



Guía de implementación

# Programador de instancias en AWS



# Programador de instancias en AWS: Guía de implementación

Copyright © 2024 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
Ejemplos de precios (mensuales) .....	5
Soportado Regiones de AWS .....	11
Programación de instancias multicuenta mediante el ID de la cuenta IDs o AWS de la organización .....	11
Habilitar la programación multicuenta mediante Account IDs .....	12
Habilitar la programación multicuenta mediante el ID de AWS la organización .....	12
Administración de cuentas IDs con el 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
AWS Consideraciones sobre el diseño de Well-Architected .....	19
Excelencia operativa .....	19
Seguridad .....	19
Fiabilidad .....	19
Eficiencia del rendimiento .....	20
Optimización de costos .....	20
Sostenibilidad .....	20
Tabla de configuración del planificador .....	21
Programador CLI .....	21
AWS servicios utilizados en esta solución .....	21
Seguridad .....	24
AWS KMS .....	24
Amazon IAM .....	24
Volúmenes de EC2 EBS cifrados .....	25

Introducción .....	27
Información general del proceso de implementación .....	27
AWS CloudFormation plantillas .....	28
Paso 1: lanza la pila de concentradores del programador de instancias .....	28
Paso 2 (opcional): lanza la pila remota en las cuentas secundarias .....	32
Configure la solución .....	35
Guía del operador .....	36
Configure los horarios .....	36
Uso de la infraestructura como código (recomendado) .....	36
Uso de la consola Amazon DynamoDB y el programador de instancias en AWS CLI .....	36
Etiquete las instancias para programarlas .....	37
Establecer el valor de la etiqueta .....	37
EC2 instancias con volúmenes de EBS cifrados .....	38
Referencia de horario .....	38
Periodos .....	38
Zona horaria .....	38
Campo de hibernación .....	39
Campo obligatorio .....	39
Conserva el campo de atletismo .....	39
Campo de ventana de mantenimiento de Systems Manager (solo se aplica a EC2 las instancias) .....	39
Tipo de instancia .....	40
Definiciones de horarios .....	41
Período de referencia .....	43
Horarios de inicio y finalización .....	43
Días de la semana .....	45
Días del mes .....	45
Meses .....	45
Definiciones de períodos .....	45
Etiquetado automatizado .....	48
Ejemplos de horarios .....	49
Horario de trabajo estándar de 9 a 5 .....	49
Detenga las instancias después de las 5 p.m. ....	51
Detenga las instancias durante el fin de semana .....	53
Recursos de soluciones .....	56
Programador CLI .....	58

Requisitos previos .....	58
Credenciales .....	58
Instale la CLI del programador .....	59
Estructura de comandos .....	60
Argumentos comunes .....	60
Comandos disponibles .....	61
período de creación .....	62
crear-programar .....	64
período de eliminación .....	66
eliminar-programar .....	67
describa los períodos .....	67
describa los horarios .....	69
describe-schedule-usage .....	70
período de actualización .....	71
calendario de actualizaciones .....	72
ayuda .....	72
Actualizar los ajustes de configuración global .....	73
Gestione los horarios mediante la infraestructura como código (IaC) .....	74
Características avanzadas .....	77
EC2 Programación de Auto Scaling Group .....	77
Supervise la solución .....	79
Registro y notificaciones .....	79
Archivos de registro .....	80
Panel de información operativa .....	80
Supervise la solución con Service Catalog AppRegistry .....	83
Rendimiento .....	87
Actualización de la solución .....	87
Cambios importantes en versiones específicas .....	89
v1.5.0 .....	89
v3.0.0 .....	90
Solución de problemas .....	91
Resolución de problemas conocidos .....	91
Problema: las instancias no se están programando en una cuenta remota .....	91
Resolución .....	91
Problema: actualización de la solución desde cualquier versión v1.3.x a la v1.5.0 .....	92
Resolución .....	92

Problema: EC2 las instancias cifradas no se inician .....	93
Resolución .....	93
Problema: las instancias de RDS no se detienen cuando la función Crear instantáneas de RDS está habilitada .....	93
Resolución .....	93
Contacto Soporte .....	93
Crear caso .....	93
¿Cómo podemos ayudar? .....	94
Información adicional .....	94
Ayúdenos a resolver su caso más rápido .....	94
Resuelva ahora o póngase en contacto con nosotros .....	94
Desinstalar la solución .....	95
Usando el AWS Management Console .....	95
Usando AWS Command Line Interface .....	95
Guía para desarrolladores .....	97
Código fuente .....	97
Referencia .....	98
Recopilación de datos anonimizados .....	98
Cuotas .....	100
Cuotas de AWS servicios en esta solución .....	101
AWS CloudFormation cuotas .....	101
Recursos relacionados .....	101
Colaboradores .....	103
Revisiones .....	104
Avisos .....	109
.....	CX

# Automatice el inicio y la detención de AWS instancias

Fecha de publicación: octubre de 2020 ([última actualización](#): enero de 2025)

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

Esta solución ayuda a reducir los costos operativos al detener los recursos que no se utilizan e iniciarlos cuando se necesita su capacidad. Por ejemplo, una empresa puede utilizar el Programador de instancias AWS para detener automáticamente las instancias fuera del horario laboral todos los días. Si deja todas las instancias en funcionamiento a pleno rendimiento, esta solución puede suponer un ahorro de costes de hasta un 70% para las instancias que solo son necesarias durante el horario laboral habitual (el uso semanal se reduce de 168 a 50 horas).

Instance Scheduler AWS on utiliza las etiquetas de recursos de Amazon Web Services (AWS) [AWS Lambda](#) para detener y reiniciar automáticamente las instancias en varias cuentas Regiones de AWS y según un cronograma definido por el cliente. Esta solución también le permite utilizar la hibernación para las instancias detenidas. EC2

Esta guía de implementación proporciona una descripción general del programador de instancias de la AWS solución, su arquitectura y componentes de referencia, las consideraciones para planificar la implementación y los 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 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.	<a href="#">Costo</a>
El costo estimado de ejecutar esta solución en la región de EE. UU. Este (Virginia del Norte) es de 13,15 USD al mes.	
Comprender las consideraciones de seguridad de esta solución.	<a href="#">AWS Seguridad WellArchitected</a>

Si quiere...	Lea...
	<a href="#">Seguridad</a>
Configure los horarios.	<a href="#">Tabla de configuración del planificador</a>
Conozca cuáles Regiones de AWS son compatibles con esta solución.	<a href="#">Soportado Regiones de AWS</a>
Vea o descargue la AWS CloudFormation plantilla incluida en esta solución para implementar automáticamente los recursos de infraestructura (la «pila») de esta solución.	<a href="#">AWS CloudFormation plantillas</a>
Acceda al código fuente y, si lo desea, utilice AWS Cloud Development Kit (AWS CDK) (AWS CDK) para implementar la solución.	<a href="#">GitHubrepositorio</a>

## Características y ventajas

El cronograma de instancias de la AWS solución ofrece las siguientes funciones:

### Programación de instancias entre cuentas

Esta solución incluye una plantilla que crea las funciones [AWS Identity and Access Management](#) (de IAM) necesarias para iniciar y detener instancias en cuentas secundarias. Para obtener más información, consulta la [sección Programación de instancias entre cuentas](#).

### Etiquetado automatizado

Instance Scheduler activado AWS puede añadir etiquetas automáticamente a todas las instancias que inicie o detenga. La solución también incluye macros que permiten añadir información variable a las etiquetas.

### Configure los horarios o períodos mediante la CLI del programador

Esta solución incluye una interfaz de línea de comandos (CLI) que proporciona comandos para configurar horarios y períodos. La CLI permite a los clientes estimar los ahorros de costos para un cronograma determinado. Para obtener más información, consulte la [CLI del programador](#).

## Administre los horarios mediante la infraestructura como código (IaC)

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

### Integración con Windows de mantenimiento de Systems Manager

Para EC2 las instancias de Amazon, Instance Scheduler on AWS puede integrarse con las 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.

### Integración con Service Catalog AppRegistry y Application Manager, una capacidad de AWS Systems Manager

Esta solución incluye un AppRegistry recurso de [Service Catalog](#) para registrar la CloudFormation plantilla de la solución y sus recursos subyacentes como una aplicación tanto en Service Catalog AppRegistry como en [Application Manager](#). Con esta integración, puede administrar de forma centralizada los recursos de la solución.

## Casos de uso

### Ejecutar instancias solo durante el horario laboral

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

### Detener las instancias fuera del horario laboral

Si quiere asegurarse de que las instancias de desarrollo estén desactivadas fuera del horario laboral y hasta que se vuelvan a necesitar, puede usar esta solución para establecer un período de finalización sin un período de inicio. Para obtener más información, consulta el [ejemplo de programa](#).

## Conceptos y definiciones

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

horario

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

periodo

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

instancia

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

horario comercial habitual

De 9:00 a 17:00 (de 9 a. m. a 5 p. m.) ET de lunes a viernes

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

## Costo

Usted es responsable del costo de los AWS servicios utilizados al ejecutar Instance Scheduler en AWS. Según la última revisión, el coste de ejecutar esta solución (una implementación pequeña en dos cuentas y dos regiones) es de aproximadamente 13,15\$ al mes. Consulte los siguientes ejemplos de tablas de costos para obtener un desglose más detallado.

Instance Scheduler on AWS está diseñado para invocar AWS Lambda funciones varias veces por ciclo de ejecución. Por ejemplo, si usa la solución para administrar las instancias de Amazon EC2 y Amazon RDS en una región para dos cuentas (una cuenta en la que se implementa la solución y la otra cuenta es una cuenta cruzada), la solución ejecuta cinco invocaciones de funciones Lambda:

- Una para gestionar la solicitud de orquestación inicial de Event Bridge, que se invoca en función de la frecuencia seleccionada (por defecto: cinco minutos).
- Una invocación Lambda adicional para cada servicio, cuenta y región.
- Si la [programación de Auto Scaling Group](#) está habilitada, se ejecuta otra invocación de orquestación para todas las cuentas o regiones cada hora.

Las métricas operativas personalizadas añaden costes adicionales en función de la cantidad de programas y tipos de instancias (como m2.medium, t3.large) que programe la solución. Si no desea realizar un seguimiento de estas métricas, puede desactivar esta función para ahorrar costes.

Consulte el [panel de información operativa](#) para obtener más información sobre estas métricas y sus costes asociados.

Esta solución utiliza el escalado bajo demanda para sus tablas de [Amazon DynamoDB](#) a fin de proporcionar suficiente capacidad de lectura y escritura.

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

El costo de la solución por ejecución depende de la cantidad de instancias que etiquete y administre la solución. A medida que aumenta el número de EC2 instancias de base de datos RDS, el tiempo de ejecución de Lambda también aumenta proporcionalmente.

Recomendamos crear un [presupuesto](#) para ayudar AWS Cost Explorer a gestionar los costes. Los precios están sujetos a cambios.

#### Note

Para optimizar los costes, Instance Scheduler agrupa todos los servicios relacionados con Amazon RDS en una sola invocación. Por lo tanto, incluso si habilita la programación de [Amazon RDS](#), [Amazon Aurora](#), [Amazon Neptune](#) y [Amazon DocDB](#), esto seguirá considerándose simplemente «RDS» a los efectos del cálculo de costes.

## Ejemplos de precios (mensuales)

### Implementación pequeña

Este ejemplo de precios se basa en las siguientes suposiciones:

- Dos cuentas, dos regiones, que programan todos los servicios posibles
- 3 horarios en uso activo
- 20 instancias de 3 tamaños diferentes
- Intervalo de programación: 5 minutos
- Tamaño de la función Lambda: 128 MB
- Tiempo de ejecución promedio de la función Lambda: 8 segundos

AWS servicio	Dimensiones	Coste mensual [USD]
AWS Lambda	288 o 24 ejecuciones programadas por día	~1,50 \$

AWS servicio	Dimensiones	Coste mensual [USD]
	<p>1+8 funciones Lambda por ejecución</p> <p>Tiempo de ejecución promedio de Lambda de 8 segundos</p> <p>(0,0000021 €/segundo)</p> <p>(0,0000002/llamada a la función Lambda)</p>	
Métricas operativas (opcional)	<p>CloudWatch panel de control (3\$ al mes)</p> <p>3 métricas per-instance-type (0,90 USD al mes)</p> <p>3 métricas por programa * 2 servicios (0,60\$ al mes)</p> <p>PutMetric Aproximadamente 80 000 llamadas al mes (0,01/1000 \$)</p>	~10,00 \$
Amazon DynamoDB	<p>Aproximadamente 75 000 WRU/mes (1,25 dólares por millón)</p> <p>Aproximadamente 100 000 RU/mes (0,5\$ por millón)</p> <p>costes de almacenamiento insignificantes (&lt; 0,01\$)</p>	~0,15 \$

AWS servicio	Dimensiones	Coste mensual [USD]
AWS KMS	1 AWS KMS llave (1\$ al mes)  Aproximadamente 140 000 solicitudes de API al mes (0,30 €/10 000\$)	~1,50 \$
Total:		~13,15 \$

## Despliegue medio

Este ejemplo de precios se basa en las siguientes suposiciones:

- 50 cuentas, 4 regiones, programando todos los servicios compatibles
- 10 programas en uso activo
- 200 instancias de 10 tamaños diferentes
- Intervalo de programación: 5 minutos
- Tamaño de la función Lambda: 128 MB
- Tiempo de ejecución promedio de Lambda: 8 segundos
- 5. Windows EC2 de mantenimiento

AWS servicio	Dimensiones	Coste mensual [USD]
AWS Lambda	288 o 24 ejecuciones programadas por día  1+400 funciones Lambda por ejecución  Tiempo de ejecución promedio de Lambda de 8 segundos  (0,0000021 €/segundo)	~64,00 \$

AWS servicio	Dimensiones	Coste mensual [USD]
	(0,0000002/llamada a la función Lambda)	
Métricas operativas (opcional)	<p>CloudWatch panel de control (3\$ al mes)</p> <p>10 métricas per-instance-type (0,90\$ al mes)</p> <p>10 métricas por programa * 2 servicios (0,60\$ al mes)</p> <p>PutMetric Aproximadamente 3,5 millones de llamadas al mes (0,01 USD/1000)</p>	~60,00 \$
Amazon DynamoDB	<p>Aproximadamente 7 millones de WRU/mes (1,25 dólares por millón)</p> <p>Aproximadamente 8 millones de RRU al mes (0,5\$ por millón)</p> <p>Costes de almacenamiento (&lt;0,01 \$)</p>	~12,00 \$
AWS KMS	<p>1 AWS KMS llave (1\$ al mes)</p> <p>Aproximadamente 7 millones de solicitudes de API al mes (0,30/10 000\$)</p>	~22,00 \$
Total:		~158,00 \$

## Gran despliegue

Este ejemplo de precios se basa en las siguientes suposiciones:

- 120 cuentas, 6 regiones, programando Amazon EC2 y Amazon RDS
- 100 programas en uso activo
- 2000 instancias de 50 tamaños diferentes
- 100 ventanas EC2 de mantenimiento
- Intervalo de programación: 5 minutos
- Tamaño de la función Lambda: 128 MB
- Tiempo de ejecución promedio de la función Lambda: 8 segundos

AWS servicio	Dimensiones	Coste mensual [USD]
AWS Lambda	288 o 24 ejecuciones programadas por día  1+1440 funciones Lambda por ejecución  Tiempo de ejecución promedio de una función Lambda de 8 segundos  (0,0000021 €/segundo)  (0,0000002/llamada Lambda)	~230,00 \$
Métricas operativas (opcional)	CloudWatch panel de control (3\$ al mes)  50 métricas per-instance-type (0,90\$ al mes)  100 métricas programadas * 2 servicios (0,60\$ al mes)	~300,00 \$

AWS servicio	Dimensiones	Coste mensual [USD]
	PutMetric Aproximadamente 3,5 millones de llamadas al mes (0,01 USD/1000)	
Amazon DynamoDB	Aproximadamente 26 millones de WRU/mes (1,25 dólares por millón)  Aproximadamente 26 millones de RRU al mes (0,5\$ por millón)  Costes de almacenamiento (<0,01 \$)	~40,00 \$
AWS KMS	1 clave KMS (1\$ al mes)  Aproximadamente 25 millones de solicitudes de API al mes (0,30/10 000\$)	~80,00 \$
Total:		~650,00 \$

Para configurar la solución de manera eficiente, tenga en cuenta lo siguiente:

1. Implemente la solución en una región donde el costo de la función Lambda sea más bajo.
2. No cambie la memoria de la función Lambda (CloudFormation parámetro Memoria a menos que sea absolutamente necesario). Esto aumentará considerablemente el coste de la solución.
3. Elimine los cronogramas no utilizados de las configuraciones de la solución.
4. Seleccione una frecuencia que reduzca el número de ejecuciones de la función Lambda por día. Por ejemplo, si las programaciones están separadas por horas, establezca la frecuencia ( CloudFormation parámetro Frecuencia) en incrementos de una hora. De forma predeterminada, la solución está configurada en cinco minutos, lo que significa que la función Lambda se invocará 288 veces al día, mientras que una frecuencia de una hora se ejecutará 24 veces al día.

## Compatible Regiones de AWS

Puede implementar Instance Scheduler en cualquier región Región de AWS, incluidas las regiones AWS GovCloud (de 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 parada adecuadas a las instancias de bases de datos etiquetadas EC2 y de RDS en cualquier región de su cuenta. Si utiliza la programación de instancias multicuenta, la solución aplica acciones a las instancias de todas las regiones configuradas de todas las cuentas.

### Important

AWS Las acciones del Programador Regiones de AWS de instancias afectan a las instancias etiquetadas adecuadamente en toda tu cuenta, aunque la función Lambda se ejecute en una sola región.

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

Cada implementación busca en cada instancia de cada región configurada de una cuenta 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

En el caso de las regiones de suscripción voluntaria, Instance Scheduler activado AWS puede segmentar instancias de cualquier región de suscripción, pero las propias CloudFormation pilas actualmente solo están disponibles para su implementación en las siguientes regiones de suscripción.

## Programación de instancias multicuenta mediante el ID de la cuenta o de la organización IDs AWS

Esta solución incluye una plantilla ([instance-scheduler-on-aws-remote.template](#)) que crea las funciones [AWS Identity and Access Management \(de IAM\)](#) y otros recursos necesarios para que

la solución pueda empezar a programar en las cuentas secundarias. Puede revisar y modificar los permisos de la plantilla remota antes de lanzar la pila.

## Habilitar la programación multicuenta mediante Account IDs

Para aplicar programas automatizados de inicio y parada a los recursos de las cuentas secundarias:

1. Inicie sesión [AWS Management Console](#) y seleccione el botón para iniciar la [instance-scheduler-on-aws](#) AWS CloudFormation plantilla en la cuenta principal.
2. Inicie la plantilla remota ([instance-scheduler-on-aws-remote](#)) en cada cuenta secundaria aplicable. Cuando se lanza cada pila remota, se crea un rol multicuenta Amazon Resource Name (ARN).
3. Actualice la pila de soluciones principal con el ID de cuenta en los IDs parámetros de proporcionar el ID de la organización o la lista de cuentas remotas para que la solución pueda iniciar y detener las instancias de las cuentas secundarias.

## Habilitar la programación multicuenta mediante el ID de AWS la organización

Para aplicar programas automatizados de inicio y parada a los recursos de las cuentas secundarias:

1. Inicie sesión [AWS Management Console](#) y seleccione el botón para iniciar la [instance-scheduler-on-aws](#) AWS CloudFormation plantilla en la cuenta principal.
2. Establezca el CloudFormation parámetro `AWS Organizations¿Usando?` como Sí, e introduzca el identificador de la organización en los IDs CloudFormation parámetros Proporcionar el identificador de la organización o la lista 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 de la misma región que la solución de la cuenta principal. Cuando cada pila remota se lance correctamente, la cuenta de la solución principal se actualizará con el identificador de la cuenta sin que se produzcan más cambios en la cuenta principal.

## Administración de cuentas IDs con el almacén de parámetros de AWS Systems Manager

Utilice el almacén de parámetros de AWS Systems Manager para almacenar la cuenta remota IDs. Puede almacenar la cuenta remota IDs como un parámetro de lista donde cada elemento es un

identificador de cuenta, o como un parámetro de cadena que contiene una lista de cuentas remotas delimitada por comas. IDs El parámetro tiene el formato {param: name}, donde el nombre es el nombre del parámetro en el almacén de parámetros.

Para aprovechar esta función, debe lanzar el programador de instancias en el AWS hub stack de la misma cuenta que su almacén de parámetros.

## Servicios compatibles con la programación

AWS Actualmente, Instance Scheduler on 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 automáticamente EC2 las instancias y supone que el comportamiento de cierre de las instancias está configurado como Detener, no como Terminarse. Ten en cuenta que no puedes reiniciar una EC2 instancia de Amazon una vez finalizada.

De forma predeterminada, EC2 las instancias están configuradas para detenerse, no para terminarse, cuando se cierran, pero puedes [modificar este comportamiento](#). Por lo tanto, asegúrese de que las instancias que controla con el programador de instancias AWS estén configuradas con un comportamiento de parada de apagado; de lo contrario, se cancelarán.

### Amazon RDS, Amazon Neptune y Amazon DocumentDB

Esta solución está diseñada para detener, no eliminar, automáticamente las instancias de RDS, Neptune y DocDB. Puede usar el parámetro de AWS CloudFormation plantilla Crear instantánea de instancia de RDS para crear instantáneas de las 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 las instancias de base de datos de RDS que forman parte de un clúster de Aurora o que administran bases de datos de Aurora. Debe etiquetar el clúster (no las instancias individuales) con la clave de etiqueta que definió durante la configuración inicial y el nombre del programa como valor de etiqueta para programar ese clúster.

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

Cuando se detiene una instancia de base de datos de RDS, se borra la caché, lo que puede provocar un rendimiento más lento al reiniciar la instancia.

## Ventana de mantenimiento de Amazon RDS

Cada instancia de base de datos de RDS tiene un período de [mantenimiento semanal durante el cual se](#) aplican los cambios en el sistema. Durante el período de mantenimiento, Amazon RDS iniciará automáticamente las instancias que hayan estado detenidas durante más de siete días para aplicar el mantenimiento. Amazon RDS no detendrá la instancia una vez que se complete el evento de mantenimiento.

La solución le permite especificar si desea añadir el período de mantenimiento preferido de una instancia de base de datos de RDS como período de ejecución a su programación. La solución iniciará la instancia al principio del período de mantenimiento y la detendrá al final del período de mantenimiento si ningún otro período de ejecución especifica que la instancia debe ejecutarse y si el evento de mantenimiento se ha completado.

Si el evento de mantenimiento no se completa al final del período de mantenimiento, la instancia se ejecutará hasta el intervalo de programación posterior a la finalización del evento de mantenimiento. Para obtener más información sobre el período de mantenimiento de Amazon RDS, consulte Mantenimiento de [una instancia](#) de base de datos en la Guía del usuario de Amazon RDS.

## Grupos de Amazon EC2 Auto Scaling

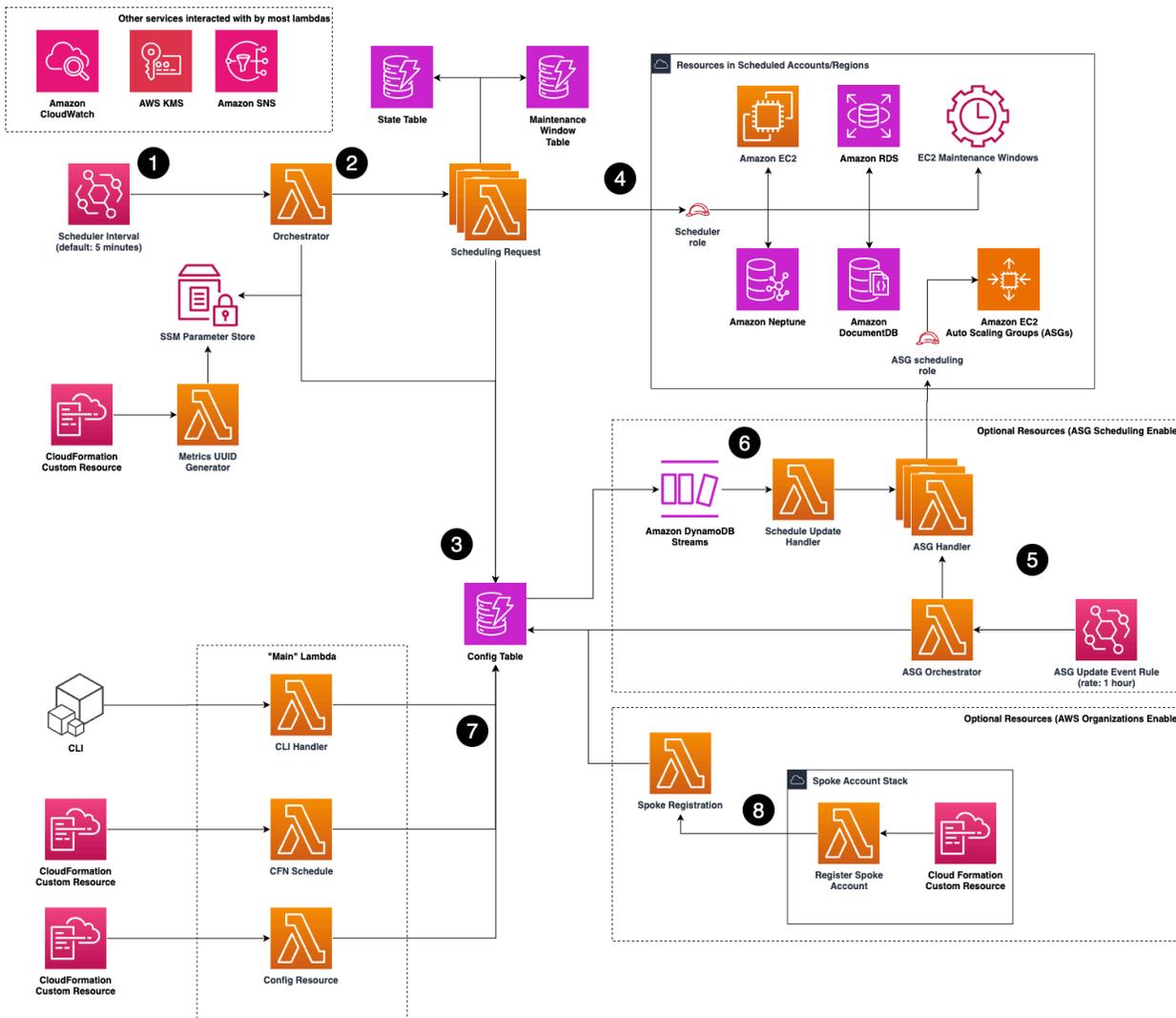
Diseñamos esta solución para detener automáticamente los grupos de Amazon EC2 Auto Scaling mediante acciones de escalado programadas. Puede usar la solución para configurar acciones de escalado programadas en el grupo Auto Scaling (ASG). Cuando una acción de escalado programada detiene un ASG, sus capacidades mínima, deseada y máxima se establecerán a 0 hasta que el ASG se vuelva a iniciar automáticamente. 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, AWS consideraciones sobre el [diseño de Well-Architected](#), [componentes de seguridad](#), [configuraciones de planificadores AWS](#) y [servicios](#) 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 AWS

1. La solución implementa una [Amazon EventBridge Rule](#) con un intervalo de programación configurable. Este intervalo de programación define la frecuencia con la que se ejecuta la solución y toma medidas para programar las instancias.
2. Cada intervalo de programación invoca una función de [AWS Lambda](#) orquestación. Esto determinará la lista de Cuentas de AWS regiones y servicios que deben programarse. A continuación, el intervalo de programación invocará varias funciones Lambda de solicitud de programación en paralelo para realizar las actividades de programación.
3. Se almacena un conjunto de programas y períodos en una tabla de configuración de [Amazon DynamoDB](#) para controlar el comportamiento de programación de la solución. Puede configurar cualquier número de programas/períodos en esta tabla y la solución programará las instancias en consecuencia.
4. Cada solicitud de programación inspecciona los recursos de su destino concreto (account/Region/service) para buscar los recursos que se hayan etiquetado para su programación mediante los cronogramas definidos en la tabla de configuración de la solución. A continuación, el gestor de solicitudes de programación inspecciona las programaciones configuradas y lleva a cabo las acciones de programación necesarias.
5. Si la programación ASG está habilitada, Instance Scheduler on AWS implementa una [EventBridge regla de Amazon](#) por hora y los recursos de orquestación y manejo asociados para administrar [las acciones de escalado programadas para los grupos de Amazon EC2](#) [Auto Scaling](#) que la solución ha etiquetado para su programación.
6. Además del análisis horario, la solución también realiza un seguimiento de las actualizaciones de las programaciones en la tabla de configuración. Cuando se actualiza una programación, se invoca una función Lambda de orquestación secundaria para garantizar que las acciones de escalado programadas de ASG se mantengan actualizadas con la configuración de programación más reciente.
7. La solución proporciona varias formas de crear o actualizar programaciones en la tabla de configuración de la solución, junto con varios ejemplos de programaciones para utilizarlas como punto de partida. [Los métodos de configuración incluyen: la consola DynamoDB, una CLI del programador y un recurso personalizado.](#) [AWS CloudFormation](#)
8. Si el modo AWS Orgs está activado y se proporciona un identificador de organización válido al implementar la solución, Instance Scheduler on registra AWS automáticamente las pilas de radios recién implementadas en la pila central de soluciones. [Los módulos hub y spoke se deben implementar en la misma región y en cuentas que sean miembros de la misma organización.](#) [AWS](#)

**Note**

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

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

Cada vez que la solución realiza un intervalo de programación, comprueba el estado actual de cada instancia debidamente etiquetada comparándolo con el estado objetivo (definido por uno o más [períodos](#) de un cronograma en la etiqueta de la instancia) del cronograma asociado. A continuación, el intervalo de programación aplica la acción de inicio o parada adecuada, según sea necesario.

Por ejemplo, si la función Lambda se invoca un viernes a las 9 a. m. (ET) e identifica una instancia de base de datos parada EC2 o de RDS con la etiqueta `schedule=Office-hours`, comprobará en Amazon DynamoDB los detalles de configuración del horario de oficina. Si el horario de oficina contiene un período que indica que la instancia debe ejecutarse de lunes a viernes de 9 a. m. ET a 5 p. m. ET, la función Lambda iniciará esa instancia.

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

**Note**

Detener una EC2 instancia de Amazon no es lo mismo que terminar una EC2 instancia de Amazon. De forma predeterminada, EC2 las instancias de Amazon están configuradas para detenerse, no para terminarse, cuando se cierran, pero puedes modificar este comportamiento. Antes de usar esta solución, compruebe que las instancias estén configuradas para detenerse o terminar según corresponda.

# AWS Consideraciones sobre el diseño de Well-Architected

Diseñamos esta solución con las mejores prácticas del [AWS Well-Architected](#) Framework, 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 Framework 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 Lambda.
- Utiliza Amazon SNS para informar 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 funciones de IAM.
- Todas las comunicaciones entre varias cuentas utilizan funciones de IAM.
- Todas las funciones que utiliza la solución se basan en 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, tiene 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 AWS servicios sin servidor siempre que es posible (como Lambda y DynamoDB) para garantizar una alta disponibilidad y recuperación en caso de fallo del servicio.
- El procesamiento de datos utiliza funciones 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 dispositivo Región de AWS que admita los AWS servicios utilizados en esta solución (como Lambda y DynamoDB). [Para obtener más información, consulte Compatible. Regiones de AWS](#)
- La solución se prueba e implementa automáticamente todos los días. Nuestros arquitectos de soluciones y expertos en la materia revisan la solución en busca de áreas en las que experimentar y mejorar.

## 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 servidores y los clientes solo pagan por lo que utilizan.
- El valor predeterminado de la capa de cómputo es Lambda, que usa pay-per-use un modelo.

## 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 gestionados y sin servidor para minimizar el impacto ambiental de los servicios de backend.

- El diseño sin servidores de la solución tiene como objetivo reducir la huella de carbono en comparación con la huella de los servidores locales que funcionan continuamente.

## Tabla de configuración del programador

Cuando se implementa, Instance Scheduler on AWS crea una tabla de Amazon DynamoDB que contiene los ajustes de configuración global.

Los elementos de configuración global contienen un atributo de tipo con el valor config en la tabla de configuración. Los cronogramas y los períodos contienen atributos de tipo con los valores de cronograma y período, respectivamente. Puede añadir, actualizar o eliminar programas y períodos de la tabla de configuración mediante la consola de DynamoDB o la interfaz de línea de comandos [de](#) la solución. Sin embargo, no se edita ningún elemento con un tipo de configuración porque estos elementos los administra la solución.

## Programador CLI

La solución incluye una CLI que proporciona comandos para configurar horarios y períodos. La CLI le permite estimar los ahorros de costos para un programa determinado. Las estimaciones de costos proporcionadas por la CLI del cronograma son únicamente con fines aproximativos. Para obtener más información sobre la configuración y el uso de la CLI del programador, consulte la [CLI del programador](#).

## AWS servicios utilizados en esta solución

AWS servicio	Descripción
<a href="#">AWS Lambda</a>	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.

AWS servicio	Descripción
<a href="#">Amazon DynamoDB</a>	Principal. La solución crea tablas de DynamoDB para almacenar la configuración de la programación, la información de estado, las últimas acciones realizadas en las instancias y una tabla para almacenar el período de mantenimiento de Systems Manager con fines de programación.
<a href="#">Amazon CloudWatch</a>	Principal. La solución almacena los registros de depuración e información.
<a href="#">AWS IAM</a>	Principal. La solución usa IAM para obtener permisos para programar instancias.
<a href="#">Amazon SNS</a>	Principal. La solución crea un tema de SNS para enviar mensajes de error para que los usuarios se suscriban y solucionen los problemas en caso de que se produzca algún error.
<a href="#">AWS KMS</a>	Principal. La solución crea una AWS KMS clave para cifrar el tema de SNS.
<a href="#">Amazon EventBridge</a>	Principal. La solución crea una EventBridge solución: crea reglas EventBridge programadas que invocan AWS lambda en un intervalo constante»
<a href="#">AWS Systems Manager</a>	Admite. Proporciona monitoreo de recursos a nivel de aplicación y visualización de las operaciones de los recursos y los datos de costos.

AWS servicio	Descripción
<a href="#">Amazon EC2</a>	Programado. La solución se utiliza para iniciar y detener EC2 las instancias. Las instancias se identifican mediante etiquetas, clave/valores específicos que se configuran en la solución.
<a href="#">Amazon RDS</a>	Programado. La solución se utiliza para cambiar el estado de las instancias de base de datos de RDS a Disponibles o Detenidas. Las instancias se identifican mediante etiquetas, clave/valores específicos que se configuran en la solución.
<a href="#">Amazon Aurora</a>	Programado. La solución se utiliza para cambiar el estado de los clústeres de Aurora a Disponible o Detenido. Los clústeres se identifican mediante etiquetas clave/valores específicos que se configuran en la solución.
<a href="#">Amazon Neptune</a>	Programado. La solución se utiliza para cambiar el estado de las instancias de Neptune a Disponible o Detenido. Las instancias se identifican mediante etiquetas, clave/valores específicos que se configuran en la solución.
<a href="#">Amazon DocumentDB</a>	Programado. La solución se utiliza para cambiar el estado de las instancias de DocumentDB a Disponible o Detenido. Las instancias se identifican mediante etiquetas clave/valores específicos que se configuran en la solución.

AWS servicio	Descripción
<a href="#">Grupos de Amazon EC2 Auto Scaling</a>	Programado. La solución se usa para administrar las reglas de escalado programadas para los grupos EC2 de Auto Scaling. Estas reglas son las start/stop Auto Scaling groups in accordance with an associated schedule. Groups are identified by specific tags key/values que están configuradas en la solución.

## Seguridad

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

### AWS KMS

La solución crea una clave AWS gestionada por el cliente, que se utiliza para configurar el cifrado del lado 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 RDS get/put Systems Manager parameters, access to CloudWatch log groups, AWS KMS key encryption/decryption, and publish messages to SNS. In addition, Instance Scheduler on AWS will also create Scheduling Roles in all managed accounts that will provide access to start/stop EC2, a los recursos de escalado automático, a las instancias de base de datos, para modificar los atributos de las instancias y actualizar las etiquetas de esos recursos. La solución al rol de servicio Lambda creado como parte de la plantilla de solución proporciona todos los permisos necesarios.

Durante la implementación, el programador de instancias en AWS implementará funciones de IAM específicas para cada una de sus funciones de Lambda, junto con funciones de programador que solo se pueden asumir mediante Lambdas de programación específicas en la plantilla del hub

implementado. Estos roles programados tendrán nombres que sigan el patrón, y. {namespace}-Scheduler-Role {namespace}-ASG-Scheduling-Role

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

## Volúmenes de EC2 EBS cifrados

Al programar EC2 instancias adjuntas a volúmenes de EBS cifrados por AWS KMS, debe conceder AWS permiso a Instance Scheduler para usar las AWS KMS claves asociadas. Esto permite EC2 a Amazon descifrar los volúmenes de EBS adjuntos durante la función iniciada. Este permiso debe concederse a la función de programación de la misma cuenta que las EC2 instancias que utilizan la clave.

Para conceder permiso para usar una AWS KMS clave con el programador de instancias activado AWS, añada el ARN de la AWS KMS clave al programador AWS de instancias de la pila (central o radial) de la misma cuenta que EC2 las instancias que utilizan las claves:

### 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*

## Armas clave KMS para EC2

Esto generará automáticamente la siguiente política y la agregará a la función de programación de 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"
}
]
}
```

# Introducción

Esta guía contiene una breve descripción general e instrucciones para implementar la solución rápidamente. Esta solución utiliza [plantillas y pilas de AWS CloudFormation](#) para automatizar su implementación. Las CloudFormation plantillas especifican los AWS recursos 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 anónimas a AWS. Usamos estos datos para comprender mejor cómo los clientes utilizan esta solución y los servicios y productos relacionados. AWS es propietario de los datos recopilados a través de esta encuesta. La recopilación de datos está sujeta al [aviso de privacidad](#).

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

Siga las step-by-step instrucciones de esta sección para configurar e implementar la solución en su cuenta.

Tiempo de implementación: aproximadamente de 5 a 10 minutos (sin incluir la configuración).

### [Paso 1: lanza la pila de planificadores de instancias](#)

1. Lanza la AWS CloudFormation plantilla en tu. Cuenta de AWS
2. Introduzca los valores de los parámetros necesarios.
3. Revise el resto de los parámetros de la plantilla y ajústelos si es necesario.

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

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

2. Introduzca los valores de los parámetros necesarios.

## AWS CloudFormation plantillas

Esta solución utiliza [plantillas y pilas de AWS CloudFormation](#) para automatizar su implementación. Las CloudFormation plantillas especifican los AWS recursos 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 AWS Lambda función, 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 multicuenta que utiliza la solución para programar instancias en cuentas informales. En el caso de las implementaciones que se utilizan AWS Organizations, la implementación de la plantilla también registra la cuenta radial en el hub, lo que no requiere ninguna configuración manual.

### Note

Si ya implementó esta solución anteriormente, consulte [Actualizar la solución](#) para obtener instrucciones de actualización.

## Paso 1: lanza la pila central del programador de instancias

Siga las step-by-step instrucciones de esta sección para implementar la solución en su cuenta.

Tiempo de implementación: aproximadamente cinco minutos



1. Inicie sesión [AWS Management Console](#) y seleccione el botón para iniciar la `instance-scheduler-on-aws` plantilla AWS CloudFormation `.template`.
2. La plantilla se lanza en la región Este de EE. UU. (Norte de Virginia) de forma predeterminada. Para lanzar la solución en otro lugar Región de AWS, utilice el selector de regiones de la barra de navegación de la consola.
3. En la página Crear pila, compruebe que la URL de la plantilla correcta esté en el cuadro de texto URL de Amazon S3 y seleccione Siguiente.
4. En la página Especificar los detalles de la pila, especifique un nombre para la pila. Para obtener información sobre las limitaciones de caracteres en los nombres, consulte el [IAM y AWS STS las cuotas](#) en la Guía del AWS Identity and Access Management usuario.
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/a
Clave de etiqueta de programación	Schedule
Intervalo de programación (minutos)	5

Parámetro	Predeterminado/a
Zona horaria predeterminada	UTC
Programación habilitada	Yes
Habilite la programación xxx	Enabled
Etiquetas de inicio	InstanceScheduler-LastA n=Started By {scheduler {year}/{month}/{day} {H inute}{timezone},>
Etiquetas de parada	InstanceScheduler-LastA n=Stopped By {scheduler {year}/{month}/{day} {H inute}{timezone},>
Habilite las EC2 ventanas de mantenimiento de SSM	No
Clave KMS para ARNs EC2	<Optional Input>
Cree instantáneas de instancias de RDS al detener	No

Parámetro	Predeterminado/a
Clave de etiqueta programada de ASG	scheduled
Prefijo del nombre de la acción ASG	is-
Utilice AWS Organizations	No
Espacio de nombres	default
ID de organización o cuenta remota IDs	<Optional Input>
Región (es)	<Optional Input>
Habilitada la programación de cuentas centrales	Yes
Periodo de retención de registros (días)	30
Habilite los CloudWatch registros de depuración	No

Parámetro	Predeterminado/a
Supervisión operativa	Enabled
Tamaño de la memoria	128
Proteja las tablas de DynamoDB	Enabled

6. Elija Next (Siguiete).
7. En la página Configurar opciones de pila, elija Siguiete.
8. En la página Revisar y crear, revise y confirme la configuración. Marque la casilla para confirmar que la plantilla creará recursos de IAM.
9. Elija Crear para implementar la pila.

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

## Paso 2 (opcional): lanza la pila remota en las cuentas secundarias

### Important

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

Esta AWS CloudFormation plantilla 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 pila principal/central se haya instalado correctamente en la cuenta de Hub.



1. Inicia sesión en AWS Management Console la cuenta secundaria correspondiente y selecciona el botón para iniciar la plantilla instance-scheduler-on-aws AWS CloudFormation -remote.
2. La plantilla se lanza en la región Este de EE. UU. (Norte de Virginia) de forma predeterminada. Para lanzar la solución en otro lugar Región de AWS, utilice el selector de regiones de la barra de navegación de la consola. Si el hub stack está configurado para usarse AWS Organizations, despliegue la plantilla remota en la misma región que el hub stack.
3. En la página Crear pila, compruebe que la URL de la plantilla correcta esté en el cuadro de texto URL de Amazon S3 y seleccione 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 AWS Organizations opción 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 AWS Organización está establecida en No, el hub stack debe actualizarse con el nuevo ID de cuenta.

Parámetro	Predeterminado/a	Descripción
ID de cuenta de Hub	<Requires Input>	ID de cuenta del programador de instancias de la pila AWS central que programará los recursos de esta cuenta.
Usa AWS Organizations	No	Úselo AWS Organizations para automatizar el registro de cuentas Spoke. Debe tener el mismo valor que el hub stack.
Espacio de nombres	default	Identificador único que se utiliza para diferenciar entre múltiples implementaciones de soluciones. Debe configurarse

Parámetro	Predeterminado/a	Descripción
		con el mismo valor que la pila central.
Clave Kms ARNs para EC2	<Entrada opcional>	Lista de KMS separados por comas ARNs para conceder a la solución kms: CreateGrant los permisos de descifrado de los EC2 volúmenes de EBS cifrados al servicio. Esto permite al programador iniciar EC2 instancias con volúmenes de EBS cifrados adjuntos. Indique (*) para dar acceso limitado a todas las claves de KMS; déjelo en blanco para deshabilitarlo. Para obtener más información sobre la política exacta creada, consulte <a href="#">Volúmenes cifrados de EC2 EBS</a>

5. Elija Next (Siguiente).
6. En la página Opciones, seleccione Siguiente.
7. En la página Revisar y crear, revise y confirme la configuración. Asegúrese de marcar la casilla para confirmar que la plantilla creará recursos de IAM.
8. Elija Crear para implementar la pila.

Puede ver el estado de la pila en la AWS CloudFormation consola, en la columna Estado. Deberías ver el estado CREATE\_COMPLETE en aproximadamente cinco minutos.

## Configure la solución

Ahora que la solución se ha implementado, puede empezar a configurar los horarios y a etiquetar las instancias para el programador. Para obtener más información sobre cómo hacer estas cosas, consulte [Configurar los horarios](#) y [etiquetar las 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](#), [monitorear la solución](#), [actualizarla](#) y otras [funciones avanzadas](#).

## Configure los horarios

Una vez que la solución se haya implementado correctamente, puede empezar a configurar los horarios. Instance Scheduler on AWS admite dos métodos de administración de programaciones, tal como se describe a continuación.

### Note

La solución puede admitir cualquier número de programas, cada uno de los cuales puede contener uno o más períodos que definen cuándo deben ejecutarse las instancias controladas por ese cronograma. Para obtener más información, consulte [Horarios](#) y [períodos](#).

## Uso de la infraestructura como código (recomendado)

Instance Scheduler está activado y AWS CloudFormation CustomResource le AWS permite administrar sus horarios y períodos mediante la infraestructura como código (IaC).

Para obtener información sobre cómo gestionar los horarios mediante el IaC, consulte [Gestionar los horarios mediante la infraestructura como código](#) (IaC).

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

### Important

Si ha utilizado el recurso personalizado para gestionar cualquier programación mediante IaC, no debe utilizar la consola de DynamoDB ni la CLI del programador para eliminar o modificar esas programaciones o sus períodos. Si lo hace, creará 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 el AWS hub stack, la solución creó una tabla de Amazon DynamoDB que contiene varios ejemplos de períodos y programas que puede usar como referencia para crear sus propios períodos y programas 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.](#)

Esta sección proporciona instrucciones y referencias sobre cómo usar, monitorear y actualizar la solución, así como información de solución de problemas y soporte.

## Etiquete las instancias para programarlas

Al implementar la AWS CloudFormation plantilla, definió el nombre (clave de etiqueta) para la etiqueta personalizada de la solución. Para que Instance Scheduler AWS esté activado para reconocer 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 y Etiquetar EC2 los recursos de Amazon RDS](#).

En AWS Management Console, utilice el [editor de etiquetas](#) para aplicar o modificar etiquetas para varios recursos a la vez. También puede aplicar y modificar las etiquetas manualmente en la consola.

## Establecer el valor de la 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 la

etiqueta con el nombre del programa que debe aplicarse a la instancia. Si desea cambiar la clave de etiqueta, puede hacerlo [actualizando los parámetros de la solución](#).

#### Note

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

## EC2 instancias con volúmenes de EBS cifrados

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

## Referencia del cronograma

Los cronogramas especifican cuándo deben ejecutarse las instancias etiquetadas con ese cronograma. Cada programa debe tener un nombre único, que se utiliza como valor de etiqueta que identifica el programa que desea aplicar al recurso etiquetado.

### Periodos

Cada programación debe contener al menos un período que defina las horas en las que debe ejecutarse la instancia. Una programación puede contener más de un período. Cuando se utilice más de un período en una programación, Instance Scheduler AWS activado aplicará la acción de inicio adecuada cuando se cumpla al menos uno de los períodos. Para obtener más información, consulte la [referencia del período](#).

### Zona horaria

También puede especificar una zona horaria para la programación. Si no especifica una zona horaria, la programación utilizará la zona horaria predeterminada que especifique al lanzar la solución. Para obtener una lista de los valores de zona horaria aceptables, consulte la columna TZ de la [lista de zonas horarias de la base de datos TZ](#).

## Campo Hibernar

El campo hibernar te permite usar la hibernación para las instancias de Amazon EC2 detenidas. Si este campo está establecido en true, EC2 las instancias deben usar una Amazon Machine Image (AMI) que admita la hibernación. Para obtener más información, consulta [Linux AMIs](#) y [Windows compatibles AMIs](#) en la Guía del EC2 usuario de Amazon. 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 está establecido en true, las instancias pasan a hibernación en lugar de detenerse cuando la solución las detiene.

Si configuras la solución para que utilice la hibernación, pero las instancias no están [configuradas para la hibernación](#) o no cumplen los [requisitos previos de 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 EC2 usuario de Amazon.

## Campo obligatorio

Las planificaciones contienen un campo obligatorio que permite evitar que una instancia se inicie manualmente fuera de un período de ejecución o que se detenga manualmente durante un período de ejecución. Si este campo se establece en true y un usuario inicia manualmente una instancia fuera de un período de ejecución, la solución detendrá la instancia. Si este campo se establece en verdadero, también reinicia una instancia si se detuvo manualmente durante un período de ejecución.

## Conserva el campo en ejecución

El campo retain\_running impide que la solución detenga una instancia al final de un período de ejecución si la instancia se inició manualmente antes del comienzo del período. Por ejemplo, si una instancia con un período que va de las 9 a. m. a las 5 p. m., se inicia manualmente antes de las 9 a. m., la solución no detendrá la instancia a las 5 p. m.

## Campo de ventana de mantenimiento de Systems Manager (solo se aplica a EC2 las instancias)

El ssm-maintenance-window campo le permite añadir automáticamente las ventanas de mantenimiento de AWS Systems Manager como períodos de ejecución a una programación. Cuando especificas el nombre de un período de mantenimiento que existe en la misma cuenta y Región de

AWS que tus EC2 instancias de Amazon, la solución iniciará la instancia al menos 10 minutos antes del inicio del período de mantenimiento y la detendrá al final del período de mantenimiento si no hay otro período de ejecución que especifique que la instancia debe ejecutarse.

Una vez creada la ventana de mantenimiento de SSM y configurada la programación con el nombre de la ventana de mantenimiento de SSM, los cambios se recogen en la siguiente ejecución programada de la Lambda. Por ejemplo, si ha seleccionado una frecuencia de 5 minutos para que se ejecute el programador Lambda, Lambda recogerá los cambios en el período de mantenimiento en el siguiente intervalo de 5 minutos.

El programador de instancias activado AWS garantizará que las instancias se inicien al menos 10 minutos antes de que comience el período de mantenimiento. Según el valor que establezcas para el AWS CloudFormation parámetro Intervalo de programación, es posible 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 estableces el intervalo de programación en 30 minutos, el programador iniciará la instancia entre 10 y 40 minutos antes del inicio del período de mantenimiento.

#### Note

Para 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 [Windows de mantenimiento de AWS Systems Manager](#) en la guía del usuario de AWS Systems Manager.

## Tipo de instancia

Solo para EC2 las instancias de Amazon, una programación te permite especificar el tipo de instancia opcional que desees para cada período de una programación. Al especificar un tipo de instancia en el período, la solución cambiará automáticamente el tamaño de las EC2 instancias para que coincidan con el tipo de instancia solicitado.

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

Si el tipo de instancia de una instancia en ejecución es diferente del 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, consulta [Cambiar el tipo de instancia](#) en la Guía del EC2 usuario de Amazon para instancias de Linux.

## Definiciones de programación

El programador de instancias de la tabla AWS de configuración de Amazon DynamoDB contiene las definiciones de programación. Una definición de cronograma puede contener los siguientes campos:

Campo	Descripción
<code>description</code>	Una descripción opcional de la programación.
<code>hibernate</code>	Elija si desea hibernar las EC2 instancias de Amazon que ejecutan Amazon Linux. Si este campo se establece en true, el programador hibernará las instancias cuando las detenga. <a href="#">Ten en cuenta que las instancias deben activar la hibernación y cumplir los requisitos previos de hibernación.</a>
<code>enforced</code>	Elija si desea hacer cumplir el cronograma. Si este campo se establece en true, el programador detendrá una instancia en ejecución si se inicia manualmente fuera del período de ejecución o iniciará una instancia si se detiene manualmente durante el período de ejecución.
<code>name</code>	El nombre utilizado para identificar la programación. Este nombre debe ser único e incluir únicamente caracteres alfanuméricos, guiones (-) y guiones bajos (_).
<code>periods</code>	El nombre de los puntos que se utilizan en este programa. Introduzca los nombres exactamente como aparecen en el campo del nombre del período.

Campo	Descripción
	También puede especificar un tipo de instancia para el período mediante la sintaxis @ <period-name><instance-type>. Por ejemplo, weekdays@t2.large.
retain_running	Elija si desea evitar que la solución detenga una instancia al final de un período de ejecución si la instancia se inició manualmente antes del comienzo del período.
ssm_maintenance_window	<p>Elija si desea añadir las ventanas de mantenimiento de AWS Systems Manager como período de ejecución adicional para este programa. Acepta StringSet los nombres de las ventanas de mantenimiento que se compararán con los nombres de las ventanas de la misma cuenta o región que las instancias programadas EC2.</p> <p>Nota: Esta función solo se aplica a las EC2 instancias.</p>
stop_new_instances	Elija si desea detener una instancia la primera vez que se etiqueta si se está ejecutando fuera del período de ejecución. De forma predeterminada, este campo está establecido en true.
timezone	La zona horaria que utilizará la programación. Si no se especifica ninguna zona horaria, se utiliza la zona horaria predeterminada (UTC). Para obtener una lista de los valores de zona horaria aceptables, consulte la columna TZ de la <a href="#">lista de zonas horarias de la base de datos tz</a> .

Campo	Descripción
<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 cobrarán 0,90 USD al mes por programa o servicio programado.</p>

## Período de referencia

Los períodos contienen condiciones que te 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 las condiciones deben cumplirse para que el programador de instancias pueda AWS aplicar la acción de inicio o parada adecuada.

## Horas de inicio y finalización

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

Del mismo modo, si solo especificas una hora de parada, la instancia debe iniciarse manualmente. Si no especificas ninguna hora, esta solución utiliza las reglas de los días de la semana, los días del mes o los meses para iniciar y detener las instancias al principio o al final de cada día, según corresponda.

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

Si su programación contiene varios períodos, le recomendamos que especifique siempre a `begintime` y `endtime` en sus períodos.

Si inicias 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 de la mañana y se detenga a las 5 de la tarde.



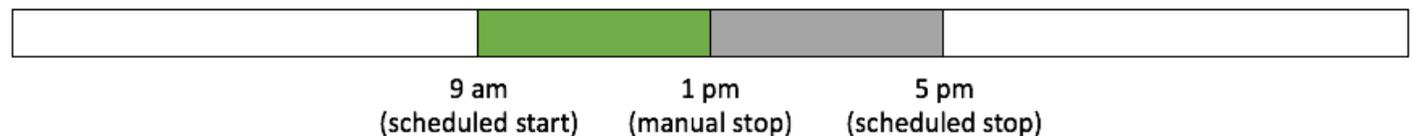
### 9-5 horas de inicio y parada programadas

Si inicia esa instancia manualmente a las 5 de la mañana, la solución detendrá la instancia a las 5 de la tarde. Si utiliza el [campo de retención en ejecución](#), la solución no detendrá la instancia a las 5 p. m.



### Parada programada a las 5 de la mañana

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



### Parada programada a las 5 p. m.

### Periodos adyacentes

La solución no detendrá la ejecución de instancias si la programación contiene dos períodos de ejecución adyacentes. Por ejemplo, si tiene una programación con un periodo a `endtime` las 23:59

horas y otro a medianoche `begin_time` del día siguiente, la solución no detendrá la ejecución de las instancias si no hay `weekdays`, `monthdays`, or `months` reglas que las detengan.

Para implementar un programa que ejecute las instancias desde las 9 de la mañana del lunes hasta las 5 de la tarde del viernes, la solución requiere tres períodos. El primer período ejecuta las instancias aplicables desde las 9:00 a. m. hasta las 23:59 p. m. del lunes. El segundo período ejecuta las instancias desde la medianoche del martes hasta las 23:59 horas del jueves. El tercer período ejecuta las instancias desde la medianoche del viernes hasta las 17.00 horas del viernes. Para obtener más información, consulte un [ejemplo de programa](#).

## Días de la semana

El `weekdays` campo define los días de la semana en que se ejecutará una instancia. Puede especificar una lista de días, un rango de días, la <sup>enésima</sup> aparición de ese día de un mes o la última vez que se produce ese día de un mes. La solución admite nombres abreviados de días (lunes) y números (0).

## Días del mes

El campo `monthdays`, define los días del mes en que se ejecutará una instancia. Puede especificar una lista de días, un rango de días, todos los días del mes, el último día del mes o el día de la semana más cercano a una fecha específica.

## Meses

El `months` campo define los meses en los que se ejecutará una instancia. Puede especificar una lista de meses, un rango de meses o todos <sup>los</sup> meses. La solución admite nombres de meses abreviados (enero) y números (1).

## Definiciones de períodos

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

### Important

Debe especificar al menos uno de los siguientes elementos: hora de inicio, hora de finalización, días de la semana, meses o días del mes.

Campo	Descripción
<code>begintime</code>	La hora, en formato HH:MM, a la que se iniciará la instancia.
<code>description</code>	Una descripción opcional del período.
<code>endtime</code>	La hora, en formato HH:MM, en la que se detendrá la instancia.
<code>months</code>	<p>Introduce una lista de meses delimitados por comas, o un rango de meses separados por guiones, durante los que se ejecutará la instancia. Por ejemplo, introduce <code>jan, feb, mar</code> o <code>1, 2, 3</code> ejecuta una instancia durante esos meses. O bien, puede escribir <code>jan-mar</code> o <code>1-3</code>.</p> <p>También puedes programar una instancia para que se ejecute cada <sup>mes</sup> o cada <sup>mes</sup> dentro de un rango. Por ejemplo, introduce <code>Jan/3</code> o <code>1/3</code> ejecuta una instancia cada tres meses a partir de enero. Introduzca esta <code>Jan-Jul/2</code> opción para que se ejecute cada dos meses, de enero a julio.</p>
<code>monthdays</code>	<p>Introduce una lista de días del mes delimitados por comas, o un intervalo de días separados por guiones, durante los que se ejecutará la instancia. Por ejemplo, introduce <code>1, 2, 3</code> o <code>1-3</code> ejecuta una instancia durante los tres primeros días del mes. También puede introducir varios rangos. Por ejemplo <code>1-3, 7-9</code> para ejecutar una instancia <sup>del</sup> <code>1 al 3</code> y <sup>del</sup> <code>7 al 9</code>.</p> <p>También puedes programar una instancia para que se ejecute todos los días del mes o todos los días del mes de un rango. Por ejemplo, introduce</p>

Campo	Descripción
	<p>1/7 ejecutar una instancia cada siete días a partir del día <sup>1</sup>. Introduzca 1-15/2 ejecutar una instancia cada dos días <sup>del 1 al 15</sup>.</p> <p>Introduzca L esta opción para ejecutar una instancia el último día del mes. Introduce una fecha y una W para ejecutar una instancia el día de la semana más cercano a la fecha especificada. Por ejemplo, introduce 15W ejecutar una instancia el día de la semana más cercano al día 15.</p>
name	<p>El nombre utilizado para identificar el período. Este nombre debe ser único e incluir únicamente caracteres alfanuméricos, guiones (-) y guiones bajos (_).</p>
weekdays	<p>Introduce una lista delimitada por comas de los días de la semana, o un rango de días de la semana, durante los que se ejecutará la instancia. Por ejemplo, introduce 0, 1, 2 o ejecuta una instancia de lunes 0-2 a miércoles. También puede introducir varios rangos. Por ejemplo, introduzca 0-2, 4-6 para ejecutar una instancia todos los días excepto los jueves.</p> <p>También puedes programar una instancia para que se ejecute cada día de la semana del mes. Por ejemplo, introduce Mon#1 o 0#1 ejecuta una instancia el primer lunes del mes.</p> <p>Introduce un día y una L para ejecutar una instancia el último día de la semana del mes. Por ejemplo, introduce friL o 4L ejecuta una instancia el último viernes del mes.</p>

Cuando un período contiene varias condiciones, tenga en cuenta que todas las condiciones deben cumplirse para que Instance Scheduler pueda AWS aplicar la acción adecuada. Por ejemplo, un período que contenga un weekdays campo con un valor de Mon#1 y un campo de meses con un valor de Jan/3 aplicará la acción el primer lunes del trimestre.

## Etiquetado automatizado

El programador de instancias activado AWS puede añadir etiquetas automáticamente a todas las instancias que inicie o detenga. Puede especificar una lista de nombres o tagname=tagvalue pares de etiquetas en los parámetros Etiquetas iniciadas y Etiquetas detenidas. La solución también incluye macros que permiten añadir información variable a las etiquetas:

- {scheduler}: El nombre de la pila de planificadores
- {year}: El año (cuatro dígitos)
- {month}: El mes (dos dígitos)
- {day}: El día (dos dígitos)
- {hour}: La hora (dos dígitos, formato de 24 horas)
- {minute}: El minuto (dos dígitos)
- {timezone}: La zona horaria

En la siguiente tabla se muestran ejemplos de diferentes entradas y las etiquetas resultantes.

Ejemplo de entrada de parámetros	Etiqueta Instance Scheduler
ScheduleMessage=Started by scheduler {scheduler}	ScheduleMessage=Started by scheduler MyScheduler
ScheduleMessage=Started on {year}/{month}/{day}	ScheduleMessage=Started on 2017/07/06
ScheduleMessage=Started on {year}/{month}/{day} at {hour}:{minute}	ScheduleMessage=Started on 2017/07/06 at 09:00

Ejemplo de entrada de parámetros	Etiqueta Instance Scheduler
ScheduleMessage=Started on {year}/{month}/{day} at {hour}:{minute} {timezone}	ScheduleMessage=Started on 2017/07/06 at 09:00 UTC

Al utilizar el parámetro Etiquetas iniciadas, las etiquetas se eliminan automáticamente cuando el planificador detiene la instancia. Al utilizar el parámetro Etiquetas detenidas, las etiquetas se eliminan automáticamente cuando se inicia la instancia.

## Ejemplos de horarios

Instance Scheduler on AWS le permite iniciar y detener automáticamente las instancias de Amazon Elastic Compute Cloud (Amazon EC2) y Amazon Relational Database Service (Amazon RDS). En la siguiente sección, se proporcionan algunos ejemplos de cronogramas que se pueden adaptar a muchos casos de uso habituales.

### De 9 a 5 horas de trabajo estándar

Este programa muestra cómo ejecutar las instancias de lunes a viernes, de 9 a. m. a 5 p. m., en Londres.

#### Periodos

Durante este período, las instancias comenzarán a las 9 a. m. y las finalizarán a las 5 p. m., de lunes a viernes, de lunes a viernes.

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

## Programación

El nombre del programa 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, debes añadir la `Schedule=london-working-hours` etiqueta a las instancias. Si cambias el nombre de etiqueta predeterminado en el parámetro de nombre de etiqueta del Programador de AWS CloudFormation Instancias, tu etiqueta será diferente. Por ejemplo, si lo has introducido `Sked` como nombre de etiqueta, será `Sked=london-working-hours` tu etiqueta. Para obtener más información, consulte [Etiquete sus recursos](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

### Programador CLI

Para configurar el programa 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 la programación anterior utilizando el [recurso personalizado de programación](#).

Para implementar esta plantilla, tendrá que proporcionar el `ServiceInstanceScheduleServiceTokenARN` que se encuentra en la AWS CloudFormation consola. 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
```

## Detenga las instancias después de las 5 p.m.

Las instancias se pueden iniciar libremente en cualquier momento del día y este programa garantizará que se les envíe automáticamente un comando de parada todos los días a las 5 p. m., hora del este.

### Periodos

Durante este período, las instancias se detendrán todos los días a las 17:00 horas.

Campo	Tipo	Valor
endtime	String	16:59

Campo	Tipo	Valor
name	String	stop-at-5

## Programación

El nombre del programa 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
periods	StringSet	stop-at-5
timezone	String	America/New York

## Etiqueta de instancia

Para aplicar esta programación a las instancias, debes añadir la `Schedule=stop-at-5-new-york` etiqueta a las instancias. Si cambiaste el nombre de etiqueta predeterminado en el parámetro de nombre de etiqueta del Programador de AWS CloudFormation Instancias, tu etiqueta será diferente. Por ejemplo, si lo ingresó Sked como nombre de etiqueta, será `Sked=stop-at-5-new-york` su etiqueta. Para obtener más información, consulte [Etiquete sus recursos](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

## Programador CLI

Para configurar el programa 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 la programación anterior utilizando el [recurso personalizado de programación](#).

Para implementar esta plantilla, necesitará proporcionar el `ServiceInstanceScheduleServiceTokenARN` que se encuentra en la AWS CloudFormation consola. Para ello, haga clic en la [pila Instance Scheduler Hub 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'
```

## Detenga las instancias durante el fin de semana

Este programa muestra cómo ejecutar las instancias desde el lunes a las 9 a. m. hora del este hasta el viernes a las 5 p. m., hora del este. Como los lunes y viernes no son días completos, este programa incluye tres períodos para adaptarlos: lunes, martes a jueves y viernes.

### Periodos

El primer período comienza con las instancias etiquetadas a las 9 de la mañana del lunes y termina 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 durante todo el día de martes a jueves. Este período incluye los siguientes campos y valores.

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

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

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

## Programación

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

Campo		Valor
name	String	lunes, de 9 a.m. a viernes a 5 p.m.
periods	StringSet	lunes a partir de las 9 a.m., viernes a las 5 p.m. tue-thu-full-day
timezone	String	America/New_York

## Etiqueta de instancia

Para aplicar esta programación a las instancias, debes añadir la `Schedule=mon-9am-fri-5pm` etiqueta a las instancias. Tenga en cuenta que si ha cambiado el nombre de etiqueta predeterminado en el parámetro de nombre de etiqueta del Programador de AWS CloudFormation Instancias, su etiqueta será diferente. Por ejemplo, si ingresó `Sked` como nombre de etiqueta, será su etiqueta. `Sked=mon-9am-fri-5pm` Para obtener más información, consulte [Etiquete sus recursos](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

## Programador CLI

Para configurar el programa 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 la programación anterior utilizando el [recurso personalizado de programación](#).

Para implementar esta plantilla, tendrá que proporcionar el `ServiceInstanceScheduleServiceTokenARN` que se encuentra en la AWS CloudFormation consola. 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 de la solución

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

Nombre del recurso	Tipo	Descripción
Principal	AWS::Lambda::Function	El programador de instancias está en AWS Lambda funcionamiento.
Asistente de configuración de Scheduler	Custom::ServiceSetup	Almacena los ajustes de configuración global en Amazon DynamoDB.
Permiso de invocación del programador	AWS::Lambda::Permission	Permite que el CloudWatch evento de Amazon invoque la función del programador de AWS Lambda instancias.
Registros del programador	AWS::Logs::LogGroup	CloudWatch Grupo de registros para el programador de instancias.
Política del programador	AWS::IAM::Policy	Política que permite al programador realizar acciones de inicio y parada, cambiar los atributos de las EC2 instancias de Amazon, establecer etiquetas y acceder a los recursos del programador.
Regla del programador	AWS::Events::Rule	Regla de EventBridge eventos de Amazon que invoca la función Lambda del programador.
Regla de eventos de métricas de configuración	AWS::Events::Rule	Regla de EventBridge eventos de Amazon que invoca periódicamente la función de métricas anónimas de la descripción de la configuración. Se desactiva cuando las

Nombre del recurso	Tipo	Descripción
		métricas anonimizadas están deshabilitadas.
Tabla de estados	AWS::Dynamamodb::Table	Tabla de DynamoDB que almacena el último estado deseado de las instancias.
Config Table	AWS::Dynamamodb::Table	Tabla de DynamoDB que almacena datos globales de configuración, programación y período.
Tema SNS del programador de instancias	AWS::SNS::Topic	Envía mensajes de advertencia y error a las direcciones de correo electrónico suscritas.

## Programador CLI

El programador de instancias en la interfaz de línea de AWS comandos (CLI) le permite configurar programas y períodos, y estimar los ahorros de costos para un programa determinado.

## Requisitos previos

La CLI de esta solución requiere Python 3.8+ y la última versión de boto3.

## Credenciales

Para utilizar la CLI del programador, debe tener credenciales para AWS CLI. Para obtener más información, consulte los [ajustes de configuración y del archivo de credenciales](#) en la Guía del AWS CLI usuario.

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 AWS Lambda función de la pila para gestionar la solicitud de CLI

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

#### 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 del nombre del perfil, consulte Argumentos comunes.](#)

## Instale la CLI del programador

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

#### Important

La instalación fallará si no coloca los archivos en su propio directorio y, a continuación, los instala desde ese directorio.

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

#### Note

`Scheduler-cli` requiere Python 3.8 o superior y las últimas versiones de `pip` y `boto3`. Si no tiene todos estos instalados en su máquina local, consulte la [documentación oficial de pip para obtener las instrucciones de](#) instalación antes de intentar instalar el `Scheduler-CLI`.

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

5. Verifique que la instalación se haya realizado correctamente con:

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

### Note

Si lo prefiere, es un [sdist de la CLI](#) y se puede instalar mediante el mismo proceso que el anterior.

## Estructura de comandos

La CLI del programador utiliza una estructura multiparte en la línea de comandos. La siguiente parte especifica el script de python CLI del programador. La CLI del programador tiene comandos que especifican las operaciones que se deben realizar en períodos y horarios. 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	Descripción
<code>--stack &lt;stackname&gt;</code>	<p>El nombre de la pila de planificadores.</p> <p>Importante: este argumento es obligatorio para todos los comandos.</p>
<code>--region &lt;regionname&gt;</code>	<p>El nombre de la región en la que se despliega la pila de planificadores.</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>

Argumento	Descripción
<code>--profile-name</code> <i>&lt;profilename&gt;</i>	El nombre del perfil que se va a utilizar para ejecutar los comandos. Si no se especifica ningún nombre de perfil, se utiliza el perfil predeterminado.
<code>--query</code>	JMESPath Expresión que controla el resultado del comando. Para obtener más información sobre el control de la salida, consulte <a href="#">Control de la salida de comandos AWS Command Line Interface de la</a> Guía del AWS CLI usuario.
<code>--help</code>	Muestra comandos y argumentos válidos para la CLI del programador. Cuando se usa con un comando específico, muestra los subcomandos y argumentos válidos para ese comando.
<code>--version</code>	Muestra el número de versión de la CLI del programador.

## Comandos disponibles

- [período de creación](#)
- [crear-programar](#)
- [período de eliminación](#)
- [eliminar-programar](#)
- [describir períodos](#)
- [describir horarios](#)
- [describe-schedule-usage](#)
- [período de actualización](#)
- [calendario de actualizaciones](#)
- [help](#)

## período de creación

### Descripción

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

### Argumentos

`--name`

El nombre del período

Tipo: cadena

Obligatorio: sí

`--description`

Una descripción del período

Tipo: cadena

Requerido: no

`--begintime`

Hora en la que comienza el período de ejecución. Si `begintime` no `endtime` se especifican, el período de ejecución es de 00:00 a 23:59.

Tipo: cadena

Restricciones: o formato `H:MM HH:MM`

Obligatorio: no

`--endtime`

Hora en la que se detiene el período de ejecución. Si `begintime` no `endtime` se especifican, el período de ejecución es de 00:00 a 23:59.

Tipo: cadena

Restricciones: o formato `H:MM HH:MM`

Obligatorio: no

--weekdays

Los días de la semana del período

Tipo: cadena

Restricciones: lista de nombres de días abreviados (mon) o números (0) delimitados por comas. Utilice — para especificar un rango. Utilice/para especificar todos <sup>los</sup> días de la semana.

Obligatorio: no

--months

Los meses del período

Tipo: cadena

Restricciones: lista de nombres abreviados de meses (jan) o números (1) delimitados por comas. Utilice — para especificar un rango. Utilice/para especificar cada uno de <sup>los</sup> meses.

Obligatorio: no

--monthdays

Los días del mes del período

Tipo: cadena

Restricciones: lista de nombres abreviados de meses (jan) o números (1) delimitados por comas. Utilice — para especificar un rango. Utilice/para especificar todos <sup>los</sup> días 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"  
    ]  
}
```

## crear-programar

### Descripción

Crea un horario.

### Argumentos

`--name`

El nombre de la programación

Tipo: cadena

Obligatorio: sí

`--description`

Una descripción del horario

Tipo: cadena

Requerido: no

`--enforced`

Aplica el estado programado de la instancia

Obligatorio: no

`--use-metrics`

Recopila CloudWatch las métricas de Amazon

Obligatorio: no

`--periods`

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

Tipo: cadena

Restricciones: lista de períodos delimitada por comas. Se utiliza `<period-name>@<instance type>` para especificar un tipo de instancia para un período. Por ejemplo, `weekdays@t2.large`.

Obligatorio: sí

`--retain-running`

Evita que la solución detenga una instancia al final de un período de ejecución si la instancia se inició manualmente antes del comienzo del período.

Obligatorio: no

`--ssm-maintenance-window`

Añade una ventana AWS Systems Manager de mantenimiento como período de ejecución a una programación de EC2 instancias de Amazon. Para usar este comando, debe usar el `use-maintenance-window` comando.

Tipo: cadena

Requerido: no

`--do-not-stop-new-instances`

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

Obligatorio: no

`--timezone`

La zona horaria que utilizará la programación

Tipo: matriz de cadenas

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

`--use-maintenance-window`

Añade una ventana de mantenimiento de Amazon RDS como período de ejecución a una programación de instancias de Amazon RDS o una ventana de AWS Systems Manager mantenimiento como período de ejecución a una programación de instancias de Amazon EC2

Obligatorio: no

## 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"
  }
}
```

## período de eliminación

--name

El nombre del período aplicable

Tipo: cadena

Obligatorio: sí

### Important

Si el período se utiliza en los programas existentes, debe eliminarlo de esos programas antes de eliminarlo.

## Ejemplo

```
$ scheduler-cli delete-period --name weekdays --stack Scheduler
{
```

```
"Period": "weekdays"  
}
```

## eliminar-programar

### Descripción

Elimina un horario existente

### Argumentos

`--name`

El nombre del programa aplicable

Tipo: cadena

Obligatorio: sí

### Ejemplo

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

## describa los períodos

### Descripción

Muestra los períodos configurados para la pila del Programador de Instancias

### Argumentos

`--name`

El nombre de un período específico que desea describir

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"
  }
]
}
```

## describa los horarios

### Descripción

Muestra los horarios configurados para la pila del programador de instancias.

### Argumentos

`--name`

El nombre de un programa específico que desea describir

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

### Descripción

Enumera todos los períodos incluidos en un cronograma y calcula las horas de facturación, por ejemplo. Utilice este comando para simular una programación y calcular los posibles ahorros y los períodos de ejecución después de crear o actualizar una programación.

### Argumentos

**--name**

El nombre del programa aplicable

Tipo: cadena

Obligatorio: sí

**--startdate**

La fecha de inicio del período utilizado para el cálculo. La fecha por defecto es la fecha actual.

Tipo: cadena

Requerido: no

--enddate

La fecha de finalización del período utilizado para el cálculo. La fecha por defecto 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"
```

## período de actualización

### Descripción

Actualiza un período existente

### Argumentos

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

**⚠ Important**

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

## calendario de actualizaciones

### Descripción

Actualiza un horario existente

### Argumentos

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

**⚠ Important**

Si no especifica un argumento, dicho argumento se eliminará de la tabla de planificación.

## ayuda

### 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}
```

Commands help

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 `--help` argumento muestra los subcomandos y argumentos válidos para 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
```

## Actualizar los ajustes 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 en la CloudFormation consola.

Para actualizar la configuración global de Instance Scheduler, inicie sesión en la cuenta o región en la que se encuentre la implementación del hub y vaya a la consola. AWS CloudFormation Busca la pila de concentradores del programador de instancias y selecciona 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.

## Gestione los cronogramas mediante la infraestructura como código (IaC)

### Important

Implemente los cronogramas utilizando una plantilla independiente una vez que se complete el despliegue del hub stack.

Instance Scheduler on AWS proporciona un recurso personalizado (`ServiceInstanceSchedule`) que puede usar para configurar y administrar los cronogramas. 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 de los cronogramas, consulte las definiciones de los cronogramas.](#) Para obtener más información sobre los campos de los períodos, consulte las [definiciones de períodos.](#)

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

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

Para empezar a gestionar los horarios con laC, copia y pega la siguiente plantilla de ejemplo y personaliza tantos o tan pocos horarios como desees. Guarde el archivo como un archivo.template (por ejemplo:my-schedules.template) y, a continuación, implemente la nueva plantilla utilizando AWS CloudFormation Para ver ejemplos de plantillas de cronogramas completadas, consulte [Ejemplos de cronogramas](#).

```

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

Antes de eliminar la pila principal del programador de instancias, 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 DynamoDB de la pila principal.

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

# Características avanzadas

## EC2 Programación de Auto Scaling Group

Instance Scheduler on AWS admite la programación de grupos de EC2 Auto Scaling (ASGs) mediante acciones de escalado programadas. Esto difiere de la implementación de la programación de EC2 /RDS y se explicará con más detalle en esta sección

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

### Descripción general de la programación de ASG

ASGs se puede programar mediante la aplicación de una etiqueta de programación, tal como se describe en [Etiquetado de instancias para la programación](#)

A continuación, dos sistemas gestionarán las reglas de escalado programadas para su ASG:

En primer lugar, una función Lambda de ASG Orchestrator se ejecuta cada hora e inicia una función de controlador de ASG para cada una de sus cuentas/regiones programadas. Esta función buscará las acciones de escalado programadas recién etiquetadas ASGs o ASGs cuyas acciones de escalado programadas y configuradas hayan quedado obsoletas. A continuación, reconfigurará todas las acciones de escalado programadas que comiencen con el prefijo del nombre de la acción de ASG (especificado en la implementación de la solución) para que coincidan con la programación asociada.

En segundo lugar, cuando se actualice una programación en la tabla de configuración del planificador, se iniciará una transmisión de DynamoDB (mediante la función Lambda del controlador de actualización del cronograma) solicitudes de controlador ASG adicionales que actualizarán las acciones de escalado programadas en todas las etiquetas con la programación recién actualizada. ASGs

### Definición de «En ejecución/parada» para ASGs

Cuando se configura un grupo de Auto Scaling, el usuario especifica una capacidad mínima, deseada y máxima para ese ASG. El programador de instancias denomina a estos valores los min-desired-max de un ASG.

Cuando el programador de instancias configure por primera vez las acciones de escalado programadas para un ASG, los min-desired-max valores configurados actualmente se utilizarán

para definir el estado de ejecución del ASG. Si el ASG está configurado actualmente con un valor min-desired-max de 0-0-0, el programador de instancias notificará un error y no configurará ninguna acción de escalado programada hasta que se configure una nueva que pueda usarse para definir un min-desired-max estado de ejecución del ASG.

Al actualizar las acciones de escalado programadas para un ASG, el programador de instancias analizará las actuales min-desired-max en el momento de la actualización y utilizará esos valores para definir el nuevo estado de ejecución de la programación. Si actualmente min-desired-max es 0-0-0 en el momento de la actualización, se utilizará el estado de ejecución anterior.

En general, ASGs el estado detenido se define como 0-0-0. min-desired-max

## Etiqueta programada ASG

Cuando la solución programa un grupo de autoescalado, se agrega una etiqueta programada de grupo de autoescalado al grupo de autoescalado. La etiqueta contiene la siguiente información en formato JSON:

Clave	Tipo de valor	Valor
schedule	Cadena	El nombre del programa está alineado con la tabla de configuración del planificador.
ttl	Cadena	Hasta que la etiqueta sea válida.
min_size	Entero	Se ajusta automáticamente el tamaño mínimo del grupo según lo programado.
max_size	Entero	Escalado automático del tamaño máximo del grupo según lo programado.
desired_size	Entero	El escalado automático agrupa la capacidad deseada cuando esté programado.

La presencia de una etiqueta programada válida que aún no haya caducado su TTL indica al Programador de Instancias que un ASG se ha configurado correctamente para la programación. Esta etiqueta se puede eliminar manualmente para que el programador de instancias vuelva a configurar las acciones de escalado programadas en un ASG durante la siguiente ejecución de programación del ASG.

## Limitaciones

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

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

- Indicadores de programación avanzados, como Forzar y Mantener en ejecución.
- Expresiones del n-ésimo día de la semana, el día de la semana más cercano y el último día de la semana en puntos.
- Programaciones de varios períodos con períodos inmediatamente adyacentes o superpuestos. \*

\*al configurar acciones de escalado programadas para programas de varios períodos, Instance Scheduler AWS activa directamente las beginning/end of periods to start/stop acciones para el ASG, incluso cuando otro período superpuesto o adyacente normalmente provocaría que se omitiera esa acción.

## Supervise la solución

### Registro y notificaciones

Instance Scheduler on AWS utiliza Amazon CloudWatch Logs para el registro. Esta solución registra la información de procesamiento de cada instancia etiquetada, los resultados del período de evaluación de la instancia, el estado deseado de la instancia durante ese período, la acción aplicada y los mensajes de depuración. Para obtener más información, consulte los [recursos de la solución](#).

Los mensajes de advertencia y error también se publican en un tema de Amazon SNS creado por una solución, que envía mensajes a una dirección de correo electrónico suscrita. Para obtener más información, consulte ¿Qué es Amazon SNS? en la Guía para desarrolladores de Amazon SNS. Puede encontrar el nombre del tema de Amazon SNS en la pestaña Salidas de la pila de soluciones.

## Archivos de registro

El programador de instancias activado AWS crea un grupo de registros que contiene los archivos de AWS Lambda registro predeterminados y un grupo de registros que contiene los siguientes archivos de registro:

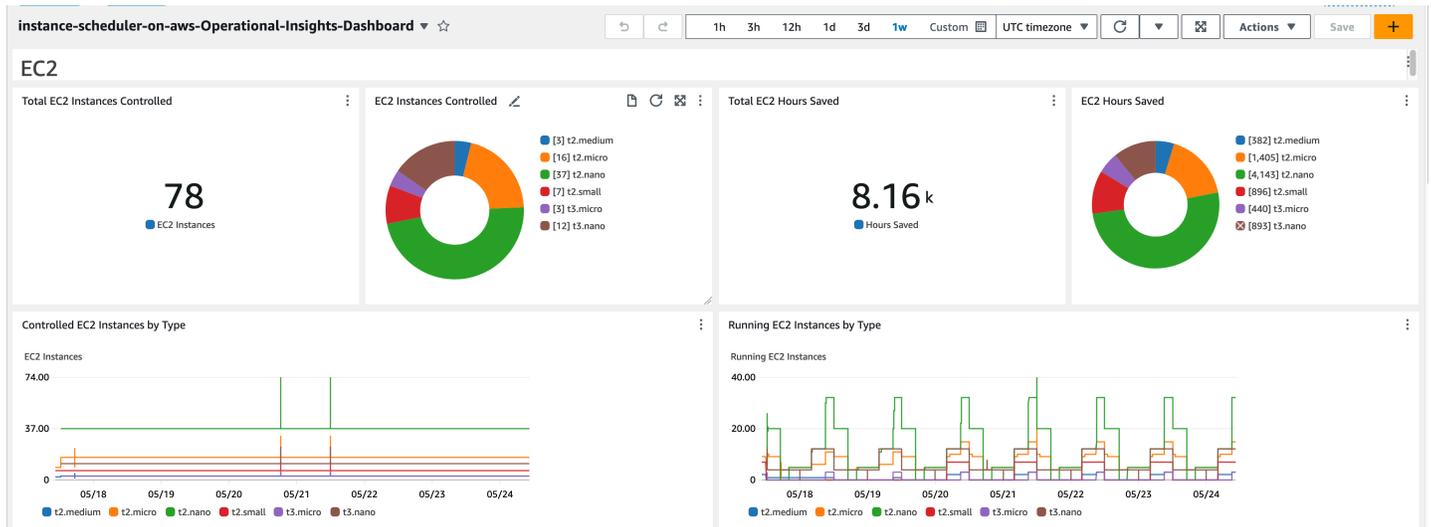
- `InstanceScheduler-yyyyymmdd`: Registra los mensajes generales del programador
- `SchedulingOrchestratorHandler-yyyyymmdd`: Registra la información general de orquestación para cuando se inician las ejecuciones programadas
- `SchedulerSetupHandler-yyyyymmdd`: Registra el resultado de las acciones de configuración
- `Scheduler-<service>-<account>-<region>-yyyyymmdd`: Registra la actividad de programación en cada servicio, cuenta y región
- `CliHandler-yyyyymmdd`: Registra las solicitudes de la CLI de administración
- `Eventbus_request_handler-yyyyymmdd`: Registra las llamadas a los EventBus recursos, si la solución se implementa en las AWS organizaciones.
- `CollectConfigurationDescription-yyyyymmdd`: Registra los datos métricos de descripción de la configuración que se envían periódicamente

## Panel de información operativa

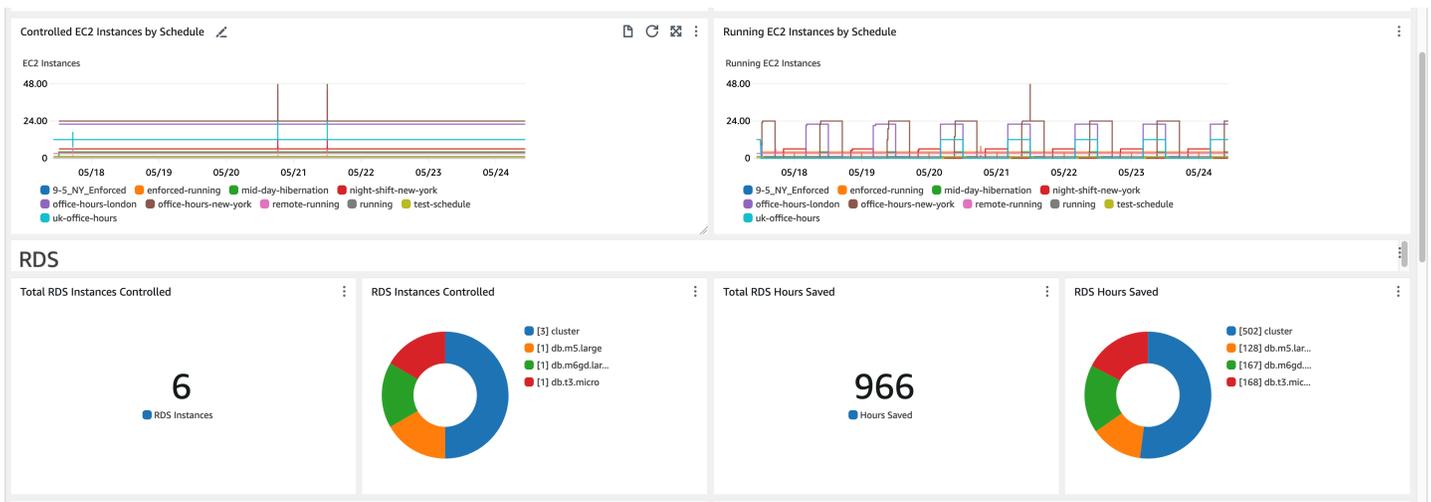
Instance Scheduler on AWS incluye un panel de información operativa que le permite monitorear el funcionamiento de la solución y obtener información sobre las horas de trabajo que se han ahorrado al usar esta solución.

Para usar este panel, asegúrese de que la supervisión operativa esté configurada como «habilitada» en los parámetros del hub stack de la solución. AWS CloudFormation A continuación, vaya a «Paneles de control» AWS CloudWatch y seleccione «Paneles de control» en el menú de navegación. El nombre del panel será `{stack-name}-Operational-Insights-Dashboard`

El panel mostrará varias métricas operativas sobre las operaciones de la solución, incluidos los recuentos del número de instancias que gestiona actualmente la solución, información sobre cuándo y cuántas instancias se están ejecutando a lo largo del día y una estimación del número de horas de funcionamiento que se han ahorrado al cerrar las instancias. A continuación, se muestran algunos datos:



### Programador de instancias (AWS) apilado CloudWatch



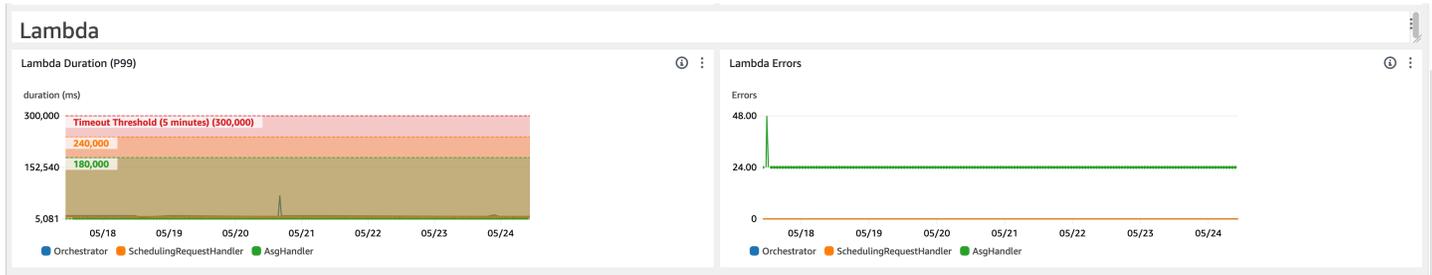
### EC2 Instancias controladas por cronograma

**Note**

La información de estos gráficos depende del intervalo de programación configurado en la pila de centros de soluciones. Al actualizar el intervalo de programación de la solución, el panel solo mostrará las métricas de programación posteriores a la actualización más reciente del intervalo de programación.

El panel también proporciona información sobre el estado de las funciones de Lambda que son fundamentales para el funcionamiento de la solución. Si la duración media de Lambda de alguna

de las funciones de Lambda representadas comienza a acercarse a la zona amarilla, puede que sea el momento de aumentar la propiedad de tamaño de Lambda en la pila de concentradores de soluciones.



### Duración Lambda

Representado: AsgHandler genera errores consistentes a lo largo de varios días. Esto indica un posible problema con la programación de ASG y debería impulsar una mayor investigación de los registros de esa Lambda.

### Costes adicionales asociados a esta función

Este panel operativo se basa en CloudWatch métricas personalizadas recopiladas por la solución, que generarán un costo adicional. Esta función se puede desactivar deshabilitando la «supervisión operativa» en el conjunto de centros de soluciones. Esta función cuesta 3\$ adicionales al mes, más los costes de escalado adicionales en función del tamaño de la implementación. Los costos son los siguientes:

CloudWatch Panel de control personalizado	3 dólares
Métricas programadas	0,60 USD por programa*
Métricas de Per-instance-type	0,90 USD por tipo de instancia*
Uso de la API	Aproximadamente 0,10\$ por cuenta/región

\* Estos costos se registran por categoría de servicio (. EC2/RDS) and only for schedules/instance types actually used for scheduling. For example, if you have 15 schedules configured, with three for RDS and five for EC2, the total cost will be 8\*\$0.60 or \$4.80/month No se facturarán los horarios inactivos.

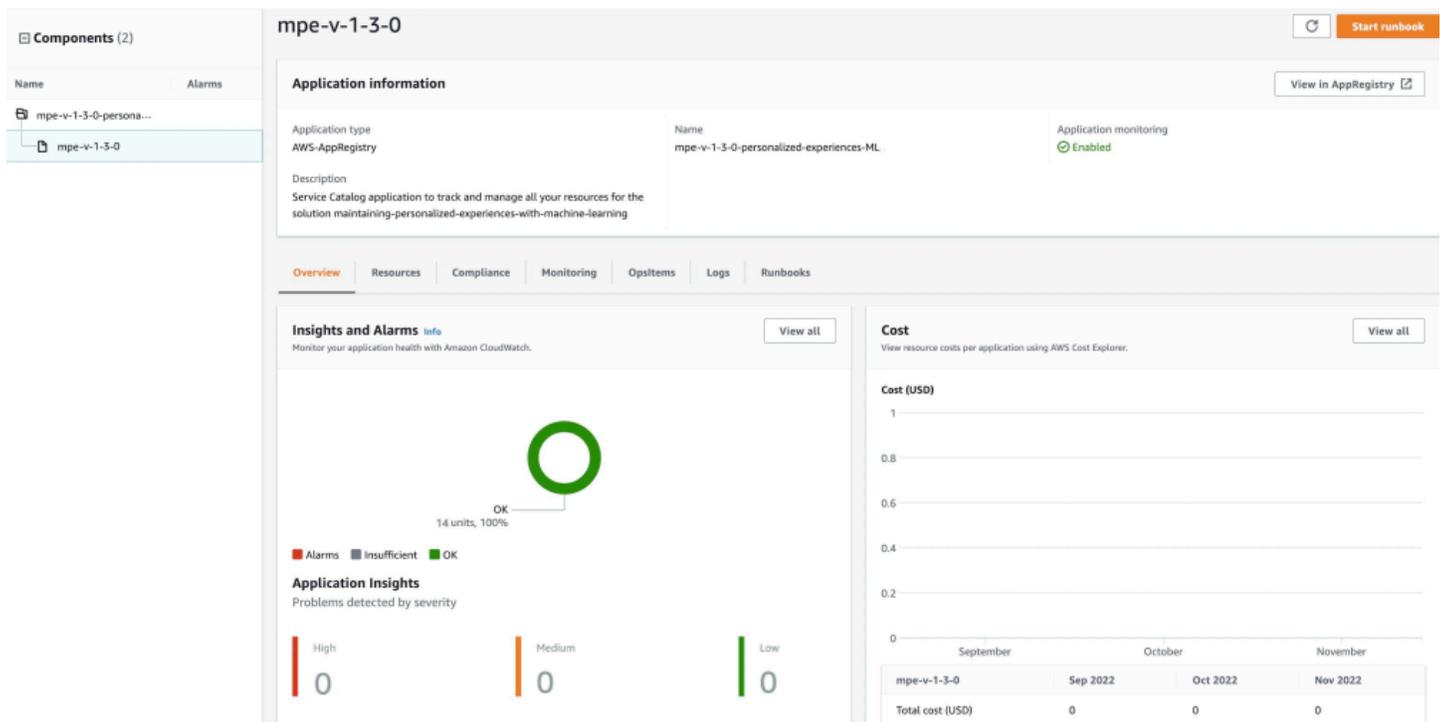
## Supervise la solución con Service Catalog AppRegistry

La solución incluye un AppRegistry recurso de Service Catalog para registrar la CloudFormation plantilla y los recursos subyacentes como una aplicación tanto en [Service Catalog AppRegistry](#) como en [AWS Systems Manager Application Manager](#).

AWS Systems Manager Application Manager le ofrece una visión a nivel de aplicación de esta solución y sus recursos para que pueda:

- Supervisar sus recursos, los costos de los recursos implementados en todas las pilas y Cuentas de AWS, así como los registros asociados a esta solución desde una ubicación central.
- Vea los datos de operaciones de los recursos de esta solución en el contexto de una aplicación, como el estado de la implementación, CloudWatch las alarmas, las configuraciones de los recursos y los problemas operativos.

En la siguiente figura, se muestra un ejemplo de la vista de la aplicación para Instance Scheduler AWS apilada en Application Manager.



### Pila de soluciones en Application Manager

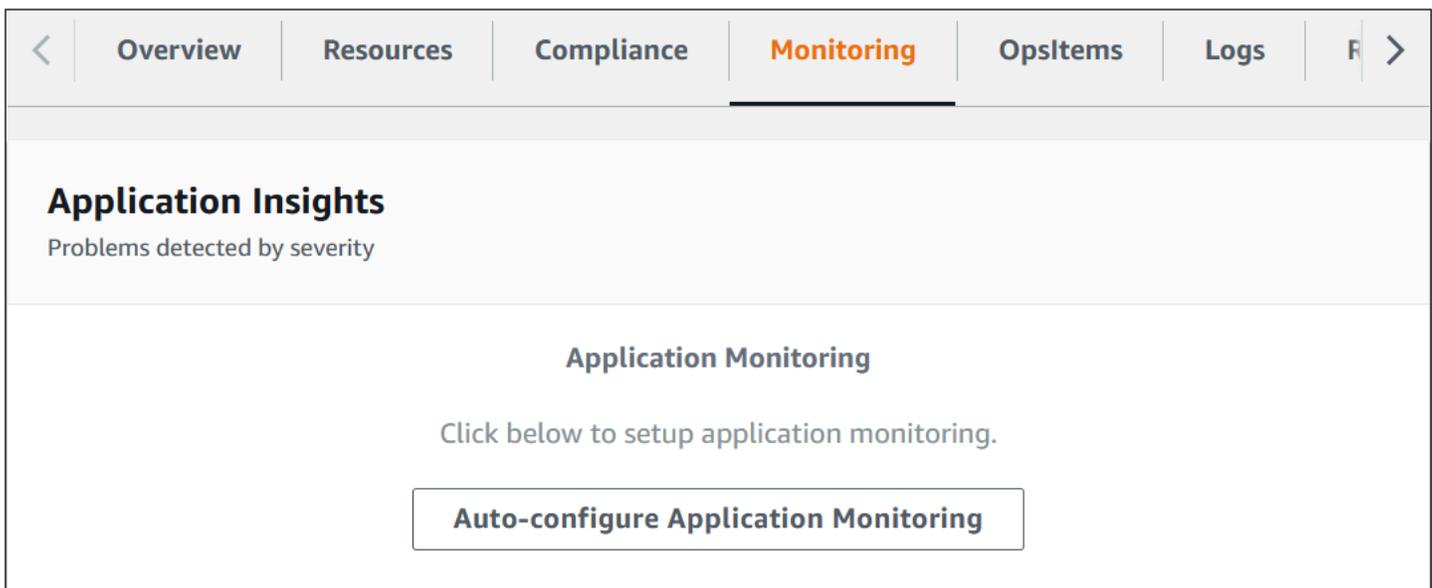
Nota: Debe activar CloudWatch Application Insights y las etiquetas de asignación de costes asociadas a esta solución. AWS Cost Explorer No están activadas de forma predeterminada.

## Active CloudWatch Application Insights

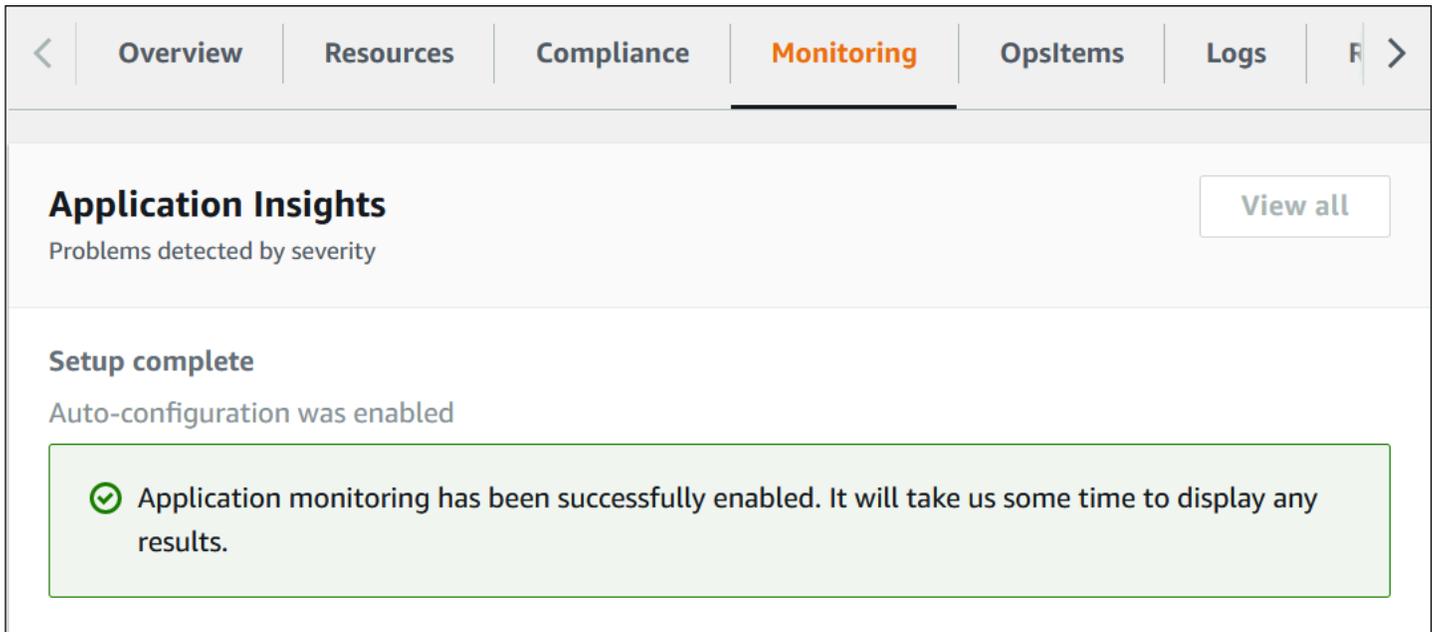
1. Inicie sesión en la [consola de Administrador de aplicaciones](#).
2. En el panel de navegación, elija Administrador de aplicaciones.
3. En Aplicaciones, busque el nombre de la aplicación para esta solución y selecciónela.

El nombre de la aplicación tendrá el registro de aplicaciones en la columna Fuente de la aplicación y tendrá una combinación del nombre de la solución, la región, el identificador de cuenta o el nombre de la pila.

4. En el árbol de componentes, elija la pila de aplicaciones que desee activar.
5. En la pestaña Supervisión, en Application Insights, seleccione Configurar automáticamente Application Insights.



Ahora, al estar activada la supervisión de sus aplicaciones, aparece el siguiente cuadro de estado:

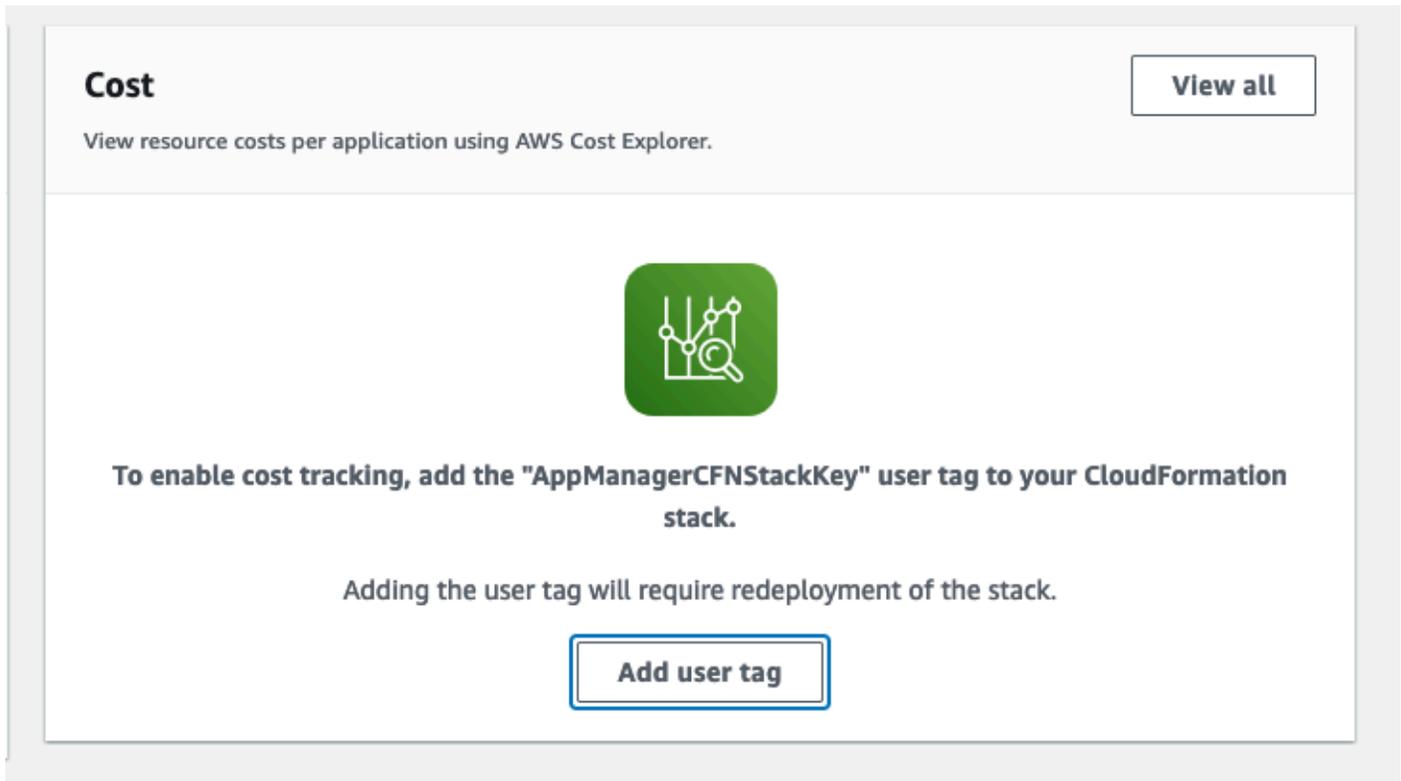


The screenshot shows the AWS Application Insights Monitoring page. At the top, there is a navigation bar with tabs for Overview, Resources, Compliance, Monitoring (which is selected and highlighted in orange), OpsItems, and Logs. Below the navigation bar, the main content area is titled "Application Insights" with the subtitle "Problems detected by severity" and a "View all" button. A green box with a checkmark icon and the text "Setup complete" and "Auto-configuration was enabled" is displayed. Below this, a larger green box contains a success message: "Application monitoring has been successfully enabled. It will take us some time to display any results."

## Confirmación de las etiquetas de costos asociadas a la solución

Después de activar Cost Explorer, debe activar las etiquetas de asignación de costos asociadas a esta solución para ver los costos de la solución. Para confirmar las etiquetas de asignación de costos:

1. Inicie sesión en la [consola de Administrador de aplicaciones](#).
2. En el panel de navegación, elija Administrador de aplicaciones.
3. En Aplicaciones, busque el nombre de la aplicación para esta solución y selecciónela.
4. En la pestaña Descripción general, en Costo, seleccione Agregar etiqueta de usuario.



5. En la página Agregar etiqueta de usuario, escriba `confirm` y, a continuación, seleccione Agregar etiqueta de usuario.

El proceso de activación puede tardar hasta 24 horas en completarse y en aparecer los datos de la etiqueta.

## Activar las etiquetas de asignación de costos asociadas a la solución

Después de activar Cost Explorer, debe activar las etiquetas de asignación de costos asociadas a esta solución para ver los costos de la solución. Las etiquetas de asignación de costos sólo se pueden activar desde la cuenta de administración de la organización. Para activar las etiquetas de asignación de costos:

1. Inicie sesión en la [consola de Administración de facturación y costos de AWS and Cost Management](#).
2. En el panel de navegación, seleccione Etiquetas de asignación de costes.
3. En la página Etiquetas de asignación de costos, filtre por la etiqueta `AppManagerCFNStackKey` y, a continuación, selecciónela entre los resultados que se muestran.
4. Seleccione Activar.

## AWS Cost Explorer

Puede ver un resumen de los costes asociados a la aplicación y a los componentes de la aplicación en la consola de Application Manager AWS Cost Explorer , cuya integración debe activarse primero. Cost Explorer lo ayuda a administrar los costos al proporcionarle una vista de los costos y el uso de sus recursos de AWS a lo largo del tiempo. Para activar Cost Explorer para la solución:

1. Inicie sesión en la [Consola de administración de AWS](#).
2. En el panel de navegación, seleccione Cost Explorer para ver los costos y el uso de la solución a lo largo del tiempo.

## Rendimiento

Si la AWS Lambda función de la solución no procesa todas las instancias programadas antes de su siguiente invocación, la solución registra el error en Amazon CloudWatch Logs y envía una notificación de Amazon Simple Notification Service (Amazon SNS) al tema SNS del error. Para garantizar que todas las instancias se procesen antes de la siguiente invocación, puede cambiar el intervalo predeterminado en el que se ejecuta la función Lambda o lanzar varias implementaciones de la solución con nombres de etiqueta diferentes.

Si aumentas el intervalo predeterminado, esto podría reducir la granularidad de tus programaciones. Por ejemplo, una función Lambda configurada para ejecutarse en un intervalo de 15 minutos solo realizará acciones de inicio y parada cada 15 minutos.

Para programar un gran número de instancias, se recomienda utilizar un intervalo de al menos cinco minutos y aumentar el tamaño de la memoria de la AWS Lambda función principal del programador de instancias mediante el parámetro Tamaño de memoria.

## Actualización de la solución

### Important

Instance Scheduler v1.5.0 tiene un problema de compatibilidad conocido AppRegistry que le impide actualizarse directamente a versiones más recientes de la solución.

Si planea actualizar de la v1.5.0 a cualquier versión AppRegistry habilitada para el futuro, primero debe actualizar a la pila intermedia 1.5.0-u mediante las siguientes plantillas:

[Hub Stack: v1.5.0/aws- instance-scheduler-1.5.0-u.template https://solutions-reference.s3.amazonaws.com/aws-instance-scheduler/](https://solutions-reference.s3.amazonaws.com/aws-instance-scheduler/v1.5.0/aws-instance-scheduler-1.5.0-u.template)

Pila remota: [https://solutions-reference.s3.amazonaws.com/aws-instance-scheduler/ v1.5.0/aws-instance-scheduler-remote -1.5.0-u.template](https://solutions-reference.s3.amazonaws.com/aws-instance-scheduler/v1.5.0/aws-instance-scheduler-remote-1.5.0-u.template)

La instalación de estas pilas deshabilitará la AppRegistry integración en su implementación, lo que permitirá que las versiones más recientes de la solución vuelvan a crear la asociación.

Ejemplo de ruta de actualización: 1.5.0 -> 1.5.0-u -> 3.0.4

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

1. Inicia sesión en la [AWS CloudFormation consola](#), en la cuenta o región en la que esté instalada tu pila de Hub **instance-scheduler-on-aws**, selecciona y selecciona Actualizar.
2. Seleccione Reemplazar la plantilla actual.
3. En Especificar plantilla:
  - Seleccione URL de Amazon S3.
  - Copia 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.
4. 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 si es necesario actualizar los parámetros). Para obtener más información sobre cada uno de los parámetros, consulte el [paso 1. Lanza la pila de planificadores de instancias](#).
5. Elija Next (Siguiente).
6. En la página Configurar opciones de pila, elija Siguiente.
7. En la página Revisar, revise y confirme la configuración. Marca la casilla para confirmar que la plantilla creará recursos AWS Identity and Access Management (IAM).
8. Seleccione Ver conjunto de cambios y verifique los cambios.
9. Seleccione Crear pila para implementar la pila.

Puedes ver el estado de la pila en la AWS CloudFormation consola, en la columna Estado. Deberías recibir el estado UPDATE\_COMPLETE en unos minutos.

Repita los pasos anteriores para las `aws-instance-scheduler-remote` pilas de cada una de tus cuentas habladas.

## Cambios importantes en versiones específicas

Al actualizar la solución, puede actualizarla directamente de una versión anterior a una versión más reciente sin pérdida de datos ni interrupciones en las operaciones de programación, excepto cuando se indique explícitamente a continuación. Al actualizar versiones anteriores a determinadas versiones, es posible que deba realizar determinadas acciones en relación con las versiones que vaya a aprobar. Por ejemplo, al actualizar de la versión 1.4.1 a la versión 3.0.2, sigue las instrucciones para corregir los cambios de las versiones 1.5.0 y 3.0.0.

### v1.5.0

La versión 1.5.0 reemplaza la necesidad de proporcionar una lista de funciones ARNs de programación multicuenta por la posibilidad de administrarlos automáticamente a través de su organización de AWS. Si no desea utilizar AWS Organizations, puede proporcionar una lista de Spoke Account IDs e Instance Scheduler se encargará de gestionar las funciones de programación por usted.

Al actualizar a la versión 1.5.0 o posterior, debe:

1. Actualice la plantilla del hub siguiendo las instrucciones de actualización normales y, al mismo tiempo, actualice los siguientes parámetros:
  - a. Elija un espacio de nombres único para la solución.
  - b. Seleccione si desea utilizarlo para gestionar el registro AWS Organizations de voces en el futuro.
    - i. Si seleccionó Sí, sustituya el ID de la organización o la cuenta IDs remota por el ID de su AWS organización.
    - ii. Si seleccionó No, sustituya el ID de la organización RemoteAccount IDs por una lista separada por comas de la cuenta de sus cuentas de Spoke. IDs
2. Actualice todas las pilas remotas siguiendo las instrucciones de actualización normales y, al mismo tiempo, actualice los siguientes parámetros:
  - a. Espacio de nombres: el mismo que elegiste para la cuenta hub.
  - b. Uso AWS Organizations : igual que el de la cuenta central.

- c. ID de cuenta de Hub: ID de cuenta de la cuenta de hub (no debe modificarse con respecto a la anterior).

## v3.0.0

La versión 3.0.0 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.](#)
- Las métricas programadas se CloudWatch han trasladado de. Schedule/Service/MetricName → Schedule/Service/SchedulingInterval/MetricName
- Se mantendrán todas las métricas existentes, pero las nuevas se recopilarán ahora en el nuevo espacio de nombres y estarán disponibles en el panel de control de la solución.
- La clave KMS ARNs para su uso con volúmenes de EBS cifrados en EC2 instancias de bases de datos ahora debe proporcionarse a la CloudFormation pila hub/spoke de sus cuentas respectivas. (Para obtener más información, consulte los volúmenes de [EC2 EBS cifrados](#)).
- Si está programando EC2s con volúmenes de EBS cifrados, tendrá que copiar los arns de claves de KMS que se utilizan en los parámetros de su pila de 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 individualmente.
- Instance Scheduler 3.0 no es compatible con versiones anteriores de la CLI del Instance Scheduler.
  - Deberá actualizar a la última versión de la CLI del programador de instancias para seguir utilizando los comandos de la CLI.

Además de lo anterior, se actualizó el esquema de la tabla de ventanas de mantenimiento y se reemplazará como parte de la actualización. Esto restablecerá el seguimiento de los EC2 períodos de mantenimiento durante los primeros minutos tras la actualización a la versión 3.x y, en raras ocasiones, puede provocar que las instancias que se encuentren dentro de un período de mantenimiento se detengan prematuramente inmediatamente después de la actualización. Una vez regenerados estos datos, las operaciones de programación continuarán con normalidad.

# Solución de problemas

En esta sección se proporcionan instrucciones de solución de problemas para implementar y usar la solución.

La resolución de problemas conocidos proporciona instrucciones para mitigar los errores conocidos. Si estas instrucciones no resuelven el problema, [Contact Soporte](#) proporciona instrucciones para abrir un Soporte caso para esta solución.

## Resolución de problemas conocidos

### Problema: las instancias no están programadas en una cuenta remota

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

### Resolución

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

1. En la cuenta principal, navega hasta la [CloudWatch consola](#)
2. En el panel de navegación, selecciona Registros > Grupos de registros.
3. Seleccione el grupo de registros 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 identificador 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 y vuelva a instalar la plantilla remota en la cuenta remota.
11. Si la solución está configurada para usar una cuenta remota IDs, actualice el parámetro de `cloudformation` Proporcione el identificador de la organización o la lista de cuentas remotas IDs con la lista de cuentas en las IDs que se programarán las instancias y en las que se implementará la plantilla remota.

## Problema: la solución se actualizó de cualquier versión v1.3.x a la v1.5.0

La función Lambda no funciona, por ejemplo, no se está realizando la programación.

### Resolución

1. Asegúrese de que la actualización de la CloudFormation pila se haya completado.
2. Ve a la CloudFormation consola y selecciona la pila de soluciones.
3. Seleccione la pestaña Recursos.
4. Busque Main en el filtro Buscar recursos.
5. Seleccione la función Lambda en la columna ID física.
6. En la consola Lambda, seleccione Configuración.
7. Seleccione las variables de entorno.
8. Asegúrese de que estén disponibles las siguientes variables de entorno.
  - CUENTA
  - CONFIG\_TABLE
  - DDB\_TABLE\_NAME
  - HABILITAR\_SSM\_MAINTENANCE\_WINDOWS
  - ISSUES\_TOPIC\_ARN
  - LOG\_GROUP
  - TABLA\_VENTANA DE MANTENIMIENTO
  - METRICS\_URL
  - PLANIFICADOR\_FRECUENCIA
  - SEND\_METRICS
  - ID\_SOLUCIÓN
  - STACK\_ID
  - STACK\_NAME
  - START\_BATCH\_SIZE EC2
  - TABLA\_DE ESTADOS
  - NOMBRE\_ETIQUETA
  - TRACE
  - AGENTE DE USUARIO

- USER\_AGENT\_EXTRA
- UUID\_KEY

## Problema: las instancias cifradas EC2 no se inician

El programador de instancias informa que EC2 las instancias con volúmenes de EBS cifrados se están iniciando, pero en realidad nunca se inician.

### Resolución

Consulta los [volúmenes de EC2 EBS cifrados](#) para saber cómo conceder acceso al programador de instancias para poder programar EC2 instancias con volúmenes de EBS cifrados

## Problema: las instancias de RDS no se detienen cuando la opción Crear instantáneas de RDS está habilitada

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 posterior, o bien añada el `rds:CreateDBSnapshot` permiso a la función de programador de la solución en cada cuenta programada.

## Contacto Soporte

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

### Cree un caso

1. Inicie sesión en [Support Center](#).
2. Seleccione Crear caso.

## ¿Cómo podemos ayudar?

1. Elija Técnico.
2. Para el servicio, seleccione Soluciones.
3. En Categoría, seleccione Instance Scheduler en AWS (Linux o Windows).
4. En Gravedad, selecciona la opción que mejor se adapte a tu caso de uso.
5. Al introducir el servicio, la categoría y la gravedad, la interfaz rellena los enlaces a las preguntas de solución de problemas más frecuentes. Si no puede resolver su pregunta con estos enlaces, seleccione Siguiente paso: información adicional.

## Información adicional

1. En Asunto, introduce un texto que resuma tu pregunta o problema.
2. En Descripción, describe el problema en detalle.
3. Selecciona Adjuntar archivos.
4. Adjunta la información Soporte necesaria para procesar la solicitud.

## Ayúdenos a resolver su caso más rápido

1. Introduzca la información solicitada.
2. Elija Siguiente paso: Resuelva ahora o póngase en contacto con nosotros.

## Resuelva ahora o póngase en contacto con nosotros

1. Revise las soluciones Solve now.
2. Si no puede resolver su problema con estas soluciones, elija Contactar con nosotros, introduzca la información solicitada y pulse Enviar.

# Desinstalar la solución

## Important

Al desinstalar la solución, asegúrese de desinstalar todas las pilas de programas personalizadas antes de desinstalar la propia solución.

Puede desinstalar el programador de instancias de la AWS solución AWS Management Console o mediante el. AWS Command Line Interface Para desinstalar la solución, elimine la pila central de Nube de AWS Formation 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 manualmente una vez que la pila central ya se haya eliminado.

## Usando el AWS Management Console

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

## Usando AWS Command Line Interface

Determine si el AWS Command Line Interface (AWS CLI) está disponible en su entorno. Para obtener instrucciones de instalación, consulte [¿Qué es? AWS Command Line Interface](#) en la Guía del AWS CLI usuario. Tras confirmar que AWS CLI 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 de la solución y se enumeran las secciones añadidas aquí, además de incluir enlaces a cada subtema.

## Código fuente

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

El programador de instancias de las AWS plantillas se genera mediante [AWS CDK](#). Consulte el archivo [README.md](#) para obtener información adicional.

# Referencia

Esta sección incluye información sobre una función opcional para recopilar métricas únicas para esta solución, [cuotas](#), indicadores a [recursos relacionados](#) y una [lista de los creadores](#) que han contribuido a esta solución.

## Recopilación de datos anonimizados

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. Cuando se invoca, la siguiente información se recopila periódicamente y se envía a AWS:

- ID de solución: el identificador de la solución de AWS.
- ID único (UUID): identificador único generado aleatoriamente para cada programador de instancias en la implementación de AWS.
- Marca de tiempo: marca de tiempo de recopilación de datos.
- Programación de acciones: cuenta la frecuencia con la que el programador de instancias realiza determinadas acciones en las instancias y el tiempo que tarda en realizarlas.

Datos de ejemplo:

```
num_unique_schedules: 4
num_instances_scanned: 23
duration_seconds: 6.7
actions: [
  {
    action: Started
    instanceType: a1.medium
    instances: 8
    service: ec2
  },
  ...
]
```

- Recuento de instancias: recuento del número de instancias y programas que se están procesando en cada región.

**Datos de ejemplo:**

```

service: [ec2]
regions: [us-east-1]
num_instances: 35
num_schedules: 6

```

- Descripción de la implementación Una descripción general de la implementación:

**Datos de ejemplo:**

```

services: [ec2, rds]
regions: [us-east-1, us-east-2]
num_accounts: 6
num_schedules: 12
num_cfn_schedules: 3
default_timezone: UTC
schedule_aurora_clusters: True,
create_rds_snapshots: False,
schedule_interval_minutes: 10,
memory_size_mb: 128,
using_organizations: False,
enable_ec2_ssm_maintenance_windows: True,
num_started_tags: 1,
num_stopped_tags: 0,
schedule_flag_counts: {
  stop_new_instances: 10,
  enforced: 3,
  retain_running: 0,
  hibernate: 0,
  override: 0,
  use_ssm_maintenance_window: 2,
  use_metrics: 0,
  non_default_timezone: 4,
},

```

- Uso de la CLI Con qué frecuencia se utiliza cada función de la CLI del programador.

**Datos de ejemplo:**

```

command_used: describe-schedule-usage

```

AWS es propietario de los datos recopilados a través de esta encuesta. La recopilación de datos está sujeta a la [Política de privacidad de AWS](#). Para excluirse de esta función, complete los siguientes pasos antes de lanzar la CloudFormation plantilla.

1. Descarga la `instance-scheduler-on-aws.template` [AWS CloudFormation plantilla](#) en tu disco duro local.
2. Abre la CloudFormation plantilla con un editor de texto.
3. Modifique la sección CloudFormation de mapeo de plantillas desde:

```
"Send": {  
  "AnonymousUsage": {  
    "Data": "Yes"  
  }  
}
```

a

```
"Send": {  
  "AnonymousUsage": {  
    "Data": "No"  
  }  
}
```

4. Inicie sesión en la [CloudFormation consola de AWS](#).
5. Elija Crear pila.
6. En la página Crear pila, en la sección Especificar plantilla, seleccione Cargar un archivo de plantilla.
7. En Cargar un archivo de plantilla, seleccione Elegir archivo y después seleccione la plantilla editada de su unidad local.
8. Seleccione Siguiente y siga los pasos de la sección [Lanzar la pila](#) de esta guía.

## Cuotas

Service Quotas, también denominadas límites, establecen el número máximo de recursos u operaciones de servicio para su cuenta de Cuenta de AWS.

## Cuotas de AWS los servicios de esta solución

Asegúrese de tener una cuota suficiente para cada uno de los [servicios implementados en esta solución](#). Para más información, consulte [Service Quotas de AWS](#).

Utilice los siguientes enlaces para ir a la página de ese servicio. Para ver las cuotas de servicio de todos los AWS servicios de la documentación sin cambiar de página, consulte la información en la página de [puntos finales y cuotas del servicio](#) en formato PDF.

## AWS CloudFormation cuotas

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

## Cuotas de AWS Lambda

Su cuenta tiene AWS Lambda una cuota de ejecución simultánea de 1000. Si la solución se utiliza en una cuenta en la que hay otras cargas de trabajo en ejecución y que utilizan Lambda, esta cuota debe establecerse en un valor adecuado. Este valor es ajustable. Para obtener más información, consulte la guía de [AWS Lambda introducción](#).

## Recursos relacionados

El [Programador de recursos](#) es similar al Programador de Instancias AWS, pero su implementación difiere en los siguientes aspectos:

Instance Scheduler on AWS utiliza una función Lambda para evaluar con frecuencia las planificaciones almacenadas en su configuración y comprueba si las instancias se encuentran en el estado deseado. La configuración rápida del programador de recursos utiliza las horas de inicio y parada para realizar acciones de inicio y parada mediante los manuales de ejecución de SSM. Esto ocurre una vez cuando la hora actual es igual a la hora de inicio o la hora actual ha pasado la hora de inicio.

El programador de instancias activado AWS actualmente permite la programación de EC2 clústeres de RDS y Aurora. El programador de recursos solo programa, inicia y detiene las instancias. EC2

Utilice el programador de recursos para identificar las EC2 instancias e iniciarlas o detenerlas en momentos específicos.

Utilice el programador de instancias AWS cuando haya que escanear las cuentas con regularidad para iniciar o detener las instancias.

La tabla identifica qué solución es mejor 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 Auto Scaling	No	Sí
Programe EC2 instancias	Sí	Sí
Programe instancias de RDS	No	Sí
Programar cúmulos de auroras	No	Sí
Gestione los horarios en una sola cuenta (cuenta central)	No	Sí
Administre los horarios en cuentas individuales	Sí	No
Cambiar la integración del calendario	Sí	No
Solo acciones de inicio y parada	Sí	No
Supervise las instancias periódicamente e inicie y	No	Sí

Escenario	Programador de recursos	Programador de instancias de AWS
detenga en función del estado actual de la instancia		

## Colaboradores

- Arie Leeuwesteijn
- Mahmoud ElZayet
- Rudald Andreas
- Nikhil Reddy
- Caleb Pearson
- Jason DiDomenico
- Max Granat
- Pratyush Das
- Amanda Jones
- Kevin Hargita
- Beomseok Lee

# Revisiones

Date	Cambio
de febrero de 2018	Versión inicial
Julio de 2018	Se agregó una aclaración sobre la hora de inicio y finalización; se agregó un ejemplo de configuración de programación
Octubre de 2018	Se agregó información sobre los volúmenes cifrados de Amazon EBS; se agregó una aclaración sobre la longitud del nombre de la pila y las restricciones de las etiquetas de Amazon RDS
Mayo de 2019	Se agregó información sobre la hibernación de instancias de Amazon Ec2, la programación de instancias de bases de datos de RDS que forman parte de un clúster de Amazon Aurora, los nuevos argumentos de la CLI de programación, los períodos de mantenimiento de SSM, la integración del almacén de parámetros y una aclaración sobre los cronogramas con períodos de ejecución adyacentes
Octubre de 2019	Se agregó información sobre la ubicación en la que Regiones de AWS se ha validado el programador de instancias AWS
Marzo de 2020	Correcciones de errores
Junio de 2020	Se ha aclarado la información sobre la zona horaria para las horas de inicio y finalización de las reglas del período; para obtener información sobre las actualizaciones y los cambios de

Date	Cambio
	la versión 1.3.2, consulte el archivo ChangeLog .md del repositorio. GitHub
Septiembre de 2020	AWS Cloud Development Kit (AWS CDK) (AWS CDK) mejoras en la conversión y la documentación; para obtener más información sobre los cambios en la versión 1.3.3, consulte el archivo ChangeLog.md del repositorio. GitHub
Octubre de 2020	Se actualizó la plantilla del apéndice D y se actualizó la documentación del campo Hibernate.
Abril de 2021	Se actualizó la funcionalidad de la ventana de mantenimiento del SSM, por EC2 ejemplo, la programación, la solución configurada para trabajar y las correcciones de errores. AWS GovCloud Para obtener más información sobre la versión 1.4.0, consulte el archivo ChangeLog .md del repositorio. GitHub
Mayo de 2022	Actualizaciones menores y correcciones de errores. Para obtener más información sobre la versión 1.4.1, consulta el archivo ChangeLog .md del repositorio. GitHub
Enero de 2023	Actualizaciones menores y correcciones de errores. Para obtener más información sobre la versión 1.4.2, consulta el archivo ChangeLog .md del repositorio. GitHub

Date	Cambio
Mayo de 2023	<p>Se ha eliminado el indicador de programación <code>override_status</code>, el programador de recursos frente al programador de instancias. Se ha añadido una función para simplificar la programación entre cuentas y eliminar la necesidad de funciones entre AWS Organizations cuentas.</p> <p>Para obtener más información sobre la versión 1.5.0, consulte el archivo <code>ChangeLog.md</code> del repositorio. GitHub</p>
Julio de 2023	<p>Se agregó una guía sobre cómo administrar la infraestructura como código y se actualizó a la versión 1.5.x. Se agregaron funciones y beneficios adicionales para la solución de problemas.</p> <p>Para obtener más información sobre la versión 1.5.1, consulte el archivo <code>ChangeLog.md</code> del repositorio. GitHub</p>
Octubre de 2023	<p>Actualización menor, parches de seguridad y ejemplos adicionales de tablas de costos adicionales en la documentación. Para obtener más información sobre la versión 1.5.2, consulte el archivo <code>ChangeLog.md</code> del repositorio. GitHub</p>
Octubre de 2023	<p>Publique el parche de seguridad v1.5.3. Para obtener más información, consulte el archivo <code>ChangeLog.md</code> del repositorio. GitHub</p>

Date	Cambio
Diciembre de 2023	Actualización de la documentación para resolver las discrepancias; se agregó una sección de acciones a la sección. AppRegistry Para obtener más información, consulta el archivo <a href="#">ChangeLog.md</a> del repositorio. GitHub
Febrero de 2024	Publique el parche de seguridad v1.5.4. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md</a> del repositorio. GitHub
Abril de 2024	Publique el parche de seguridad v1.5.5. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md</a> del repositorio. GitHub
Junio de 2024	Versión v3.0.0: versión principal. Se agregó soporte para EC2 Auto Scaling Groups, Amazon Neptune y Amazon DocumentDB. Se agregó un panel de información operativa. Actualización de la documentación para incluir guías para operadores y desarrolladores. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md del repositorio</a> . GitHub
Junio de 2024	v3.0.1: actualización menor. Instalación de CLI modificada, instrucciones de solución de desinstalación y enlaces actualizados. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md del repositorio</a> . GitHub
Julio de 2024	v3.0.2: actualización menor. Parche de seguridad. Se ha corregido el problema de la ruta de actualización de los horarios AWS CloudFormation gestionados. Para obtener más información, consulta el archivo <a href="#">ChangeLog.md del repositorio</a> . GitHub

Date	Cambio
Julio de 2024	v3.0.3: parche de seguridad. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md</a> del repositorio. GitHub
Agosto de 2024	v3.0.4: actualización menor. Se han actualizado las instrucciones de actualización. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md del repositorio</a> . GitHub
Septiembre de 2024	v3.0.5: actualización menor. Se corrigió el enésimo error de programación del día laborable y se actualizaron los permisos de programación de RDS. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md</a> del repositorio. GitHub
Noviembre de 2024	v3.0.6: actualización menor. Se corrigió la lógica de reintentos de Amazon EC2 y Amazon RDS y se actualizaron los períodos de mantenimiento de Amazon RDS para que comenzaran 10 minutos antes. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md</a> del repositorio. GitHub
Noviembre de 2024	v3.0.7: actualización menor. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md</a> del repositorio. GitHub
Enero de 2025	v3.0.8: actualización menor. Para obtener más información, consulte el archivo <a href="#">ChangeLog.md</a> del repositorio. GitHub

# Avisos

Es responsabilidad de los clientes realizar su propia evaluación independiente de la información que contiene este documento. Este documento: (a) tiene únicamente fines informativos, (b) representa las ofertas y prácticas de productos AWS actuales, que están sujetas a cambios sin previo aviso, y (c) no implica ningún compromiso ni garantía por parte de AWS sus filiales, proveedores o licenciantes. AWS los productos o servicios se proporcionan «tal cual» sin garantías, representaciones o condiciones de ningún tipo, ya sean expresas o implícitas. AWS Las responsabilidades y obligaciones con sus clientes están reguladas por AWS acuerdos, y este documento no forma parte de ningún acuerdo entre sus clientes AWS y sus clientes ni lo modifica.

Instance Scheduler on AWS está licenciado según 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.