `t3.medium` de ese momento `m5.large` hasta que se `t3.large` ejecute correctamente o se agoten todas las opciones.

Programación de grupos de EC2 Auto Scaling

El programador de instancias de AWS admite la programación de grupos de escalado automático (ASG) de EC2 Auto Scaling mediante acciones de escalado programadas. Esto difiere de la implementación de la EC2/RDS programación y se explicará con más detalle en esta sección

Consulte [Escalado programado para Amazon EC2 Auto Scaling](#) a fin de obtener más información sobre las acciones de escalado programadas.

Información general de la programación de ASG

Los ASG se pueden programar mediante la aplicación de una etiqueta de programación, tal y como se describe en [Etiquetado de instancias para la programación](#).

Definición de Running/Stopped para los ASG

Al configurar un grupo de escalado automático, un usuario especifica una capacidad mínima, deseada y máxima para ese ASG. El programador de instancias denomina estos valores al mínimo-máximo deseado o MDM de un ASG.

El estado de ejecución de un ASG se define mediante la etiqueta de control. `IS-MinDesiredMax` Esta etiqueta debe contener los valores de MDM deseados en el formato `min,desired,max` (por ejemplo: `1,3,5`).

Si la `IS-MinDesiredMax` etiqueta no se proporciona cuando se etiqueta el ASG por primera vez para su programación, se generará automáticamente a partir del tamaño actual del ASG en el momento del etiquetado.

Para todos los ASG, el estado de parada se define como un MDM de `0-0-0`.

Comportamiento del ASG Start/Stop

Cuando el Programador de Instancias inicia o detiene un ASG, modifica la configuración de capacidad del ASG:

Iniciar un ASG: establece la capacidad mínima, deseada y máxima en los valores definidos en la `IS-MinDesiredMax` etiqueta (o en los valores generados automáticamente desde que se etiquetó el ASG por primera vez).

Detener un ASG: establece la capacidad mínima, deseada y máxima en `0-0-0`, lo que provoca la finalización de todas las instancias del ASG.

Limitaciones

La programación del ASG se realiza mediante la conversión de programaciones del programador de instancias de AWS en reglas de escalado programadas compatibles con el servicio de ASG. Esta traducción funciona mejor para programaciones sencillas de un solo período que no utilizan expresiones cron complejas.

Las siguientes características de programación no son compatibles con la programación de ASG:

- Marcas de programación avanzadas, como `enforced` y `retain running`.
- N-th Expresiones del día de la semana, el día de la semana más cercano y el último día de la semana en puntos.
- Multi-period programas con períodos inmediatamente adyacentes o superpuestos.
 - Al configurar las acciones de escalado programadas para programas de varios períodos, el programador de instancias `beginning/end` de AWS convierte directamente los períodos en `start/stop` acciones para el ASG, incluso cuando otro período superpuesto o adyacente normalmente provocaría que se omita esa acción.

Supervisión de la solución

Registro y notificaciones

El programador de instancias utiliza un registro estructurado optimizado para las consultas de Logs Insights. CloudWatch Esta solución registra información de procesamiento para cada instancia etiquetada, los resultados de la evaluación de períodos para la instancia, el estado deseado de la instancia durante ese período, la acción aplicada y mensajes de depuración.

Los registros se escriben en Amazon CloudWatch Logs en dos grupos de registros:

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

Registra el registro y la anulación del registro de recursos, las operaciones de recursos personalizadas, las solicitudes de CLI y otras actividades administrativas.

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

Registros para programar las operaciones, incluida la orquestación y la ejecución del controlador de solicitudes.

Los registros de advertencias y errores también se reenvían a un tema de Amazon SNS creado por una solución, que se puede configurar para enviar mensajes a una dirección de correo electrónico suscrita. Encontrará el nombre del tema de Amazon SNS en la pestaña Salidas de la pila de soluciones.

Etiquetas informativas

Cuando el etiquetado informativo está activado (opción predeterminada), Instance Scheduler escribe las etiquetas directamente en los recursos gestionados para ofrecer una visibilidad rápida de la actividad de programación de la solución. Estas etiquetas se aplican mediante la API de etiquetado de AWS Resource Groups y se actualizan cada vez que el programador procesa un recurso.

Puede activar o desactivar esta función mediante el parámetro Habilitar el etiquetado informativo del hub stack. Para obtener más información, consulte [Actualizar los ajustes de configuración global](#).

Claves de etiquetas informativas

Las siguientes etiquetas se escriben en los recursos gestionados:

Clave de etiqueta	Description (Descripción)
IS-ManagedBy	El ARN de la pila central de Instance Scheduler que administra a este recurso. Se aplica cuando un recurso se registra por primera vez para su programación y en cada acción de programación posterior.
IS-LastAction	La última acción de programación realizada en el recurso, junto con una marca de tiempo UTC. Por ejemplo: <code>Started 2025-06-15 09:00:00 UTC</code> o <code>Stopped 2025-06-15 17:00:00 UTC</code> . Esta etiqueta solo se actualiza cuando el programador inicia o detiene activamente un recurso (no cuando evalúa un recurso y determina que no es necesario realizar ninguna acción).
IS-Error	Si el programador encuentra un error al procesar un recurso, esta etiqueta contiene el código de error y una marca de tiempo UTC. Por ejemplo: <code>StartFailed 2025-06-15 09:00:05 UTC</code> . Esta etiqueta se borra automáticamente en la siguiente acción de programación correcta.
IS-ErrorMessage	Una descripción del error legible para las personas. Esta etiqueta solo está presente cuando también <code>IS-Error</code> está presente y se borra junto a ella.

Códigos de error

En la `IS-Error` etiqueta pueden aparecer los siguientes códigos de error:

Código de error	Description (Descripción)
<code>UnknownSchedule</code>	El nombre del programa especificado en la etiqueta del programa del recurso no coincide con ningún programa definido en la tabla de configuración.
<code>UnsupportedResource</code>	El tipo de recurso no se admite para la programación (por ejemplo, una réplica de lectura de otra instancia de RDS).
<code>IncompatibleSchedule</code>	La programación asignada al recurso no es compatible con el tipo de recurso (por ejemplo, una programación de ASG que utiliza expresiones cron no compatibles).
<code>StartFailed</code>	El programador intentó iniciar el recurso, pero la operación falló.
<code>StopFailed</code>	El programador intentó detener el recurso, pero la operación falló.
<code>ConfigurationFailed</code>	El programador intentó configurar las reglas de escalado programado en un grupo de Auto Scaling, pero la operación falló.
<code>UnknownError</code>	Se produjo un error inesperado al procesar el recurso.

Comportamiento de etiquetas

- Cuando un recurso se registra por primera vez para su programación, la `IS-ManagedBy` etiqueta se aplica inmediatamente.
- Cuando se anula el registro de un recurso (se elimina la etiqueta de programación), se eliminan todas las etiquetas informativas (`IS-ManagedBy`, `IS-LastAction`, `IS-ErrorMessage`) del recurso.
- Las etiquetas de error no se vuelven a escribir en cada intervalo de programación si el mismo error persiste y la etiqueta existente sigue presente en el recurso. Solo se actualizan cuando cambia el código de error.

- Todos los valores de las etiquetas se truncan a 256 caracteres para cumplir con los límites de etiquetado de AWS.

Consideraciones sobre la gobernanza de etiquetas

Important

El programador de instancias crea y actualiza las etiquetas enumeradas anteriormente en los recursos administrados como parte del funcionamiento normal. Si su organización aplica la gobernanza de etiquetas mediante reglas de AWS Config, políticas de etiquetas, políticas de control de servicios o soluciones automatizadas, asegúrese de que sus controles de administración de cambios estén configurados para permitir las siguientes claves de etiquetas:

- `IS-ManagedBy`
- `IS-LastAction`
- `IS-Error`
- `IS-ErrorMessage`
- `IS-PreferredInstanceTypes`(si utiliza tipos de instancias alternativos)
- `IS-MinDesiredMax`(si está programando grupos de Auto Scaling)

Si no puede incluir estas etiquetas en sus políticas de gobierno, desactive el etiquetado informativo configurando el parámetro `Habilitar el etiquetado informativo` No en el `hub stack`. Tenga en cuenta que esto también deshabilitará la `IS-ManagedBy` etiqueta que se usa para confirmar el registro del recurso.

Etiquetas de control

Además de las etiquetas informativas, Instance Scheduler utiliza las siguientes etiquetas de control para funciones específicas:

Clave de etiqueta	Description (Descripción)
<code>IS-PreferredInstanceTypes</code>	Una lista separada por comas de tipos de instancias EC2 alternativos para probar si se produce un error al iniciar una

Clave de etiqueta	Description (Descripción)
	instancia por falta de capacidad. Para obtener más información, consulta Cómo gestionar los errores de capacidad insuficiente de EC2 .
IS-MinDesiredMax	Los valores de capacidad mínima, deseada y máxima para un grupo de Auto Scaling en el formato <code>min,deseada,max</code> . Para obtener más información, consulte Programación de grupos de Auto Scaling de EC2 .

Capacidad de etiquetas

Important

Los recursos de AWS suelen tener un límite de 50 etiquetas por recurso. El programador de instancias puede usar hasta 6 etiquetas en un recurso (4 etiquetas informativas más 2 etiquetas de control). Asegúrese de que sus recursos tengan una capacidad de etiquetas suficiente para incluir las etiquetas del programador de instancias junto con su estrategia de etiquetado actual.

Si un recurso está en el límite de 50 etiquetas o cerca de él, es posible que no se puedan escribir las etiquetas informativas. El programador registra estos errores pero continúa programando las operaciones. Compruebe CloudWatch los registros si sospecha que hay problemas de etiquetado.

CloudWatch Registra las consultas de Insights

El formato de registro estructurado de Instance Scheduler permite realizar consultas eficientes con CloudWatch Logs Insights. Puede usar Logs Insights para buscar, analizar y visualizar los datos de registro para solucionar problemas operativos y monitorear la actividad de programación.

El programador de instancias proporciona consultas de registro preformateadas a las que puede acceder desde la sección Consultas guardadas de la consola de registros: CloudWatch

SchedulingHistory

Consulte las acciones de programación realizadas en los recursos, incluidas las operaciones de inicio y finalización.

RegistrationEvents

Consulte los eventos de registro y anulación del registro de recursos.

Errors

Consulte los registros de errores para solucionar problemas relacionados con la solución.

Para obtener más información sobre CloudWatch Logs Insights, consulte [Análisis de datos de registro con CloudWatch Logs Insights](#) en la Guía del usuario de Amazon CloudWatch Logs.

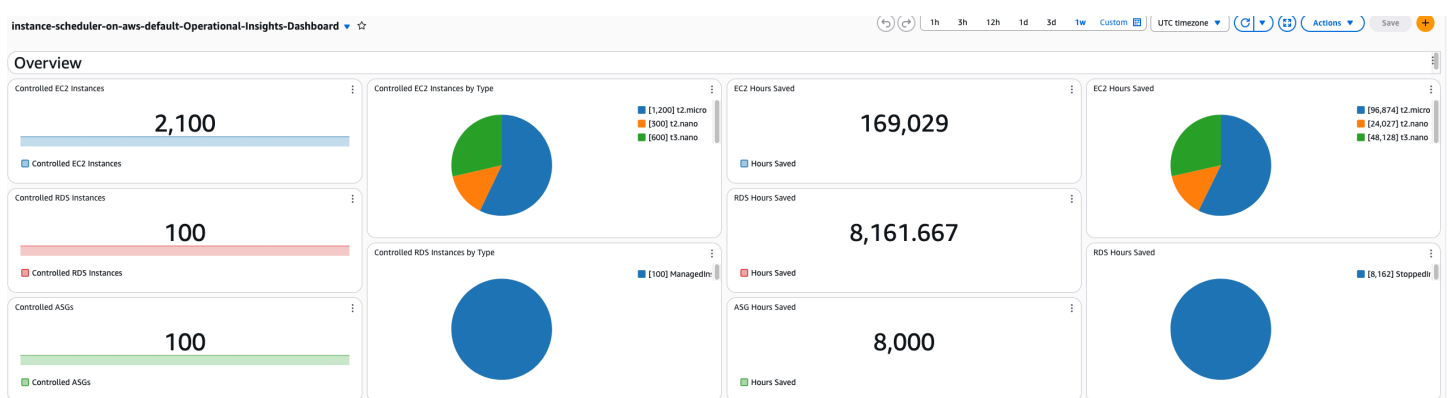
Panel de información operativa

El panel Operational Insights proporciona visibilidad del rendimiento de la solución y permite ahorrar costes gracias a la gestión programada de instancias.

Para acceder al panel, asegúrese de que la supervisión operativa esté configurada como «activada» en los parámetros del hub stack. Navegue hasta «Paneles de control» CloudWatch y selecciónelos en el menú de navegación. El nombre del panel es * {stack-name} - *. Operational-Insights-Dashboard

El panel muestra los recuentos de instancias gestionadas, las horas de ejecución guardadas y las métricas de rendimiento de la función Lambda.

Descripción general del panel de información operativa

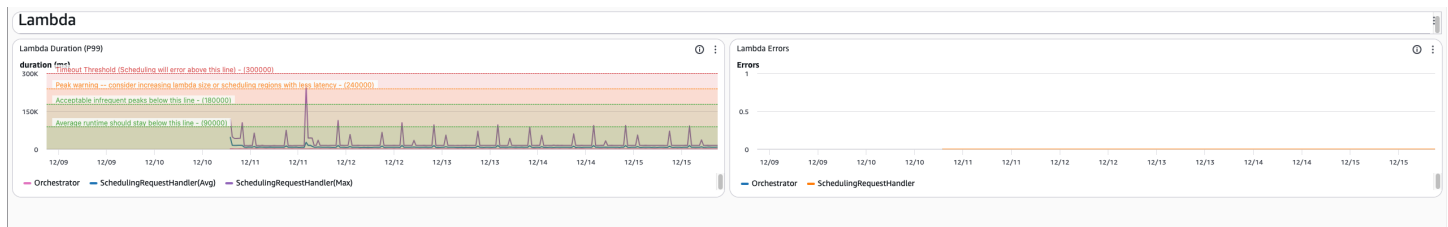


Note

La información de estos gráficos depende del intervalo de programación configurado en la pila central de la solución. Al actualizar el intervalo de programación de la solución, en el panel solo se mostrarán métricas de programación posteriores a la actualización más reciente del intervalo de programación.

Supervise los tiempos de ejecución de Lambda para garantizar un rendimiento óptimo (consulte [Cuotas](#)). Si los tiempos de ejecución se acercan constantemente al límite de tiempo de espera, considere la posibilidad de aumentar la propiedad de tamaño de Lambda o implementar Instance Scheduler en una región con menor latencia en las regiones gestionadas.

Métricas Lambda que muestran la duración y los recuentos de errores



Costos adicionales asociados a esta característica

Este panel operativo se basa en CloudWatch métricas personalizadas recopiladas por la solución, que generarán un costo adicional. Para desactivar esta característica, deshabilite “Supervisión operativa” en la pila central de la solución. Esta función cuesta 3 dólares adicionales. 00/month además de los costes de escalado adicionales en función del tamaño de la implementación. Los costos son los siguientes:

CloudWatch Panel de control personalizado	3 dólares
Per-instance-type métricas	0,90 USD por tipo de instancia*
Uso de la API	~0,10 \$ por objetivo activo**

*Estos costos se registran por categoría de servicio (EC2/RDS) y solo para los tipos de instancia que realmente se utilizan para la programación.

*

Supervisión de eventos EventBridge

Instance Scheduler publica los eventos de programación y registro en los buses de EventBridge eventos para ofrecer visibilidad de las operaciones de la solución y permitir la integración con otros servicios de AWS.

Event types (Tipos de eventos)

La solución publica dos categorías principales de eventos:

Programación de eventos: se publica cuando el Programador de instancias toma medidas para iniciar, detener o configurar los recursos administrados. Estos eventos incluyen detalles sobre la instancia, la programación y la acción realizada: iniciar, detener o configurar los recursos administrados. Estos eventos incluyen detalles sobre la instancia, el cronograma y la acción realizada.

Eventos de registro: se publican cuando los recursos están registrados o no registrados para programarlos en función de las operaciones de etiquetado.

Destinos de evento

IS-LocalEvents buses de eventos: se despliega un bus de IS-LocalEvents eventos en cada región gestionada de cada cuenta de miembro (incluida la cuenta central). Cada autobús recibe eventos para programar acciones y registros de recursos en esa región.

IS-GlobalEvents bus de eventos: el bus de IS-GlobalEvents eventos de la cuenta central recibe una copia de cada evento que se envía a cualquier bus de IS-LocalEvents eventos, lo que proporciona una supervisión centralizada en todas las cuentas y regiones.

Uso de EventBridge eventos

Puede crear EventBridge reglas para:

- Supervise las operaciones de programación en toda su infraestructura
- Active notificaciones cuando las instancias se inicien o detengan
- Intégrele con otros servicios de AWS para flujos de trabajo automatizados
- Implemente la supervisión y las alertas de conformidad

Estructura de evento

Todos los eventos utilizan el EventBridge formato estándar. Los siguientes ejemplos muestran la estructura de cada tipo de evento:

Programar un evento:

```
{
  "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"
  }
}
```

Evento de registro:

```
{
  "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"
  }
}
```

Cada evento contiene los siguientes campos clave:

- **Source**- Identifica la fuente del evento como «instance-scheduler»
- **DetailType**- Especifica la categoría del evento: «Acción de programación», por ejemplo, operaciones, o «Recurso registrado» para etiquetar eventos

- **Resources**- Matriz que contiene los ARN de los recursos de AWS afectados
- **Detail**- Contiene la carga útil del evento con el ID de cuenta, la región, el tipo de servicio (ec2/rds), el ID del recurso, el nombre del programa y, para la programación de eventos, tanto la acción solicitada como el resultado real

requested_actionValores posibles para programar eventos:

- **Start**: El programador está diseñado para iniciar la instancia
- **Stop**: El programador tenía la intención de detener la instancia
- **Configure**: El programador está destinado a configurar la instancia

action_takenValores posibles para programar eventos:

- **Started**: Se inició la instancia
- **Stopped**: Se detuvo la instancia
- **Hibernated**: La instancia estaba hibernada
- **Configured**: Se modificó la configuración de la instancia
- **Error**: Se ha producido un error durante la operación de programación

Creación de EventBridge reglas

Para supervisar los eventos del Programador de Instancias:

1. Navegue a la EventBridge consola de su cuenta de AWS
2. Cree una nueva regla dirigida al bus de **IS-GlobalEvents** eventos (para la supervisión centralizada) o al bus de **IS-LocalEvents** eventos (para la supervisión local)
3. Defina patrones de eventos para que coincidan con los eventos del Programador de Instancias
4. Configurar destinos como temas de SNS, funciones de Lambda o registros CloudWatch

Para obtener más información EventBridge, consulta [¿Qué es Amazon EventBridge?](#) en la Guía del EventBridge usuario de Amazon.

Resolución de problemas

En esta sección se proporcionan instrucciones de solución de problemas para la implementación y el uso de la solución.

La resolución de problemas conocidos proporciona instrucciones para mitigar los errores de este tipo. Si estas instrucciones no abordan su problema, el artículo de [Contacto con soporte](#) proporciona instrucciones que le ayudan a abrir un caso de soporte para esta solución.

Resolución de problemas conocidos

Problema: las instancias no se programan en una cuenta remota (v1.4-v3.0)

Si observa que las instancias no se programan en una cuenta remota.

Resolución

Actualice la pila central con el ID de cuenta secundaria o complete la siguiente tarea:

1. [En la cuenta principal, navega hasta la consola CloudWatch](#)
2. En el panel de navegación, seleccione Registros > Grupos de registro.
3. Seleccione el grupo de registro denominado `<STACK_NAME>-logs`.
4. Busque el flujo de registro para el ID de cuenta (cuenta remota).
5. Por ejemplo, si no hay ningún flujo de registro con el nombre del ID de cuenta, vaya a la consola de DynamoDB y seleccione la tabla denominada `<STACK_NAME>-<ConfigTable>-<RANDOM>`.
6. Seleccione Explorar elementos y, a continuación, Ejecutar.
7. Seleccione el tipo de elemento Config.
8. Compruebe si el atributo `remote_account_ids` tiene el ID de cuenta.
9. Compruebe si el ID de cuenta no está visible en este atributo.
10. Si la solución está configurada para AWS Organizations, desinstale la plantilla remota de la cuenta remota y vuelva a instalarla.
11. Si la solución está configurada para usar los ID de cuenta remota, actualice el parámetro de CloudFormation Proporcionar ID de organización o Lista de ID de cuentas remotas con la lista de ID de las cuentas donde se programarán las instancias y donde se implementa la plantilla remota.

Problema: las instancias no se están programando (versión 3.1 o posterior)

Si observa que las instancias no se están programando.

Resolución

1. Compruebe que el recurso tenga una IS-ManagedByetiqueta aplicada.
2. Si la etiqueta no está presente, elimine y vuelva a crear la etiqueta Schedule para volver a activar el registro.
3. Si la etiqueta sigue sin aplicarse, compruebe que la región esté habilitada para la programación:
 - a. Compruebe la configuración de hub/spoke pila de la región, o
 - b. Navegue hasta la [EventBridge consola](#) en la misma región que el recurso y compruebe que el bus de eventos predeterminado tiene reglas de eventos con el prefijo IS-Tagging.
4. Si la región no está habilitada, actualice la pila del programador de instancias para incluir la región en el parámetro regions CloudFormation .
5. Si el problema persiste, revise los [registros de administración de la solución](#) para ver si hay errores de registro del hub.
6. Confirme que su organización no cuente con políticas que impidan que los eventos se reenvíen de su cuenta a la cuenta de Solution Hub.

Problema: Las instancias de EC2 cifradas no se inician

El programador de instancias informa de que, pese a iniciarse las instancias de EC2 con volúmenes de EBS cifrados, no llegan a iniciarse nunca en realidad.

Resolución

Consulte [Volúmenes de EBS de EC2 cifrados](#) para descubrir cómo conceder acceso al programador de instancias a fin de poder programar instancias de EC2 con volúmenes de EBS cifrados.

Problema: el etiquetado informativo genera costes de API inesperados

Costes inesperadamente altos derivados del etiquetado de las llamadas a la API de AWS Resource Groups, de las evaluaciones de AWS Config o de las acciones correctivas relacionadas.

Resolución

El programador de instancias escribe [etiquetas informativas](#) en los recursos administrados en cada intervalo de programación. Si su entorno impone la gobernanza de etiquetas mediante reglas de AWS Config, políticas de etiquetas o soluciones automatizadas, asegúrese de que las claves de etiqueta del Programador de instancias estén permitidas. Para ver la lista completa de claves de etiquetas y la guía de configuración, consulte Consideraciones sobre la gobernanza de [etiquetas](#).

Si no puede actualizar sus políticas de control de etiquetas, desactive el etiquetado informativo configurando el parámetro Habilitar el etiquetado informativo No en el hub stack.

Problema: Las instancias de RDS no se detienen cuando está habilitado el permiso para crear instantáneas de RDS

Las instancias de RDS no se detienen y los registros del programador de la solución indican (AccessDenied) errores al llamar a la StopDBInstance operación debido a la falta de permiso.

```
rds:CreateDBSnapshot
```

Resolución

Actualice la solución a la versión 3.0.5 o a una más reciente, o bien agregue el permiso `rds:CreateDBSnapshot` al rol de programador de la solución en cada una de las cuentas programadas.

Póngase en contacto con AWS Support.

Si tiene [AWS Business Support+](#), [AWS Enterprise Support](#) o [Unified Operations](#), puede utilizar el AWS Support Center para obtener asistencia de expertos con esta solución. En las siguientes secciones, encontrará instrucciones.

Crear caso

1. Inicie sesión y vaya al [Centro de soporte](#).
2. Seleccione Crear caso.

¿Cómo podemos ayudarle?

1. Elija Técnico.

2. En Servicio, seleccione Soluciones.
3. En Categoría, seleccione Programador de instancias de AWS (Linux o Windows).
4. En Gravedad, seleccione la opción que mejor se adapte a su caso de uso.
5. Al especificar los valores de Servicio, Categoría y Gravedad, la interfaz rellena los enlaces a las preguntas más frecuentes de solución de problemas. Si no puede resolver su pregunta con estos enlaces, elija Paso siguiente: información adicional.

Información adicional

1. En Asunto, introduzca un texto que resuma su pregunta o problema.
2. Para obtener una descripción, describa el problema en detalle, incluyendo el nombre de este producto y la versión que está utilizando, como en este ejemplo: Instance Scheduler on AWS vX.Y.Z.
3. Elija Adjuntar archivos.
4. Adjunte la información que Soporte necesita para procesar la solicitud.

Ayúdenos a resolver su caso más rápido

1. Especifique la información requerida.
2. Elija Siguiendo paso: Resuelva ahora o póngase en contacto con nosotros.

Resuelva ahora o póngase en contacto con nosotros

1. Revise las soluciones de Resolver ahora.
2. Si estas no le ayudan a resolver su problema, elija Contactar con nosotros, especifique la información solicitada y seleccione Enviar.

Actualización de la solución

El programador de instancias está diseñado para que las actualizaciones in situ sean seguras mediante AWS CloudFormation. El procedimiento general para realizar esta acción es el siguiente:

1. Inicie sesión en la [CloudFormation consola de AWS](#), account/region donde esté instalada la pila de Hubinstance-scheduler-on-aws, seleccione y seleccione Actualizar pila.
2. Seleccione Realizar una actualización directa.
3. Seleccione Sustituir plantilla existente.
4. En Especificar plantilla:
 - Seleccione URL de Amazon S3.
 - Copie el enlace de la [plantilla más reciente](#).
 - Pegue el enlace en el cuadro URL de Amazon S3.
 - Verifique que la URL de la plantilla correcta aparezca en el cuadro de texto URL de Amazon S3 y seleccione Siguiente. Vuelva a seleccionar Siguiente.
5. En Parámetros, revise los parámetros de la plantilla y modifíquelos según sea necesario (consulte la lista de cambios importantes que aparece a continuación para ver las actualizaciones de parámetros necesarias). Para obtener más información sobre cada uno de los parámetros, consulte el [Paso 1. Lanzamiento de la pila del programador de instancias](#)
6. Elija Siguiente.
7. En la página Configurar opciones de pila, elija Siguiente.
8. En la página Revisar, revise y confirme la configuración. Seleccione la casilla para reconocer que la plantilla creará recursos de AWS Identity and Access Management (AWS IAM).
9. Seleccione Ver conjunto de cambios y verifique los cambios.
10. Seleccione Crear pila para implementar la pila.

Puede ver el estado de la pila en la CloudFormation consola de AWS en la columna Estado. Debería recibir el estado UPDATE_COMPLETE en unos minutos.

Repita los pasos anteriores para las pilas aws-instance-scheduler-remote en cada una de sus cuentas radiales.

Cambios importantes en versiones específicas

Al actualizar la solución, puede actualizar directamente desde una versión anterior a una versión más reciente sin perder datos importantes ni interrumpir la programación. Consulte a continuación una lista de los cambios de comportamiento y más importantes en cada versión principal.

Puede ver un registro de cambios completo en la página de [la solución GitHub](#)

v1.5.0

La versión 1.5.0 reemplaza la necesidad de proporcionar una lista de nombres de recurso de Amazon (ARN) del rol de programación entre cuentas por la posibilidad de administrarlos automáticamente a través de AWS Organizations. Si no desea utilizar AWS Organizations, puede proporcionar una lista de ID de cuenta radial y el programador de instancias administrará los roles de programación de forma automática.

Al actualizar a la versión 1.5.0 o a una versión más reciente, debe hacer lo siguiente:

1. Actualice la plantilla central mediante el proceso de actualización habitual y, al mismo tiempo, actualice los siguientes parámetros:
 - a. Elija un espacio de nombres único para la solución.
 - b. Seleccione si desea Usar AWS Organizations para administrar el registro radial de ahora en adelante.
 - i. Si seleccionó Sí, sustituya los ID de ID/Remote cuenta de la organización por el ID de su organización de AWS.
 - ii. Si seleccionó No, OrganizationID/RemoteAccountIDssustituya por una lista separada por comas de los ID de cuenta de sus cuentas de Spoke.
2. Actualice todas las pilas remotas mediante el proceso de actualización habitual y, al mismo tiempo, actualice los siguientes parámetros:
 - a. Espacio de nombres: el mismo que eligió para la cuenta central.
 - b. Usar AWS Organizations: el mismo valor que en la cuenta central.
 - c. ID de cuenta central: ID de cuenta de la cuenta central (no debe modificarse respecto al anterior).

v3.0.0

La versión 3.0.0 añade compatibilidad con los grupos de escalado automático de EC2 y divide la función lambda principal de la solución en funciones independientes con responsabilidades específicas a fin de proporcionar un mejor aislamiento de seguridad para cada función individual. Esta versión también actualiza el comportamiento de los registros de programación para incluir registros de «SchedulingDecision» a fin de comprender mejor las operaciones de programación.

V3.0.0 contiene los siguientes cambios importantes en comparación con las versiones anteriores:

- La función «CloudWatch Métricas» de la versión 1.5.x se ha sustituido por el panel [Operational Insights](#).
- Per-schedule las métricas de se CloudWatch han movido de Schedule/Service/MetricName → Schedule/Service/SchedulingInterval/MetricName
- Todas las métricas existentes seguirán estando disponibles, pero ahora se recopilarán las nuevas métricas en el nuevo espacio de nombres y estarán disponibles en el panel de soluciones.
- Los ARN clave de KMS para su uso con volúmenes de EBS cifrados en instancias de base de datos EC2 ahora deben proporcionarse a la hub/spoke CloudFormation pila de sus cuentas respectivas. (para obtener más información, consulte [Volúmenes de EBS de EC2 cifrados](#)).
 - Si está programando los EC2 con volúmenes de EBS cifrados, tendrá que copiar los arns clave de KMS que se utilizan en los parámetros de la pila. hub/spoke
- El CloudFormation parámetro de los servicios programados se ha dividido en parámetros individuales para cada servicio compatible.
 - Todos los servicios estarán habilitados de forma predeterminada y se pueden deshabilitar de forma individual.
- La versión 3.0 del programador de instancias no es compatible con versiones anteriores de la CLI del programador de instancias.
 - Deberá actualizar a la versión más reciente de la CLI del programador de instancias para seguir usando los comandos de la CLI.

Además de lo anterior, se ha actualizado el esquema de la tabla Ventana de mantenimiento y se reemplazará como parte de la actualización. Esto restablecerá el seguimiento de las ventanas de mantenimiento de EC2 durante los primeros minutos tras la actualización a la versión 3.x y, en raras ocasiones, puede hacer que las instancias que actualmente se encuentran dentro de una ventana de mantenimiento se detengan de forma prematura inmediatamente después de la actualización.

Una vez que se hayan regenerado estos datos, las operaciones de programación continuarán funcionando con normalidad.

v3.1.0

La versión 3.1.0 refactoriza la infraestructura principal de la solución para usar los eventos de etiquetado de AWS para realizar un seguimiento de cuándo se etiquetan los recursos para su programación. Asegúrese de que los permisos de su organización permitan enviar estos eventos de etiquetado desde las cuentas de los miembros a su cuenta central.

Al actualizar a la versión 3.1.0 o posterior:

- Las cuentas Spoke ahora declaran las regiones programadas independientemente de la cuenta central. Cada pila de radios debe especificar qué regiones programar en esa cuenta mediante el parámetro Regiones.
- El modo AWS Organizations ahora es necesario para las implementaciones con más de 40 cuentas en total. Si tiene más de 40 cuentas y no utiliza el modo Organizations, debe habilitarlo durante la actualización.
- Si tiene instancias EC2 administradas en AWS License Manager que desea programar, añada los ARN de configuración de License Manager al parámetro ARN de configuración de License Manager de sus pilas. hub/spoke CloudFormation Para obtener más información, consulte [EC2 License Manager](#).
- La solución aplicará automáticamente una IS-ManagedBy etiqueta a los recursos una vez que se hayan etiquetado para su programación, a fin de indicar que están siendo gestionados por el programador.
- (Restaurado en la versión 3.2.0) El cambio de tamaño programado de las instancias (definido `period-name@size` en una programación) se eliminó temporalmente en la versión 3.1.0, pero se volvió a implementar en la versión 3.2.0 y versiones posteriores. [Consulte](#) el tipo de instancia.
- Ya no se permite enumerar las cuentas de los miembros mediante un parámetro SSM (pasando `{param: ssm-param-name}` al parámetro de cuentas del hub stack). Todas las cuentas de confianza deben transferirse al hub stack en el momento de la implementación.
- El programador de instancias requerirá hasta 6 etiquetas únicas en los recursos durante la programación. Asegúrese de que los recursos tengan suficiente capacidad de etiquetado cuando se combinen con el resto de la estrategia de etiquetado de su organización.
- Per-schedule se han eliminado las métricas de. CloudWatch

- Los registros de soluciones se han vuelto a empaquetar en grupos de registros administrativos y de programación separados y se han optimizado para realizar consultas con CloudWatch Log Insights. Consulte [Supervisión de la solución](#) para obtener más información.
- Las etiquetas de inicio y parada ya no se pueden configurar mediante CloudFormation parámetros. La solución ahora utiliza nombres de etiquetas fijos con información más detallada para realizar un seguimiento de las acciones de programación.

 Important

El programador de instancias escribe hasta 6 etiquetas únicas en los recursos gestionados durante el funcionamiento normal. Asegúrese de que sus políticas de gobierno de etiquetas (como las reglas de AWS Config, las políticas de etiquetas o la corrección automática) estén configuradas para permitir estas etiquetas. Para obtener una lista completa de etiquetas y consideraciones importantes de gobernanza, consulte las [etiquetas informativas](#).

Desinstalar la solución

Important

Al desinstalar la solución, asegúrese de desinstalar todas las pilas de programación personalizadas antes de la solución en sí.

Puede desinstalar la solución del programador de instancias de AWS desde la Consola de administración de AWS o mediante la Interfaz de la línea de comandos de AWS. Para desinstalar la solución, elimine la pila central de AWS CloudFormation junto con todas las pilas remotas instaladas. A continuación, puede eliminar cualquier etiqueta de programación que se haya aplicado a las instancias con fines de programación.

Note

Si la opción Proteger tablas de DynamoDB está habilitada en la pila central de la solución CloudFormation, conservará las tablas de DynamoDB y la clave KMS de la solución en lugar de eliminarlas. Si desea eliminar estos recursos, asegúrese de que esta propiedad esté establecida en Deshabilitada antes de eliminar la pila central. Como alternativa, puede eliminarlos de forma manual una vez que se haya eliminado la pila central.

Uso de Consola de administración de AWS

1. Inicie sesión en la [CloudFormation consola de AWS](#).
2. En la página Pilas, seleccione la pila de instalación de esta solución.
3. Elija Eliminar.

Uso de la Interfaz de la línea de comandos de AWS

Determine si la Interfaz de la línea de comandos de AWS (AWS CLI) está disponible en su entorno. Para obtener instrucciones de instalación, consulte [¿Qué es la Interfaz de la línea de comandos de AWS?](#) de la Guía del usuario de AWS CLI. Tras confirmar que la CLI de AWS está disponible, ejecute el siguiente comando:

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

Guía para desarrolladores

En esta sección se proporciona el código fuente correspondiente a la solución y se enumeran las secciones aquí agregadas, además de incluirse enlaces a cada subtema.

Código fuente

Visite nuestro [GitHub repositorio](#) para descargar los archivos fuente de esta solución y compartir sus personalizaciones con otras personas.

Las plantillas del programador de instancias de AWS se generan mediante [AWS CDK](#). Consulte el enlace: [README.md](#) archivo para obtener información adicional.

Referencia

Esta sección incluye información sobre la recopilación de datos, sugerencias sobre [los recursos relacionados](#) y una [lista de los desarrolladores](#) que han contribuido a esta solución.

Recopilación de datos

Esta solución envía métricas operativas a AWS (los «datos») sobre el uso de esta solución. Utilizamos estos datos para comprender mejor cómo utilizan los clientes esta solución y los servicios y productos relacionados. La recopilación de estos datos por parte de AWS está sujeta al [Aviso de privacidad de AWS](#).

Recursos relacionados

El [Programador de recursos](#) es similar al programador de instancias de AWS, pero su implementación presenta las siguientes diferencias:

El programador de instancias de AWS utiliza una función de Lambda para evaluar con frecuencia las programaciones almacenadas en su configuración y comprueba si las instancias tienen el estado deseado. La configuración rápida del programador de recursos utiliza horas de inicio y de detención para realizar acciones de inicio y detención mediante manuales de procedimientos de SSM. Esto sucede cuando la hora actual es igual a la hora de inicio o la ha superado.

Actualmente, el programador de instancias de AWS permite la programación de clústeres de EC2, RDS y Aurora. El programador de recursos solo programa o inicia y detiene instancias de EC2.

Utilice el programador de recursos para identificar las instancias de EC2 y las start/stop mismas en momentos específicos.

Utilice Instance Scheduler en AWS cuando las cuentas deban escanearse con regularidad en busca de instancias. start/stop

En la tabla se identifica la mejor solución en función de los escenarios:

Escenario	Programador de recursos	Programador de instancias de AWS
Programar instancias de Amazon Neptune	No	Sí
Programar instancias de Amazon DocumentDB	No	Sí
Programar instancias de grupo de escalado automático	No	Sí
Programar instancias de EC2	Sí	Sí
Programar instancias de RDS	No	Sí
Programar clústeres de Aurora	No	Sí
Administrar programaciones en una sola cuenta (cuenta central)	No	Sí
Administrar programaciones en cuentas individuales	Sí	No
Cambiar la integración del calendario	Sí	No
Solo acciones de inicio y detención	Sí	No
Supervisión periódica de instancias e inicio y detención de estas según su estado actual	No	Sí

Colaboradores

- Arie Leeuwesteijn
- Mahmoud ElZayet
- Ruald Andreae
- Nikhil Reddy
- Caleb Pearson
- Jason DiDomenico
- Max Granat
- Pratyush Das
- Amanda Jones
- Kevin Hargita
- Beomseok Lee
- Abe Wubshet

Revisiones

Fecha de publicación: octubre de 2020

Compruebe el [CHANGELOG.md](#) archivo del GitHub repositorio para ver todos los cambios y actualizaciones importantes del software. El registro de cambios proporciona un historial claro de mejoras y correcciones de cada versión.

Avisos

Es responsabilidad de los clientes realizar su propia evaluación independiente de la información que contiene este documento. El presente documento: (a) tiene solo fines informativos, (b) representa las ofertas y prácticas actuales de los productos de AWS, que están sujetas a cambios sin previo aviso, y (c) no supone ningún compromiso ni garantía por parte de AWS y sus filiales, proveedores o licenciantes. Los productos o servicios de AWS se proporcionan “tal cual” sin garantías, declaraciones ni condiciones de ningún tipo, ya sean expresas o implícitas. Las responsabilidades y obligaciones de AWS con respecto a sus clientes se controlan mediante los acuerdos de AWS y este documento no forma parte ni modifica ningún acuerdo entre AWS y sus clientes.

El programador de instancias de AWS tiene una licencia en virtud de los términos de la [versión 2.0 de la licencia Apache](#).

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.